Documentation 8.2

ZABBIX

05.11.2026

Contents

Manuel utilisateur **5**

Bienvenue dans la documentation Zabbix **5**

Premiers pas avec Zabbix . . . 5
Zabbix Cloud . . . 5
Application mobile . . . 5
Centre de développement . . . 5
Communauté et autres ressources . . . 5
Installation et premières étapes . . . 6
Installation de l'interface web . . . 6
Guides de démarrage rapide . . . 11
Installation : guides supplémentaires . . . 85
Mise à niveau . . . 168
Problèmes connus . . . 185
Prérequis . . . 193
Nouveautés et notes de mise à niveau : 8.0 . . . 204
Notes de mise à niveau . . . 204
Modifications incompatibles . . . 204
Autres . . . 204
Nouveautés et notes de mise à niveau : 8.0.x . . . 204
Appliance Zabbix . . . 205
Processus . . . 208
Serveur . . . 210
proxy . . . 238
agent . . . 284
Agent 2 . . . 315
Java gateway . . . 380
Expéditeur . . . 384
Obtenir . . . 387
JS . . . 388
Service web . . . 389
Adaptateur bridge . . . 391
serveur MCP . . . 395
Variables d'environnement . . . 402
Configuration . . . 404
Hôtes et groupes d'hôtes . . . 416
Éléments . . . 435
Détection des problèmes avec les déclencheurs . . . 663
Visualisation . . . 716
Modèles et groupes de modèles . . . 743
Modèles prêts à l'emploi . . . 743
Notifications lors des événements . . . 757
Macros . . . 811
Utilisateurs et groupes d'utilisateurs . . . 826
Rapports planifiés . . . 837
Exportation des données . . . 841
Export/import de configuration . . . 850
Expressions régulières . . . 917
Événements et résolution . . . 924
Accusé de réception et suppression des problèmes . . . 926
Fermeture manuelle des problèmes . . . 930

Périodes de maintenance . . . 931
Étiquetage . . . 937
Complex event processing . . . 940
Corrélation des événements . . . 944
Découverte . . . 951
Découverte réseau . . . 951
Enregistrement automatique de l'agent actif . . . 960
Découverte de bas niveau . . . 964
Surveillance des machines virtuelles . . . 989
Clés d'élément de surveillance VMware . . . 991
Champs de découverte des machines virtuelles . . . 1009
Exemples JSON pour les éléments VMware . . . 1013
Exemple de configuration de surveillance VMware . . . 1018
Surveillance web . . . 1022
Éléments de surveillance web . . . 1031
Scénario réel . . . 1032
Surveillance des services . . . 1039
Arborescence des services . . . 1040
SLA . . . 1044
Exemple de configuration . . . 1045
Mise à niveau de la surveillance des services . . . 1050
Chiffrement et sécurité . . . 1051
Chiffrement à l'aide de certificats . . . 1059
Chiffrement à l'aide de clés prépartagées . . . 1067
Connexion sécurisée à la base de données . . . 1070
Connexion sécurisée entre l'interface et le serveur . . . 1082
Contrôle d'accès . . . 1086
Cryptographie . . . 1090
Sécurisation du serveur web . . . 1091
Stockage des secrets . . . 1094
Confiance du certificat TLS de Google Chrome . . . 1103
Bonnes pratiques de sécurité . . . 1105
Interface . . . 1105
Menu . . . 1106
Sections de l'interface . . . 1116
Paramètres utilisateur . . . 1394
Recherche globale . . . 1399
Mode de maintenance de l'interface . . . 1401
Paramètres de la page . . . 1402
Définitions . . . 1405
Langues supplémentaires de l'interface . . . 1405
Créer votre propre thème . . . 1406
Mode de débogage . . . 1408
Cookies utilisés par Zabbix . . . 1408
Fuseaux horaires . . . 1409
Réinitialisation du mot de passe . . . 1410
Période et sélecteurs d'hôte . . . 1411
Paramètres de configuration : interface Zabbix . . . 1413
API . . . 1419
Référence des méthodes . . . 1424
Annexe 1. Commentaire de référence . . . 2318
Annexe 2. Modifications de la version 7.4 à la version 8.0 . . . 2324
Annexe 3. Modifications dans la version 8.0 . . . 2324
Extensions . . . 2324
Modules chargeables . . . 2327
Modules de l'interface . . . 2336
Annexes . . . 2337
Installation et configuration . . . 2337
Éléments . . . 2384
Fonctions prises en charge . . . 2425
Macros . . . 2425
Compatibilité des versions . . . 2464
Syntaxe de la période de temps . . . 2466

Symboles d'unité . . . 2466
Exécution de commandes . . . 2468
Exemples d'échappement . . . 2469
Qu'est-ce que Zabbix . . . 2470

Zabbix Cloud **2475**

Zabbix Cloud **2475**
Configurer et gérer Zabbix Cloud . . . 2475
Découvrez Zabbix Cloud . . . 2475
Déployer Zabbix dans le cloud . . . 2475
Configuration du nœud . . . 2485
Présentation . . . 2486
Onglet Vue d'ensemble . . . 2487
Onglet des filtres d'accès . . . 2490
Onglet Chiffrement . . . 2491
Onglet Backups . . . 2491
Onglet Historique . . . 2492
Onglet Maintenance . . . 2493
Onglet Upgrade . . . 2494
Ajout d'utilisateurs . . . 2495
Introduction . . . 2495
Ajout d'utilisateurs aux organisations . . . 2495
Ajouter des utilisateurs aux nœuds . . . 2497
Principales différences de Zabbix Cloud . . . 2497
Introduction . . . 2497
Journal d'audit . . . 2500
Configurer un proxy Zabbix Cloud . . . 2502

Application mobile **2510**

Application mobile **2510**
Configurer Zabbix pour l'application . . . 2510
Découvrir l'application . . . 2511
Configurer Zabbix pour l'application . . . 2511
Présentation de l'application . . . 2512
Accès mobile sécurisé . . . 2520

Centre de développement **2521**

Centre de développement **2521**
Développer sur Zabbix . . . 2521
Création d'un plugin . . . 2521
Interfaces de plugins . . . 2527
Bibliothèque Python pour Zabbix . . . 2529
Installation . . . 2529
Guide de démarrage rapide . . . 2529
Utiliser l'API Zabbix . . . 2533
Collecter des données depuis l'agent Zabbix . . . 2536
Envoyer des données au serveur ou au proxy Zabbix . . . 2538
Journalisation de débogage . . . 2541
Modules . . . 2542
Structure des fichiers des modules . . . 2542
Widgets . . . 2550
Tutoriels . . . 2559
Exemples . . . 2585
Modifications du développement des extensions . . . 2586

Pages de manuel Zabbix **2586**
zabbix_agent2 . . . 2586
zabbix_agentd . . . 2587
zabbix_get . . . 2589
zabbix_js . . . 2591
zabbix_proxy . . . 2592
zabbix_sender . . . 2594

zabbix_server 2599
zabbix_web_service 2601

Mention de copyright **2601**

Manuel utilisateur

Bienvenue dans la documentation Zabbix

Votre ressource de référence pour utiliser la supervision Zabbix, des configurations de base aux paramétrages avancés. Ce manuel couvre tout ce qui est nécessaire pour installer, configurer et exploiter Zabbix.

Nouveautés de Zabbix 8.0

Premiers pas avec Zabbix

Installation Instructions étape par étape pour installer Zabbix sur votre plateforme préférée, couvrant divers systèmes d'exploitation et configurations de bases de données.

Prérequis Une liste des plateformes prises en charge et des prérequis logiciels pour vous aider à préparer votre environnement en vue d'un déploiement réussi de Zabbix.

Guides de démarrage rapide Des guides pratiques et concis, orientés tâches, qui vous accompagnent dans les bases, depuis vos premières étapes de configuration jusqu'à la réception de votre première alerte de problème.

Zabbix Cloud

Commencez à utiliser Zabbix Cloud Instructions guidées pour mettre en service rapidement votre instance Zabbix Cloud, y compris la configuration initiale et la connexion de vos appareils surveillés.

Zabbix Cloud vs. sur site Comparez les différences en matière de gestion, d'évolutivité et de maintenance entre les deux déploiements, et découvrez lequel peut prendre en charge votre flux de travail de la manière la plus fluide.

Découvrez Zabbix Cloud Obtenez une vue d'ensemble de Zabbix Cloud, y compris ses principales fonctionnalités, ses avantages et ce qui est inclus dans un abonnement. Pour les questions fréquentes sur la configuration et l'utilisation, consultez la FAQ.

Application mobile

Obtenir l'application Installez l'application mobile Zabbix (bientôt disponible) sur votre appareil et commencez à surveiller votre environnement Zabbix où que vous soyez.

Configurer Zabbix pour l'application Installez l'adaptateur bridge Zabbix pour permettre aux appareils mobiles de se connecter à Zabbix. Ajoutez ensuite votre appareil et activez les notifications push.

Présentation de l'application Découvrez les bases de l'utilisation de l'application mobile Zabbix.

Centre de développement

Modules frontend Guides et références pour créer des modules personnalisés qui étendent ou modifient le frontend de Zabbix afin de répondre à des cas d'utilisation spécifiques.

Widgets Une présentation de la structure et de la logique des widgets, avec des instructions pour créer des éléments de tableau de bord personnalisés adaptés à vos besoins.

Plugins Un aperçu de la manière de développer et de gérer des plugins Zabbix afin d'étendre les fonctionnalités ou de s'intégrer à des systèmes externes.

Communauté et autres ressources

Forums Zabbix Un espace pour échanger des idées, trouver des solutions et partager des expériences à tous les niveaux d'expertise sur Zabbix.

Blog Zabbix Actualités, tutoriels et études de cas sélectionnés par l'équipe Zabbix et les contributeurs de la communauté.

Vidéos pratiques Une vidéothèque proposant des démonstrations, des discussions et des conseils pour vous aider à tirer le meilleur parti de Zabbix.

Installation et premières étapes

La séquence suivante décrit le processus de configuration de Zabbix :

1. Vérifiez que le logiciel requis est installé et en cours d'exécution.
2. Installez Zabbix.
3. Installez l'interface web de Zabbix.
4. Commencez la surveillance.

Les détails de chaque étape sont fournis ci-dessous.

Logiciels requis Zabbix fonctionne bien avec les logiciels suivants (souvent appelés pile LAMP) :

- **Système d'exploitation Linux :** Ubuntu, Debian, CentOS ou autres
- **Serveur web :** Apache ou NGINX
- **Serveur de base de données :** MySQL/MariaDB ou PostgreSQL
- **PHP**

Sur la plupart des systèmes Linux modernes, PHP est peut-être déjà installé. Dans ce cas, vous devrez peut-être uniquement configurer le serveur de base de données et le serveur web.

Pour plus d'informations, notamment sur les versions des logiciels prises en charge, consultez la page Logiciels requis.

Installer Zabbix La méthode **la plus simple** pour installer Zabbix consiste à utiliser les **paquets de la distribution** (ou, alternativement, des conteneurs). Les paquets installent les dépendances logicielles mineures.

Des instructions d'installation précises, adaptées à votre système d'exploitation, sont disponibles sur le **site web de Zabbix**.

Sélectionnez votre plateforme et suivez les instructions fournies. Il est recommandé de commencer par installer un serveur Zabbix, un agent Zabbix 2 et l'interface. Vous pourrez ajouter des proxies Zabbix et d'autres agents ultérieurement.

Une partie du processus d'installation consiste à créer une base de données (pour conserver les données de configuration et d'historique) et à y importer les données initiales.

Installer l'interface web de Zabbix Suivez toutes les étapes de l'**assistant d'installation** de l'interface web.

Le mot de passe de la base de données (créé à l'étape précédente) vous sera demandé afin que l'interface puisse communiquer avec la base de données.

Commencer la supervision Connectez-vous à l'interface web de Zabbix.

Utilisez les **guides de démarrage rapide** pour commencer à superviser des ressources courantes (Linux, Apache, MySQL ou autres). Ces guides expliquent avec quels modèles travailler pour commencer à superviser des ressources courantes.

Vous pouvez commencer par le guide de démarrage rapide consacré à la supervision d'un serveur Linux.

Installation de l'interface web

Cette section fournit des instructions pour installer l'interface web Zabbix. Suivez ces étapes après avoir installé le backend Zabbix.

L'interface Zabbix est écrite en PHP et nécessite un serveur web pris en charge, tel qu'Apache ou NGINX, avec PHP configuré. Consultez les prérequis de l'interface pour obtenir des informations sur les logiciels et bibliothèques requis.

Pour utiliser une langue autre que l'anglais dans l'interface web Zabbix, ses paramètres régionaux doivent être installés sur le serveur web, ainsi que l'extension PHP gettext. Veuillez noter que la modification de la langue peut modifier le format de l'heure et de la date.

Pour configurer SSL pour l'interface Zabbix, consultez Configuration de SSL pour l'interface Zabbix.

Assistant d'installation

Pour installer l'interface, ouvrez l'URL de l'interface Zabbix dans un navigateur :

- Pour Apache : `http://<server_ip_or_name>/zabbix`
- Pour NGINX : `http://<server_ip_or_name>`

Écran d'accueil

L'assistant d'installation de l'interface s'ouvre sur l'écran d'accueil.

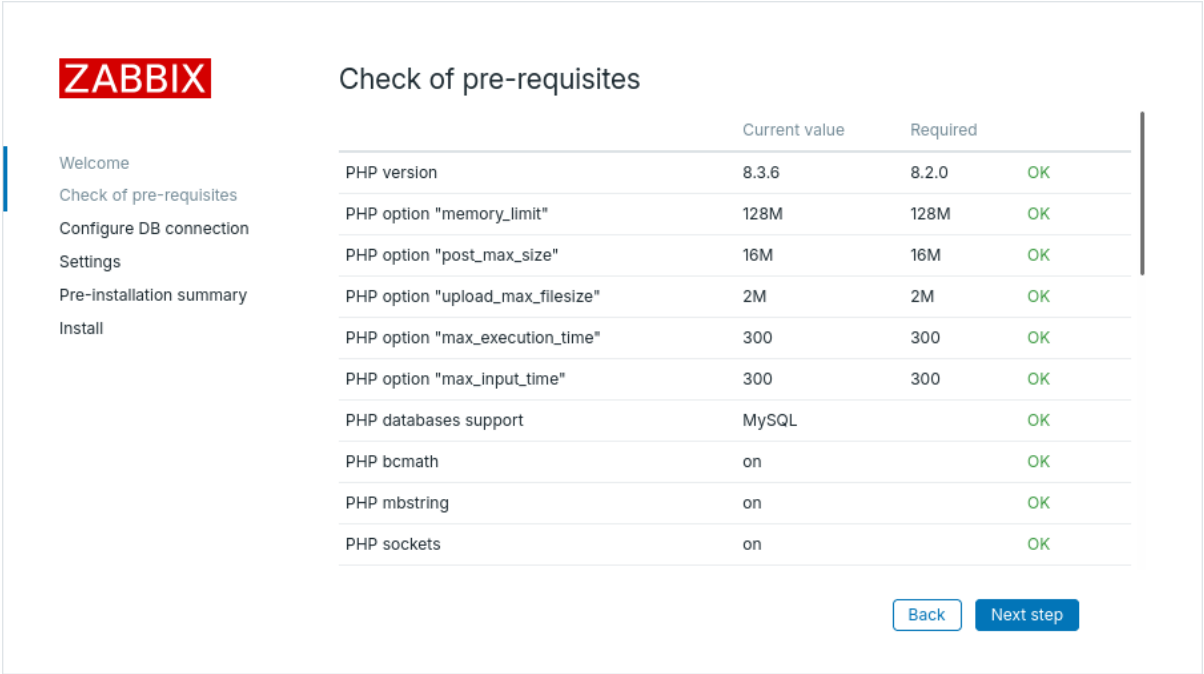


La langue par défaut du système est l'anglais (en_US), avec le format d'heure et de date des États-Unis. Pour modifier la langue par défaut du système, sélectionnez la langue souhaitée dans le menu déroulant *Langue par défaut*.

Cliquez sur le bouton *Étape suivante* pour continuer.

Vérification des prérequis

Assurez-vous que tous les prérequis obligatoires de l'interface Zabbix sont satisfaits.



Consultez la page Prérequis pour obtenir la liste des extensions PHP requises et facultatives, la version PHP requise et les bases de données prises en charge. Notez que les prérequis facultatifs non satisfaits sont affichés avec l'état *Avertissement*, et que le processus d'installation peut se poursuivre même s'ils ne sont pas satisfaits.

Valeurs requises pour les directives de configuration PHP dans le fichier `php.ini` utilisé par le serveur web :

```
memory_limit = 128M ; Valeur minimale
post_max_size = 16M ; Valeur minimale
upload_max_filesize = 2M ; Valeur minimale
```

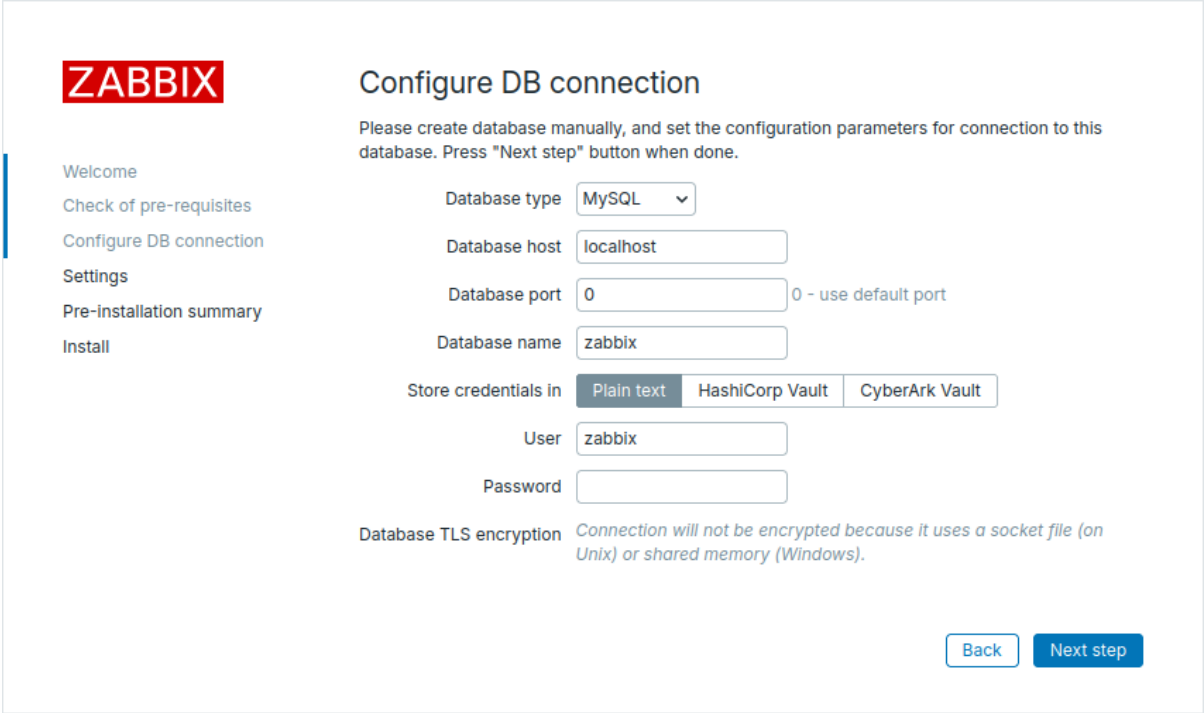
```
max_execution_time = 300 ; Valeur minimale, les valeurs 0 et -1 sont également autorisées
max_input_time = 300 ; Valeur minimale, les valeurs 0 et -1 sont également autorisées
session.auto_start = 0 ; Doit être désactivé
arg_separator.output = "&" ; "&amp;" est également autorisé
```

Attention:
Si l'utilisateur ou le groupe du serveur web doit être modifié, vérifiez que le nouvel utilisateur ou groupe dispose des autorisations requises pour le répertoire des sessions PHP ; sinon, l'installation de Zabbix risque de ne pas pouvoir continuer.

Si tous les prérequis obligatoires sont satisfaits, passez à l'étape suivante.

Configurer la connexion à la base de données

Saisissez les informations nécessaires pour vous connecter à la base de données.



Si le champ *Hôte de la base de données* est laissé vide, le socket de domaine Unix par défaut est utilisé.

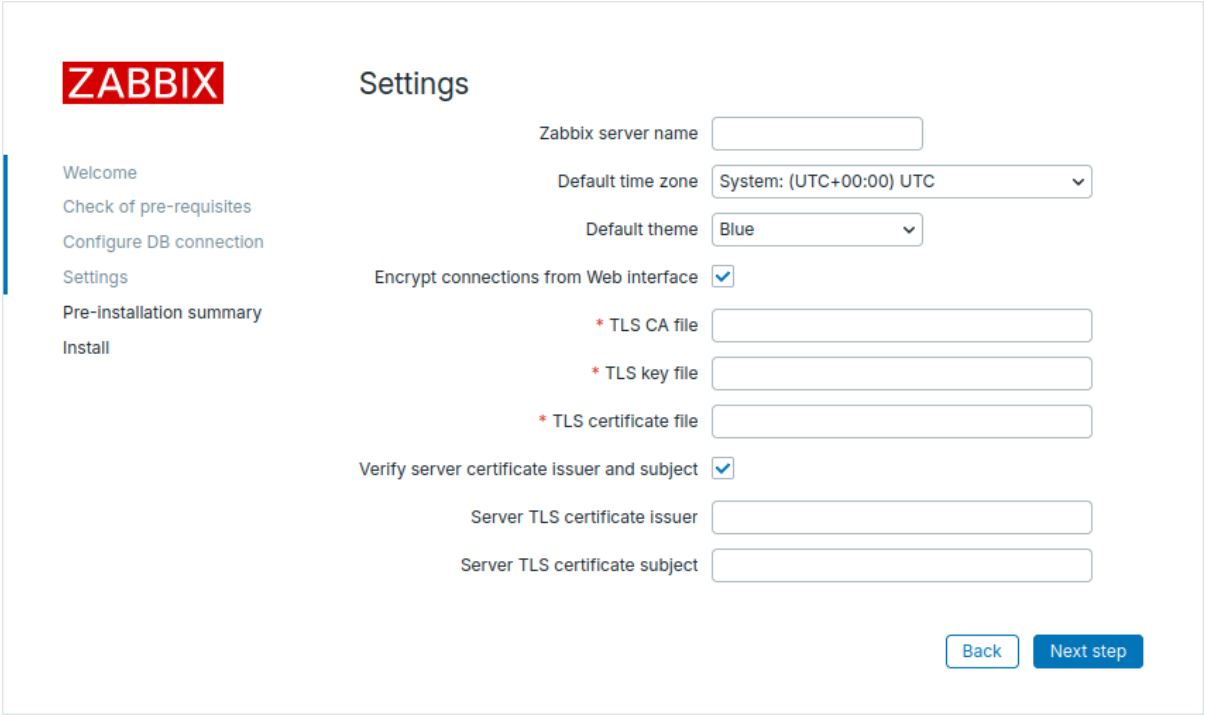
Pour MySQL, saisir `localhost` dans le champ *Hôte de la base de données* entraîne également une connexion via le socket de domaine Unix par défaut. L'assistant ne permet pas de configurer le socket de la base de données. Pour utiliser un socket personnalisé, configurez-le dans les paramètres du serveur Zabbix (par exemple, avec le paramètre DBSocket= dans le fichier de configuration du serveur Zabbix (`zabbix_server.conf`)). Cela garantit la cohérence de l'interface avec les paramètres de connexion du serveur à la base de données.

Pour PostgreSQL, le champ *Hôte de la base de données* peut contenir plusieurs entrées `host:port` séparées par des virgules. La chaîne est divisée en parties et chaque port est validé de la même manière que dans le champ *Port de la base de données*. Après validation, la chaîne est analysée pour extraire les hôtes et les ports afin de construire la chaîne de connexion. Si aucun port n'est défini pour un hôte dans la liste séparée par des virgules, la valeur de *Port de la base de données* est utilisée comme port par défaut. Si un chemin de socket est spécifié (par exemple, `/var/run/pgbouncer`), ce socket de domaine Unix est utilisé à la place.

Pour configurer une connexion sécurisée à la base de données, cochez la case *Chiffrement TLS de la base de données*, puis consultez Interface vers la base de données pour configurer les champs. La case *Chiffrement TLS de la base de données* est désactivée dans les cas suivants : - Pour MySQL, *Hôte de la base de données* est défini sur *localhost*. - Pour PostgreSQL, la valeur du champ *Hôte de la base de données* commence par une barre oblique ou le champ est vide.

Pour configurer Zabbix afin qu'il récupère les secrets depuis HashiCorp Vault, sélectionnez *HashiCorp Vault* pour *Stocker les identifiants dans*, puis consultez Configuration de HashiCorp pour remplir les champs supplémentaires. Pour configurer Zabbix afin qu'il récupère les secrets depuis CyberArk Vault, sélectionnez *CyberArk Vault* pour *Stocker les identifiants dans*, puis consultez Configuration de CyberArk pour remplir les champs supplémentaires.

Paramètres



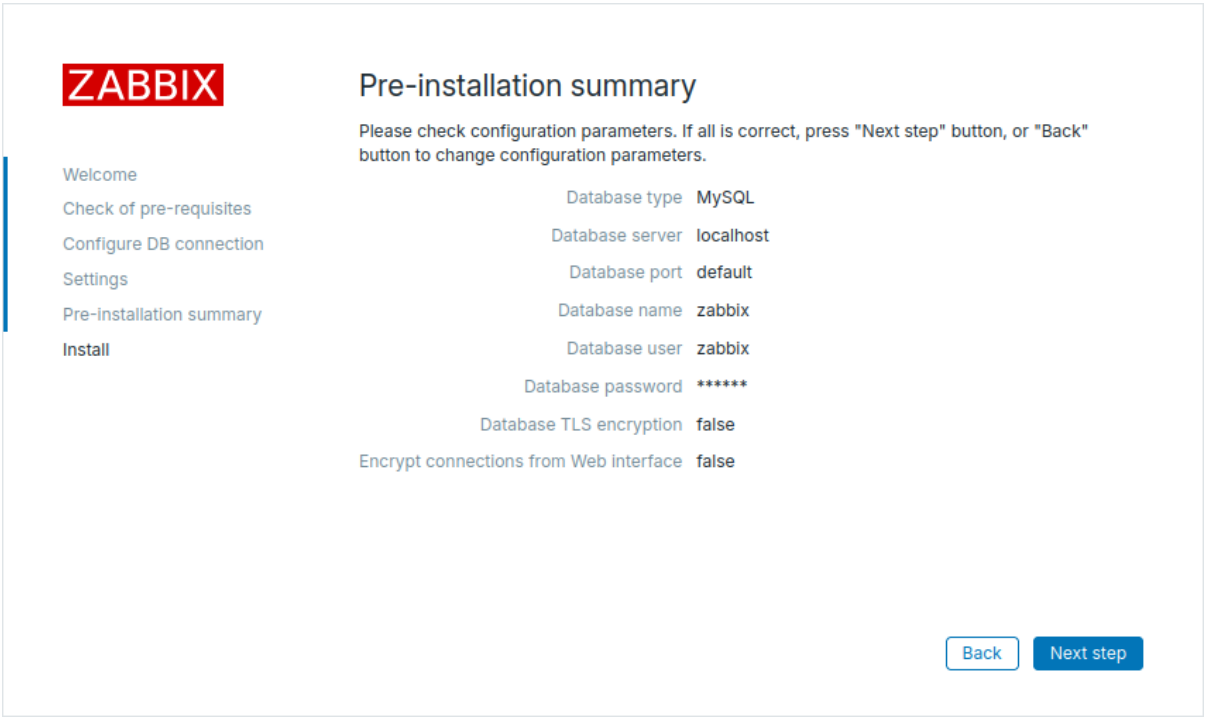
La saisie d'un nom pour le serveur Zabbix est facultative ; toutefois, si un nom est saisi, il sera affiché dans la barre de menu et dans les titres des pages.

Définissez le fuseau horaire et le thème par défaut de l'interface.

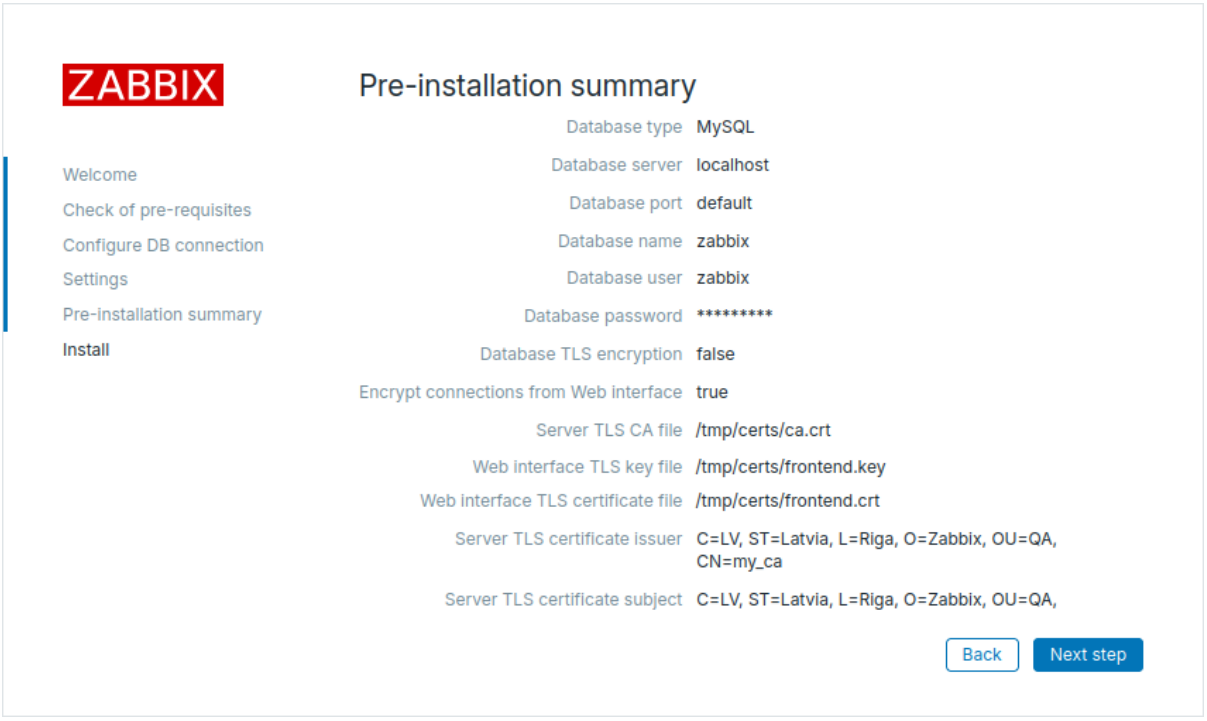
Pour configurer les connexions TLS entre l'interface Zabbix et le serveur Zabbix, cochez la case *Chiffrer les connexions depuis l'interface Web*, puis consultez la page Configuration de la connexion TLS pour remplir les champs supplémentaires.

Résumé de la préinstallation

Consultez un résumé des paramètres.



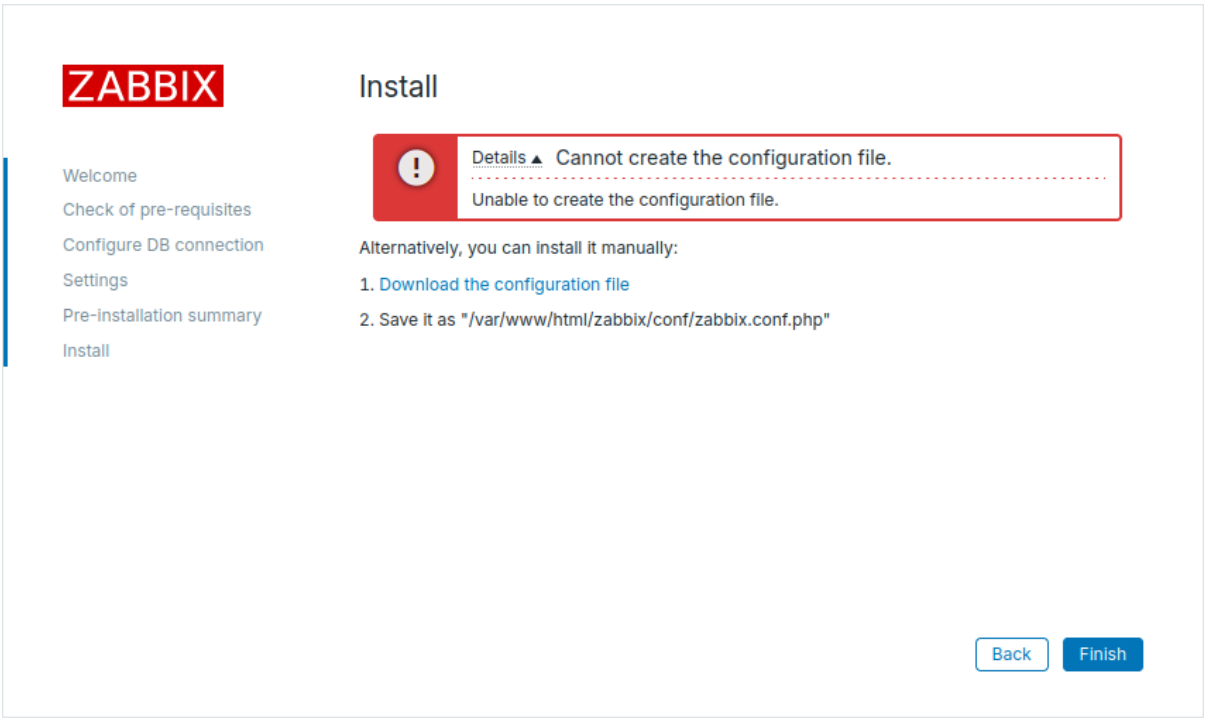
Si TLS est configuré, l'assistant affichera ses détails.



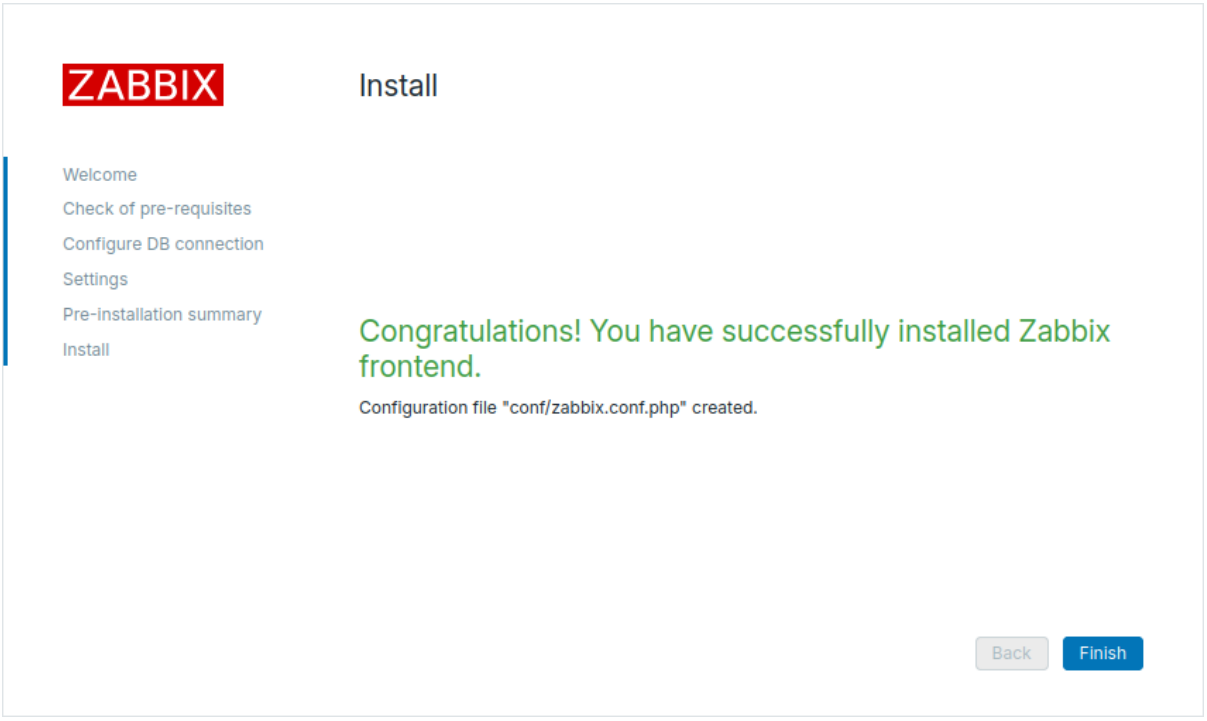
Installation

Si l'utilisateur du serveur web dispose d'un accès en écriture au répertoire `conf/`, le fichier de configuration sera enregistré automatiquement et vous pourrez passer immédiatement à l'étape suivante.

Si l'utilisateur du serveur web ne dispose pas d'un accès en écriture au répertoire `conf/`, vous devez télécharger le fichier de configuration et le placer dans le répertoire `conf/` du répertoire des documents HTML du serveur web, où vous avez copié les fichiers PHP de Zabbix.

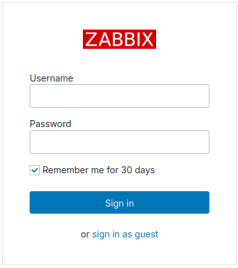


Terminez l'installation.



Se connecter

Connectez-vous avec les identifiants par défaut : - Nom d'utilisateur : *Admin* - Mot de passe : *zabbix*



Fichier de configuration de l'interface

Après l'installation, vous pouvez reconfigurer l'interface Zabbix en mettant à jour les paramètres dans le fichier de configuration de l'interface `zabbix.conf.php`.

Le chemin par défaut vers le fichier de configuration dépend du système d'exploitation et de la méthode d'installation de Zabbix :

- Pour une installation à partir de paquets : `/usr/share/zabbix/ui/conf/zabbix.conf.php`
- Pour une installation à partir des sources : `/var/www/html/zabbix/conf/zabbix.conf.php`
- Pour l'appliance : `/etc/zabbix/web/zabbix.conf.php`

Consultez les options du fichier de configuration pour plus d'informations sur la configuration de l'interface Zabbix.

Guides de démarrage rapide

Après avoir installé Zabbix et son interface, il est temps de commencer la supervision.

Utilisez ces guides rapides pour commencer à superviser les ressources courantes :

- Supervision d'un serveur Linux
- Supervision d'un serveur web Apache
- Supervision d'un serveur MySQL

Les guides de démarrage rapide se concentrent sur l'utilisation des modèles officiels Zabbix pour la supervision. Un modèle est un ensemble préconfiguré d'éléments de supervision. Un modèle vous permet de faire évoluer la supervision : vous pouvez tester un modèle sur une seule cible de supervision (appareil, application, etc.), l'ajuster, puis l'appliquer à plusieurs cibles de supervision similaires.

Bien que les modèles conviennent à la supervision prête à l'emploi, vous pouvez également être intéressé par la personnalisation de ses éléments.

Pour mieux comprendre les éléments de base qui composent un modèle, consultez Configuration de base.

Configuration de base

Ces guides présentent les éléments de base de la configuration de Zabbix :

- Connexion et configuration de l'utilisateur
- Nouvel hôte
- Nouvel élément
- Nouveau déclencheur
- Réception des notifications de problème
- Nouveau modèle

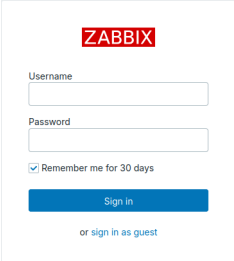
Connexion et configuration de l'utilisateur

Vue d'ensemble

Dans cette section, vous apprendrez à vous connecter et à configurer un utilisateur système dans Zabbix.

Connexion

Ouvrez l'URL de l'interface Zabbix dans un navigateur : - Pour Apache : http://<server_ip_or_name>/zabbix - Pour NGINX : http://<server_ip_or_name>



Il s'agit de l'écran d'accueil de Zabbix. Saisissez le nom d'utilisateur **Admin** et le mot de passe **zabbix** pour vous connecter en tant que superutilisateur Zabbix. L'accès à toutes les sections du menu sera accordé.

Pour des raisons de sécurité, il est fortement recommandé de modifier immédiatement le mot de passe par défaut du compte Admin après la première connexion.

Connexion persistante

Pour rester connecté jusqu'à 30 jours, sélectionnez *Se souvenir pendant 30 jours* avant de cliquer sur *Se connecter*.

Se souvenir pendant 30 jours activé :

- Votre session reste active pendant 30 jours.
- *Déconnexion automatique* est remplacée, ce qui vous maintient connecté jusqu'à la fin de cette période.
- Vous serez connecté automatiquement lors de vos prochaines visites dans les 30 jours, sans avoir à ressaisir vos identifiants.

Se souvenir pendant 30 jours désactivé :

- Toute connexion automatique précédemment activée est supprimée.
- La session expirera selon l'intervalle *Déconnexion automatique* configuré.

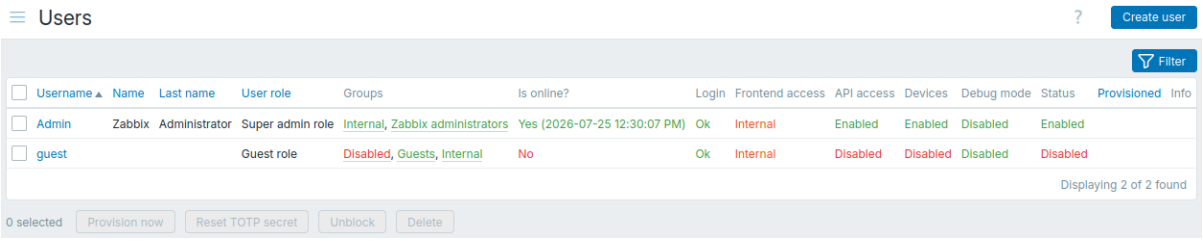
Protection contre les attaques par force brute

En cas de cinq tentatives de connexion échouées consécutives, l'interface Zabbix fera une pause de 30 secondes afin de prévenir les attaques par force brute et par dictionnaire.

L'adresse IP d'une tentative de connexion échouée sera affichée après une connexion réussie.

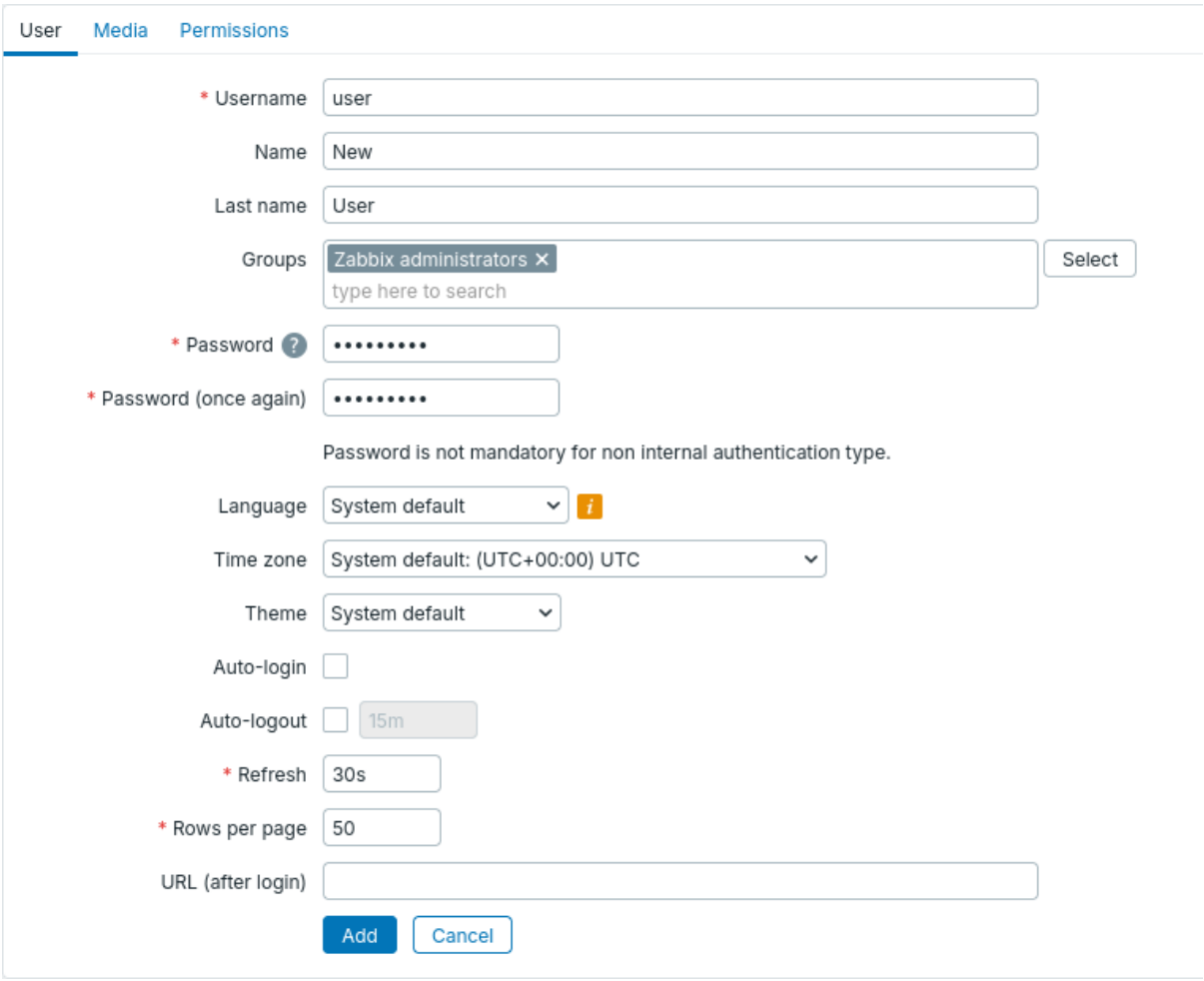
Ajout d'un utilisateur

Pour afficher les informations sur les utilisateurs, allez dans *Users* > *Users* dans le menu vertical de la barre latérale.



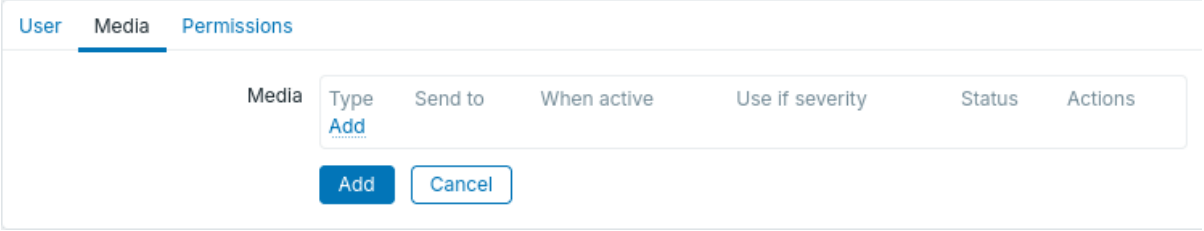
Pour ajouter un nouvel utilisateur, sélectionnez *Create user* dans le coin supérieur droit.

Dans le formulaire du nouvel utilisateur, veillez à ajouter votre utilisateur à l'un des groupes d'utilisateurs existants, par exemple 'Zabbix administrators'.



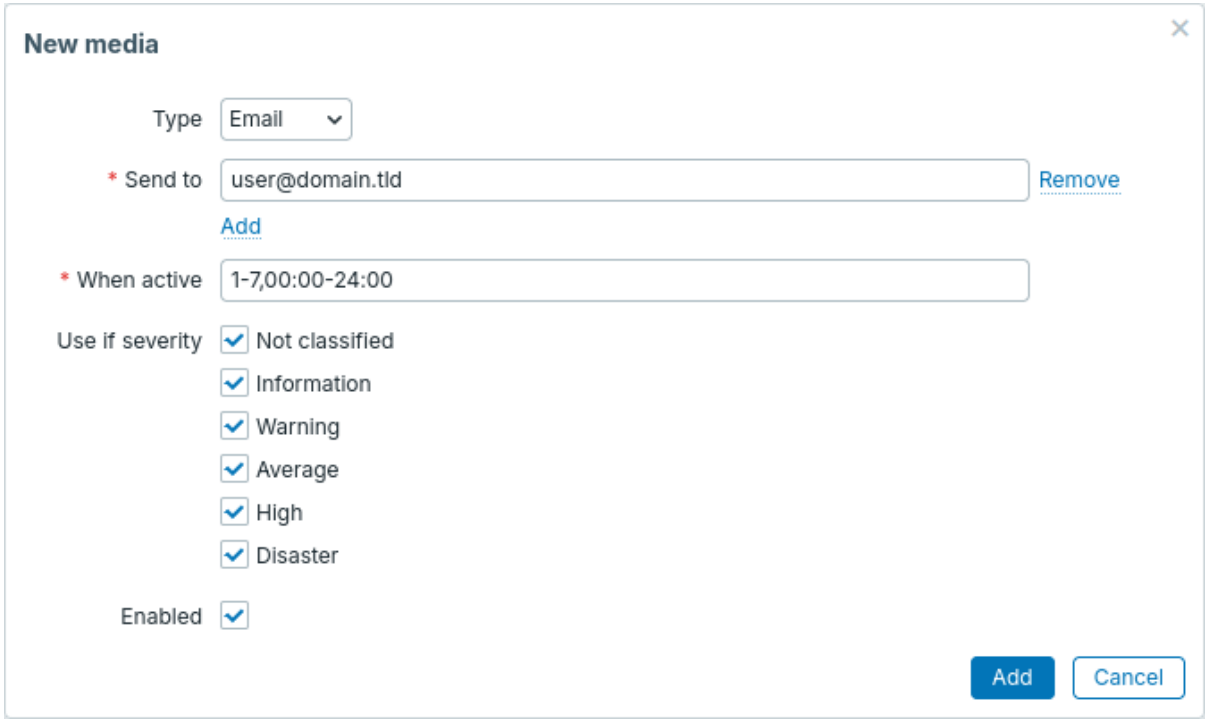
Tous les champs de saisie obligatoires sont marqués d'un astérisque rouge. Pour plus de détails sur les champs de saisie de ce formulaire de configuration, consultez la page User settings.

Par défaut, aucun média (méthodes de livraison des notifications) n'est défini pour les nouveaux utilisateurs. Pour en créer un, allez dans l'onglet *Media* et cliquez sur *Add*.



Dans la fenêtre contextuelle, saisissez l'adresse e-mail de l'utilisateur.
 Tous les types de média sont désactivés par défaut et peuvent être activés dans *Alerts* > *Media types*.

Vous pouvez spécifier une période pendant laquelle le média sera actif (voir la page Time period specification pour une description du format). Par défaut, un média est toujours actif. Vous pouvez également personnaliser les niveaux de trigger severity pour lesquels le média sera actif, mais laissez-les tous activés pour le moment.



Cliquez sur *Add* pour enregistrer le média, puis allez dans l'onglet *Permissions*.

L'onglet Permissions comporte un champ obligatoire *Role*. Le rôle détermine quels éléments de l'interface l'utilisateur peut voir et quelles actions il est autorisé à effectuer. Cliquez sur *Select* et sélectionnez l'un des rôles dans la liste. Par exemple, sélectionnez *Admin role* pour autoriser l'accès à toutes les sections de l'interface Zabbix, à l'exception de *Administration*. Par la suite, vous pourrez modifier les permissions ou créer d'autres rôles d'utilisateur. Après la sélection d'un rôle, les permissions apparaîtront dans le même onglet :

Users

User | Media | Permissions

* Role: Admin role ✕ | Select

User type: Admin

Permissions

User group permissions

Group	Type	Permissions
All groups	Hosts	None
All groups	Templates	None

Proxies: None

Proxy groups: None

Permissions can be assigned for user groups only.

Access to UI elements

Dashboards: Dashboards

Monitoring: Problems | Hosts | Latest data | Maps | Discovery

APM: Traces | Metrics | Logs

Services: Services | SLA | SLA report

Inventory: Overview | Hosts

Reports: System information | Scheduled reports | Availability report | Top 100 triggers | Notifications

Data collection: Template groups | Host groups | Templates | Hosts | Maintenance | Discovery

Alerts: Trigger actions | Service actions | Discovery actions | Autoregistration actions | Internal actions

User profile settings

Redirect URL after login

Access to services

Read-write access to services: None

Read-only access to services: All

Access to modules

Action log | Clock | Discovery status | Favorite graphs | Favorite maps | Gauge | Geomap | Graph | Graph (classic) | Graph prototype | Honeycomb | Host availability | Host card | Host navigator | Item card | Item history | Item navigator | Item value | Map | Map navigation tree | Pie chart | Problem hosts | Problems | Problems by severity | Scatter plot | SLA report | System information | Top hosts | Top items | Top triggers | Trigger overview | URL | Web monitoring

Access to API

Disabled

Devices

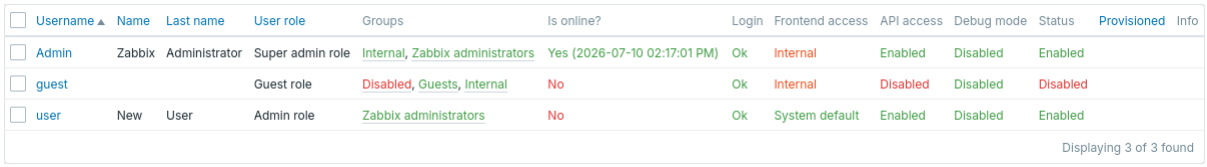
Enabled | Manage own devices

Access to actions

Create and edit dashboards | Create and edit maps | Create and edit maintenance | Add problem comments | Change severity | Acknowledge problems | Suppress problems | Close problems | Execute scripts | Manage API tokens | Manage scheduled reports | Manage SLA | Invoke "Execute now" on read-only hosts | Change problem ranking | Create and edit own media | Select "Server" for monitoring and discovery

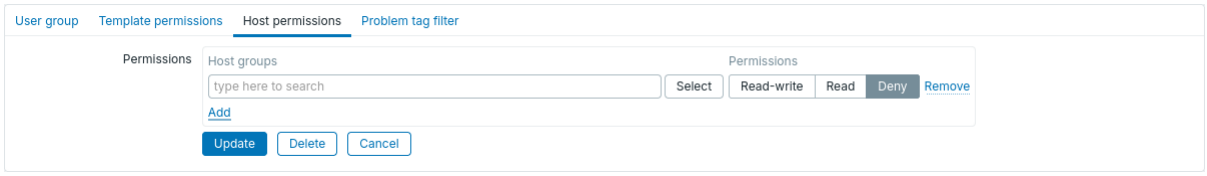
Add | Cancel

Cliquez sur *Add* dans le formulaire des propriétés de l'utilisateur pour enregistrer l'utilisateur. Le nouvel utilisateur apparaît dans la liste des utilisateurs.

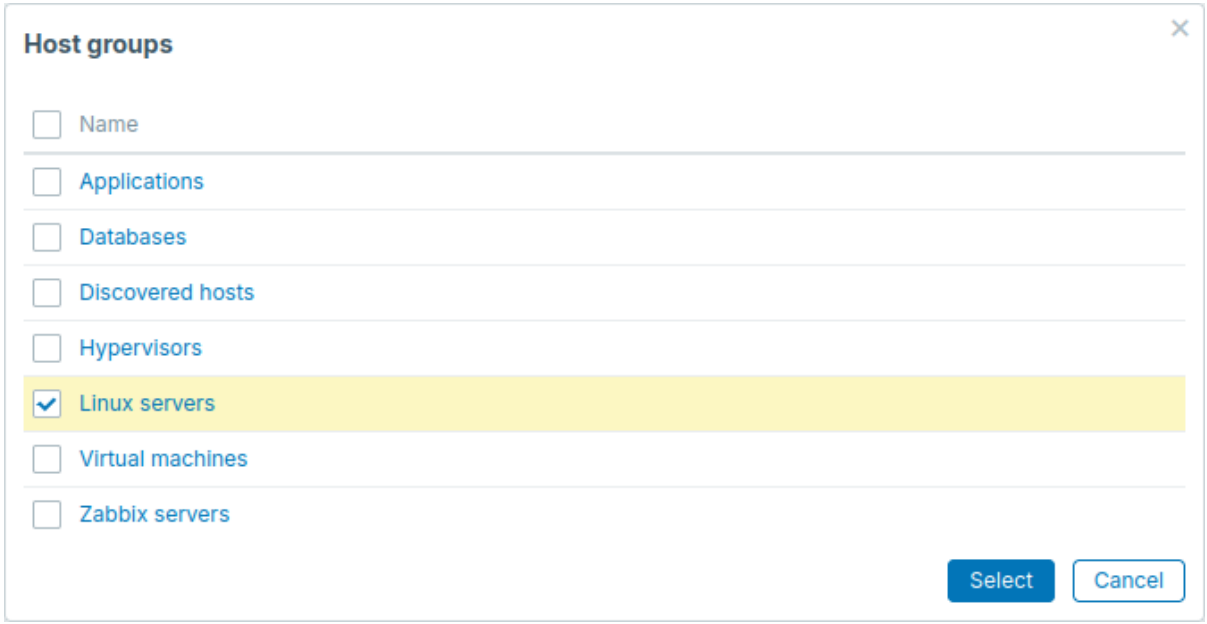


Ajout des permissions

Par défaut, un nouvel utilisateur n'a aucune permission pour accéder aux hôtes et aux modèles. Pour accorder des droits à l'utilisateur, cliquez sur le groupe de l'utilisateur dans la colonne *Groups* (dans ce cas - *Zabbix administrators*). Dans le formulaire de propriétés *User groups*, allez à l'onglet *Host permissions* pour attribuer des permissions aux groupes d'hôtes. Cliquez sur Add pour afficher le champ de sélection du groupe d'hôtes :



Cliquez ensuite sur *Select* à côté du champ pour voir la liste des groupes d'hôtes. Cet utilisateur doit avoir un accès en lecture seule au groupe *Linux servers* ; cochez donc la case appropriée dans la liste et cliquez sur *Select* pour confirmer votre choix.



Cliquez sur le bouton *Read* pour définir le niveau de permission, puis sur *Update* pour enregistrer les modifications apportées à la configuration du groupe d'utilisateurs.

Pour accorder des permissions aux modèles, vous devrez passer à l'onglet *Template permissions* et spécifier des groupes de modèles. Les étapes sont identiques à l'attribution de permissions aux groupes d'hôtes. Un aperçu des modèles est disponible dans la section Nouveau modèle de ce Quickstart.

> **Attention:**
> Dans Zabbix, les droits d'accès aux hôtes et aux modèles sont attribués aux groupes d'utilisateurs, et non aux utilisateurs individuels.

C'est fait ! Vous pouvez essayer de vous connecter en utilisant les identifiants du nouvel utilisateur.

Nouvel hôte

Vue d'ensemble

Dans cette section, vous apprendrez à configurer un nouvel hôte.

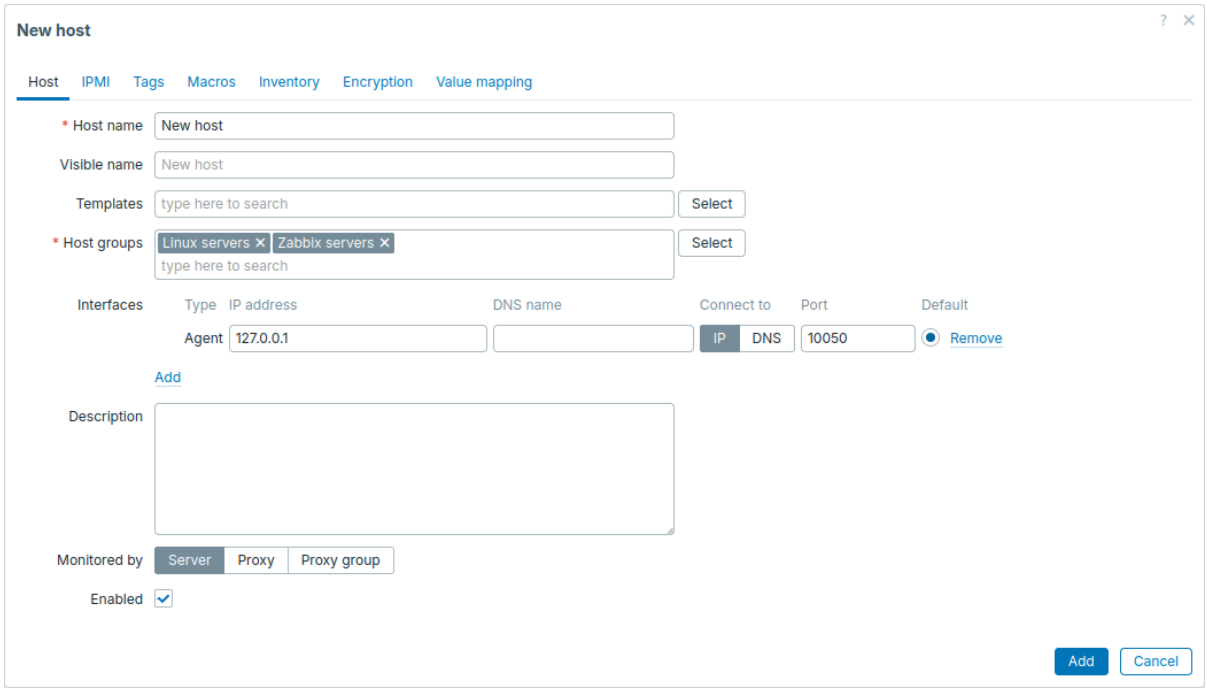
Un hôte dans Zabbix est une entité en réseau (physique, virtuelle) que vous souhaitez surveiller. La définition de ce qui peut être un "hôte" dans Zabbix est assez flexible. Il peut s'agir d'un serveur physique, d'un commutateur réseau, d'une machine virtuelle

ou d'une application.

Ajout d'un hôte

Les informations sur les hôtes configurés dans Zabbix sont disponibles dans la section de menu *Collecte de données* > *Hôtes*, ainsi que dans la section de menu *Supervision* > *Hôtes*. Un hôte prédéfini, appelé « Zabbix server », existe déjà, mais nous allons apprendre à en ajouter un autre.

Pour ajouter un nouvel hôte, cliquez sur *Créer un hôte*. Un formulaire de configuration de l'hôte s'ouvre.



Tous les champs obligatoires sont marqués d'un astérisque rouge.

Le minimum à saisir ici est le suivant :

Nom de l'hôte

- Saisissez un nom d'hôte. Les caractères alphanumériques, les espaces, les points, les tirets et les traits de soulignement sont autorisés.

Groupes d'hôtes

- Sélectionnez un ou plusieurs groupes existants en cliquant sur le bouton *Sélectionner*, ou saisissez le nom d'un groupe inexistant pour en créer un nouveau.

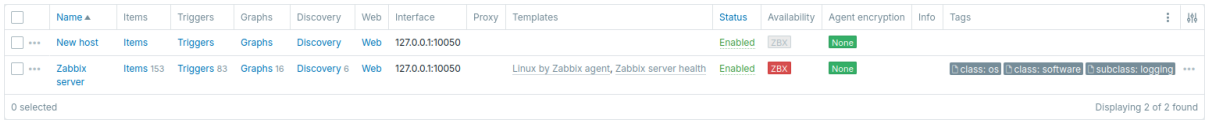
> **Note:**
> Toutes les permissions d'accès sont attribuées aux groupes d'hôtes, et non aux hôtes individuels. C'est pourquoi un hôte doit appartenir à au moins un groupe.

Interfaces : adresse IP

- Bien qu'il ne soit pas techniquement obligatoire, une interface d'hôte est nécessaire pour collecter certaines métriques. Pour utiliser les contrôles passifs de l'agent Zabbix, indiquez l'adresse IP ou le DNS de l'agent dans ce champ. Notez que vous devez également indiquer l'adresse IP ou le DNS du serveur Zabbix dans la directive « Server » du fichier de configuration de l'agent Zabbix. Si l'agent Zabbix et le serveur Zabbix sont installés sur la même machine, vous devez spécifier la même adresse IP ou le même DNS aux deux endroits.

Les autres options peuvent rester inchangées et conserver leurs valeurs par défaut.

Lorsque vous avez terminé, cliquez sur *Ajouter*. Votre nouvel hôte doit apparaître dans la liste des hôtes.



La colonne *Disponibilité* contient des indicateurs de disponibilité de l'hôte pour chaque interface. Lorsqu'une interface d'agent Zabbix est définie, nous pouvons utiliser l'icône de disponibilité de l'agent (étiquetée *ZBX*) pour déterminer la disponibilité de l'hôte :

- ZBX - l'état de l'hôte n'a pas été établi ; aucun contrôle de métrique n'a encore été effectué.
- ZBX - l'hôte est disponible, un contrôle de métrique a réussi.
- ZBX - l'hôte n'est pas disponible, un contrôle de métrique a échoué (placez le curseur de votre souris sur l'icône pour afficher le message d'erreur). Il peut y avoir un problème de communication, éventuellement dû à des identifiants d'interface incorrects. Vérifiez que le serveur Zabbix est en cours d'exécution et essayez également d'actualiser la page ultérieurement.

Nouvel élément

Vue d'ensemble

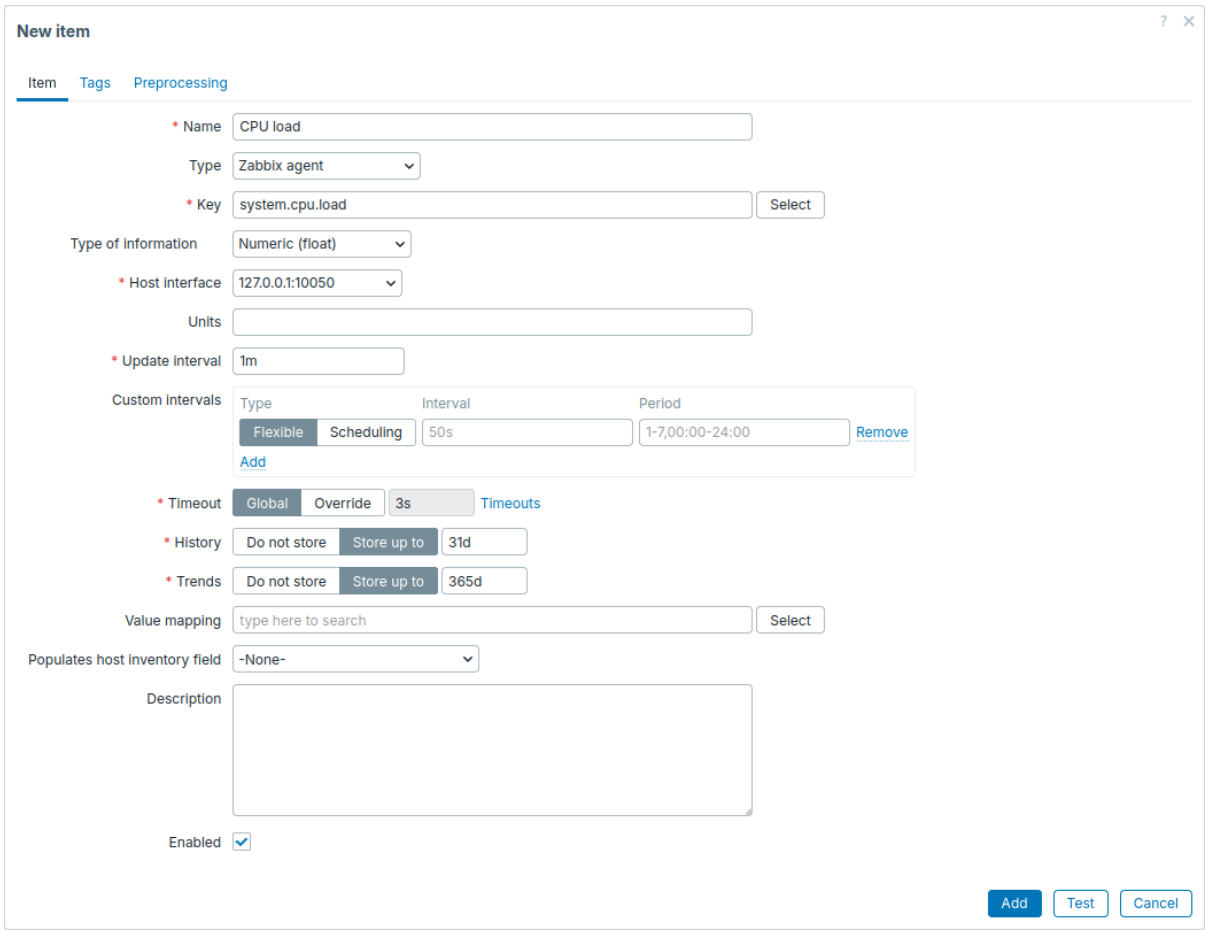
Dans cette section, vous apprendrez à configurer un élément.

Les éléments sont la base de la collecte de données dans Zabbix. Sans éléments, il n'y a pas de données, car seul un élément définit une métrique unique ou le type de données à collecter depuis un hôte.

Ajout d'un élément

Tous les éléments sont regroupés autour des hôtes. C'est pourquoi, pour configurer un élément d'exemple, nous allons dans *Collecte de données* > *Hôtes* et trouvons le "New host" que nous avons créé.

Cliquez sur le lien *Éléments* dans la ligne de "New host", puis cliquez sur *Créer un élément*. Un formulaire de configuration d'élément s'ouvre.



Tous les champs de saisie obligatoires sont marqués d'un astérisque rouge.

Pour notre élément d'exemple, les informations essentielles à saisir sont les suivantes :

Nom

- Saisissez *CPU load* comme valeur. Il s'agit du nom de l'élément affiché dans les listes et ailleurs.

Clé

- Saisissez manuellement *system.cpu.load* comme valeur. Il s'agit du nom technique d'un élément qui identifie le type d'information à collecter. Cette clé particulière n'est qu'une des clés prédéfinies fournies avec l'agent Zabbix.

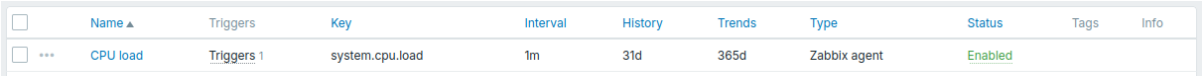
Type d'information

- Cet attribut définit le format des données attendues. Pour la clé *system.cpu.load*, ce champ sera automatiquement défini sur *Numérique (flottant)*.

Note:
Vous pouvez également réduire le nombre de jours pendant lesquels l'historique de l'élément sera conservé, à 7 ou 14. C'est une bonne pratique pour éviter à la base de données de conserver un grand nombre de valeurs historiques.

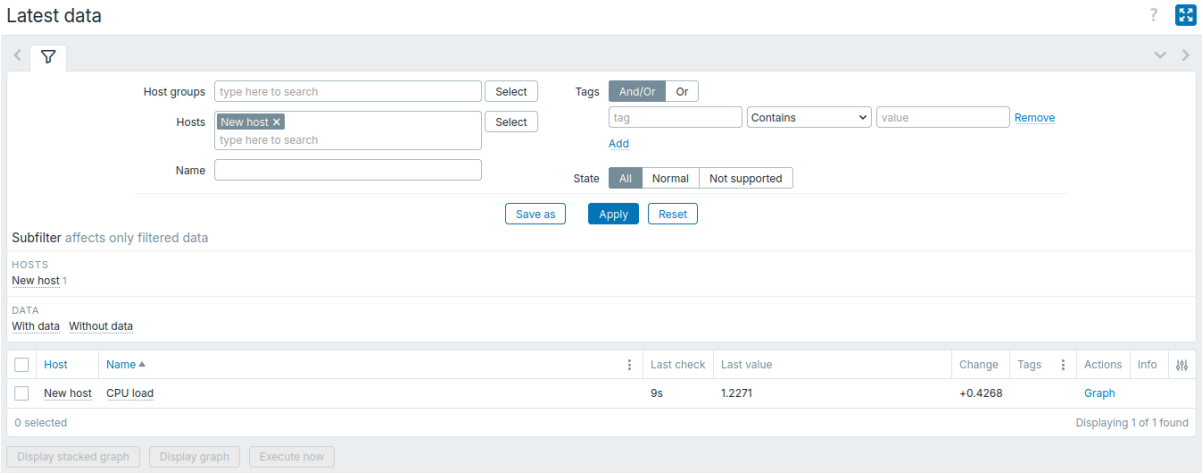
Les autres options peuvent être laissées inchangées, avec leurs valeurs par défaut.

Une fois terminé, cliquez sur *Ajouter*. Le nouvel élément devrait apparaître dans la liste des éléments.



Affichage des données

Une fois un élément défini, vous pouvez vous demander s'il collecte réellement des données. Pour cela, allez dans *Monitoring* > *Latest data*, sélectionnez 'New host' dans le filtre, puis cliquez sur *Apply*.



Cela dit, il peut s'écouler jusqu'à 60 secondes avant que les premières données n'arrivent. Par défaut, c'est la fréquence à laquelle le serveur lit les modifications de configuration et prend en compte les nouveaux éléments à exécuter.

Si aucune valeur n'apparaît dans la colonne 'Change', il est possible qu'une seule valeur ait été reçue jusqu'à présent. Attendez 30 secondes pour qu'une autre valeur arrive.

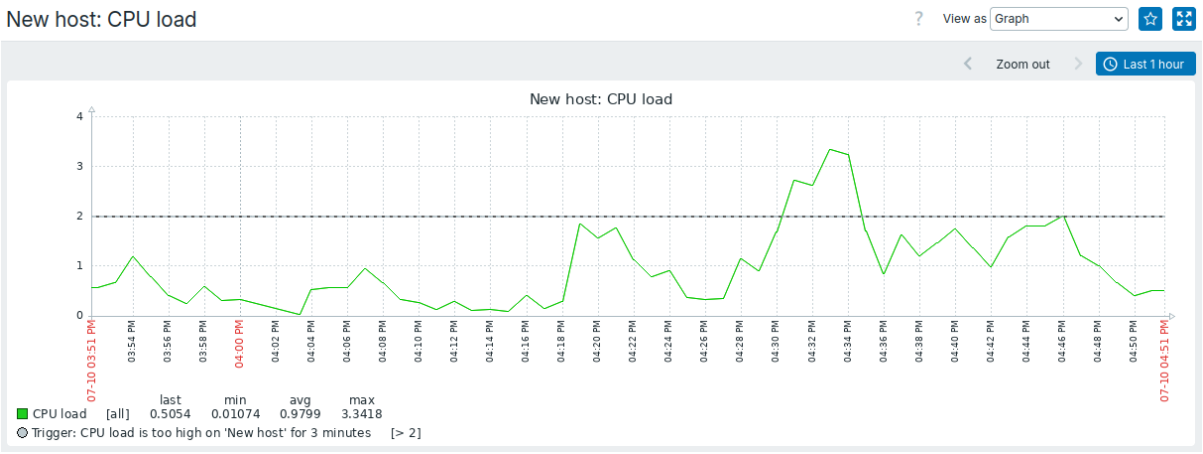
Si vous ne voyez pas d'informations sur l'élément comme dans la capture d'écran, assurez-vous que :

- vous avez renseigné les champs 'Key' et 'Type of information' de l'élément exactement comme dans la capture d'écran ;
- l'agent et le serveur sont tous deux en cours d'exécution ;
- l'état de l'hôte est 'Monitored' et son icône de disponibilité est verte ;
- l'hôte sélectionné dans le filtre d'hôtes est correct ;
- l'élément est activé.

Graphiques

Une fois que l'élément fonctionne depuis un certain temps, il peut être utile de visualiser les données. Des graphiques simples sont disponibles pour tout élément numérique surveillé, sans configuration supplémentaire. Ces graphiques sont générés lors de l'exécution.

Pour afficher le graphique, accédez à *Surveillance* > *Dernières données* et cliquez sur le lien *Graphique* à côté de l'élément.



Nouveau déclencheur

Vue d'ensemble

Dans cette section, vous apprendrez à configurer un déclencheur.

Les éléments ne collectent que des données. Pour évaluer automatiquement les données entrantes, nous devons définir des déclencheurs. Un déclencheur contient une expression qui définit un seuil correspondant à un niveau acceptable pour les données.

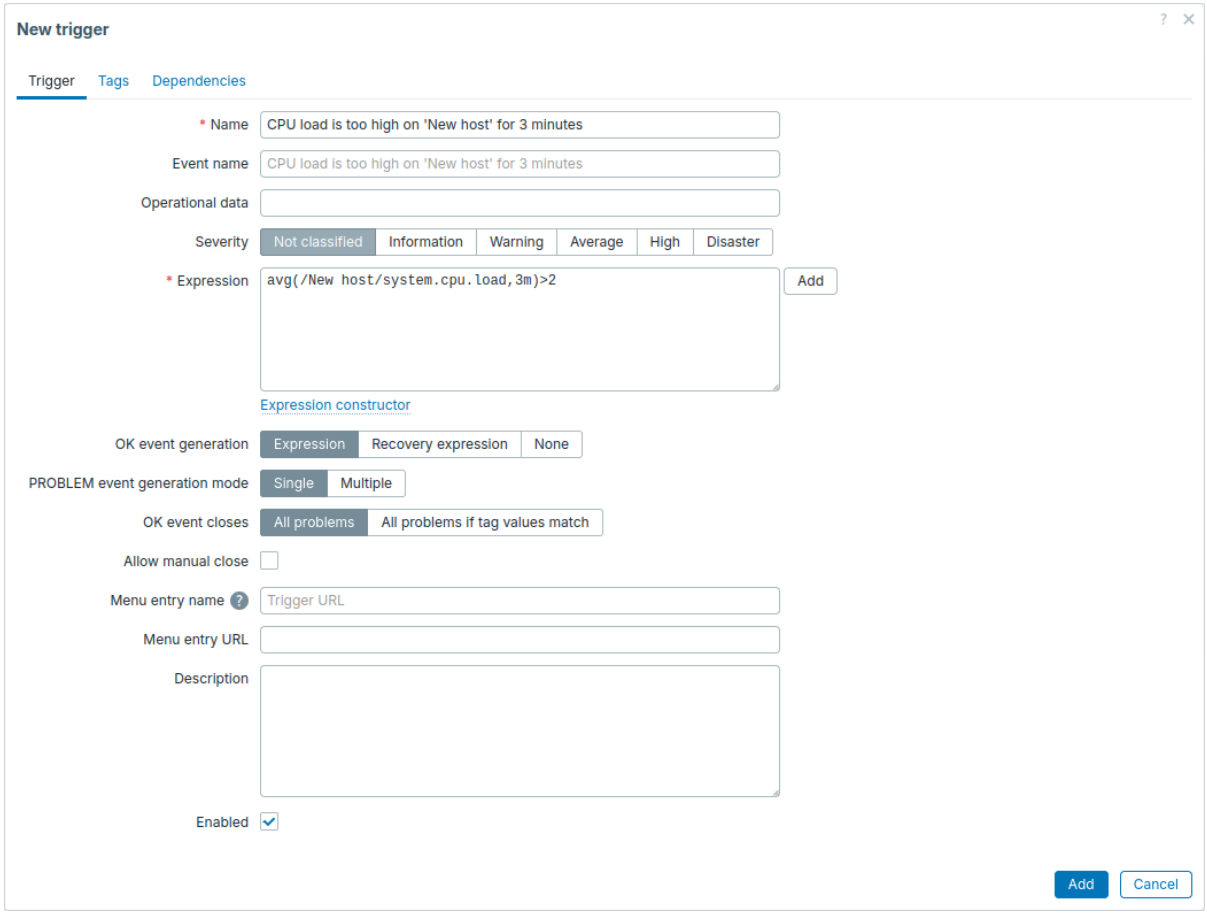
Si ce niveau est dépassé par les données entrantes, un déclencheur se "déclenche" ou passe à l'état 'Problem' - indiquant que quelque chose s'est produit et peut nécessiter une attention. Si le niveau redevient acceptable, le déclencheur revient à l'état 'Ok'.

Ajout d'un déclencheur

Pour configurer un déclencheur pour notre élément :

1. Accédez à *Data collection* > *Hosts*.
2. Cliquez sur le lien *Triggers* dans la ligne de 'New host'.
3. Cliquez sur *Create trigger*.

Un formulaire de configuration du déclencheur s'ouvre.



Pour notre déclencheur, les informations essentielles à saisir ici sont :

Name

- Saisissez *CPU load is too high on 'New host' for 3 minutes* comme valeur. Ce sera le nom du déclencheur affiché dans les listes et ailleurs.

Expression

- Saisissez : `avg(/New host/system.cpu.load,3m)>2`

Il s'agit de l'expression du déclencheur. Assurez-vous que l'expression est saisie correctement, jusqu'au dernier symbole. La clé de l'élément ici (`system.cpu.load`) est utilisée pour faire référence à l'élément. Cette expression particulière indique essentiellement que le seuil de problème est dépassé lorsque la valeur moyenne de la charge CPU sur 3 minutes est supérieure à 2. Vous pouvez en savoir plus sur la syntaxe des expressions de déclencheur.

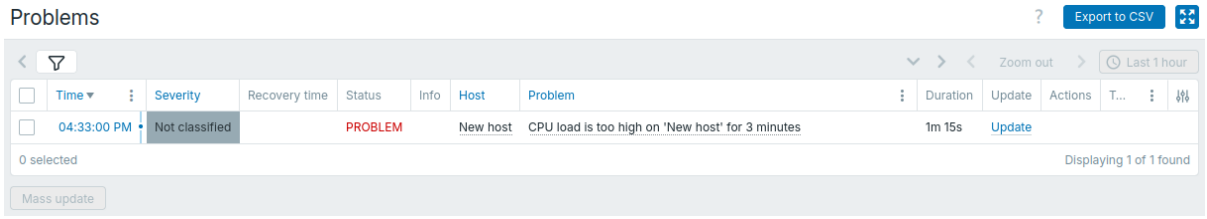
Les autres options peuvent être laissées inchangées, avec leurs valeurs par défaut.

Une fois terminé, cliquez sur *Add*. Le nouveau déclencheur devrait apparaître dans la liste des déclencheurs.

Affichage de l'état du déclencheur

Une fois un déclencheur défini, vous pouvez souhaiter voir son état.

Si la charge CPU a dépassé le seuil que vous avez défini dans le déclencheur, le problème s'affichera dans *Monitoring* > *Problems*.



Le clignotement dans la colonne d'état indique un changement récent de l'état du déclencheur, survenu au cours des 30 dernières minutes.

Réception d'une notification de problème

Vue d’ensemble

Dans cette section, vous apprendrez à configurer les alertes sous forme de notifications dans Zabbix.

Avec des éléments qui collectent des données et des déclencheurs conçus pour se ”déclencher” lors de situations problématiques, il serait également utile de mettre en place un mécanisme d’alerte afin de notifier les événements importants, même lorsque l’interface Zabbix n’est pas consultée activement.

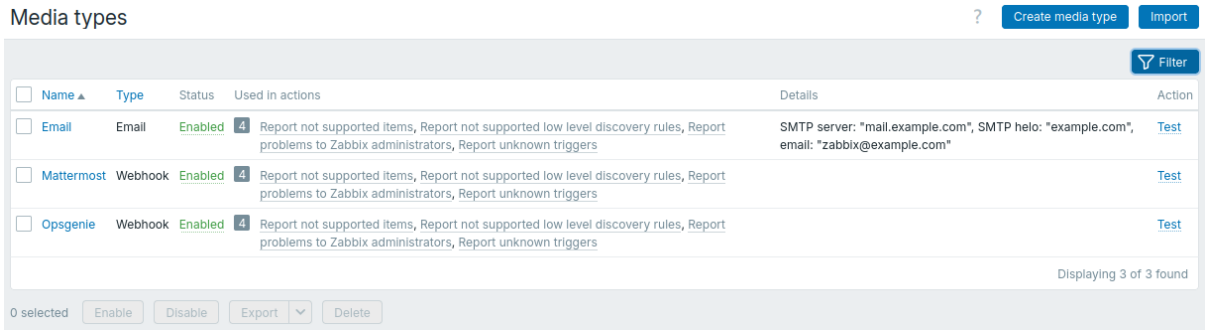
C’est le rôle des notifications.

L’e-mail étant la méthode de diffusion la plus courante pour les notifications de problème, nous allons apprendre à configurer une notification par e-mail.

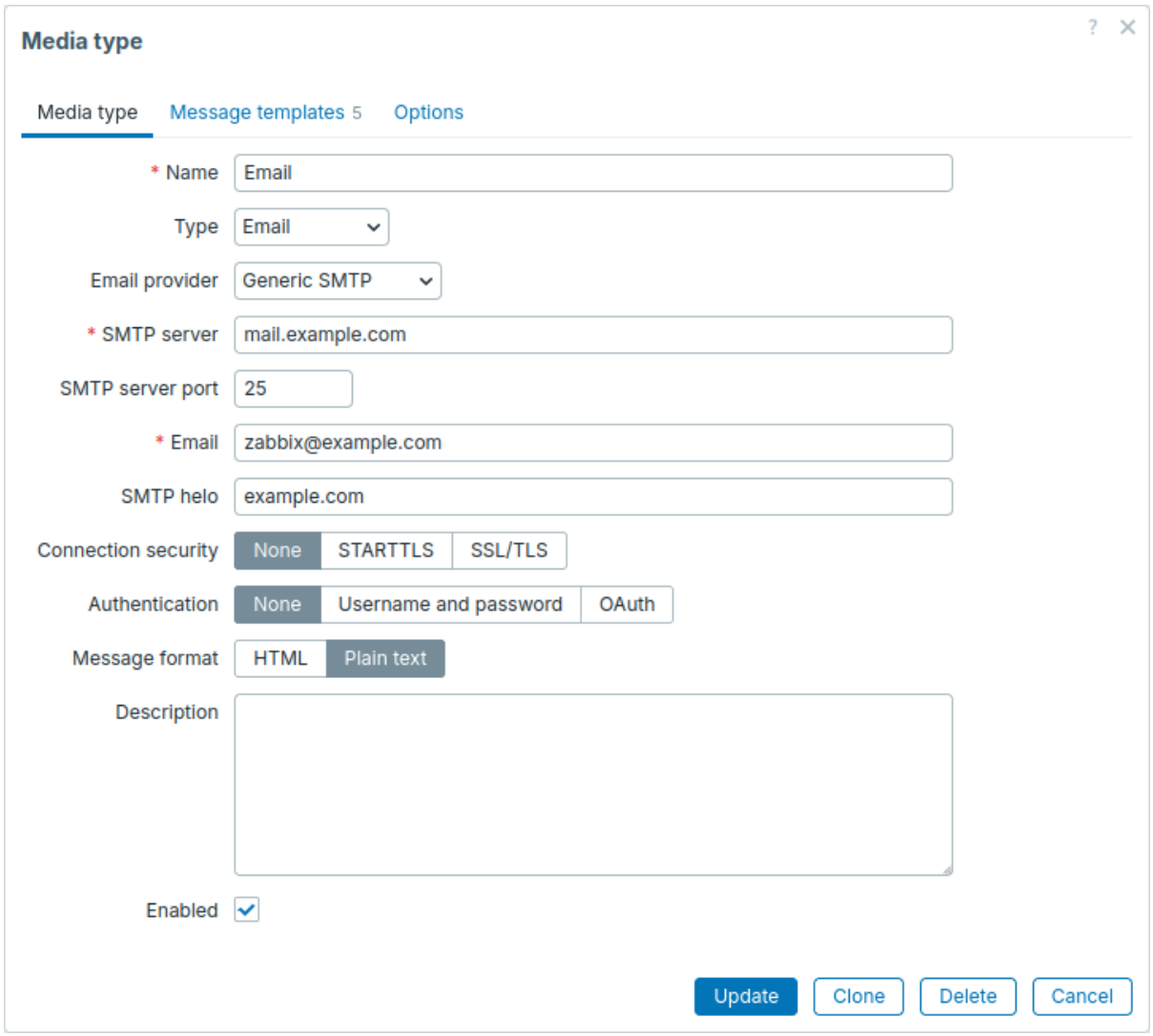
Paramètres de messagerie

Au départ, Zabbix propose plusieurs modes de livraison de notifications prédéfinis. Email en fait partie.

Pour configurer les paramètres de messagerie, allez dans *Alertes* > *Types de média* et cliquez sur *Email* dans la liste des types de média prédéfinis.



Un formulaire de configuration des paramètres de messagerie s’ouvre.



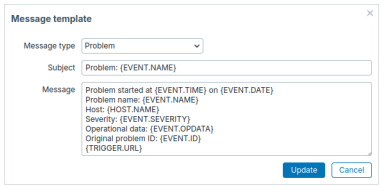
Tous les champs obligatoires sont marqués d'un astérisque rouge.

Dans l'onglet *Type de média*, définissez les valeurs de *SMTP server*, *SMTP helo* et *Email* selon votre environnement.

> **Note:**
> La valeur du champ *Email* sera utilisée comme adresse d'expéditeur pour les notifications envoyées par Zabbix.

Ensuite, il faut définir le contenu du message de problème. Ce contenu est défini au moyen d'un modèle de message, configuré dans l'onglet *Message templates*.

Cliquez sur *Add* pour créer un modèle de message, puis sélectionnez *Problem* comme type de message.

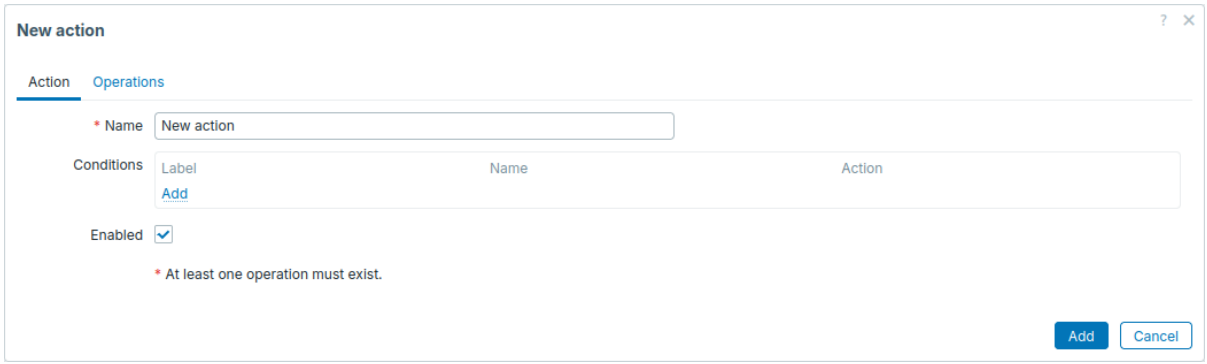


Cliquez sur *Add* lorsque vous avez terminé, puis enregistrez le formulaire.

Vous avez maintenant configuré *Email* comme type de média fonctionnel. Le type de média doit également être associé aux utilisateurs en définissant des adresses de livraison spécifiques (comme nous l'avons fait lors de la configuration d'un nouvel utilisateur), sinon il ne sera pas utilisé.

Nouvelle action

L'envoi de notifications est l'une des fonctions des actions dans Zabbix. Par conséquent, pour configurer une notification, allez dans *Alertes* > *Actions* > *Actions de déclencheur* et cliquez sur *Créer une action*.

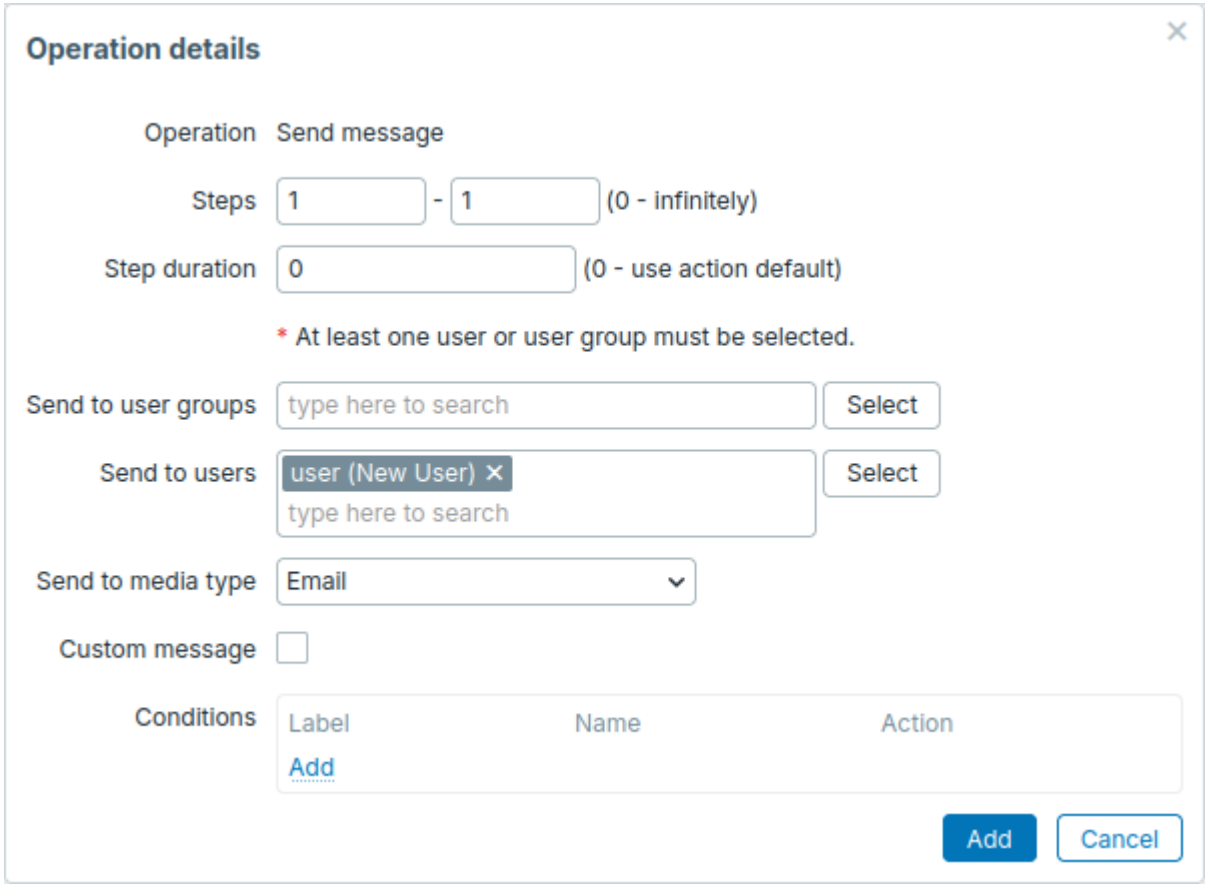


Tous les champs obligatoires sont marqués d'un astérisque rouge.

Dans ce formulaire, saisissez un nom pour l'action.

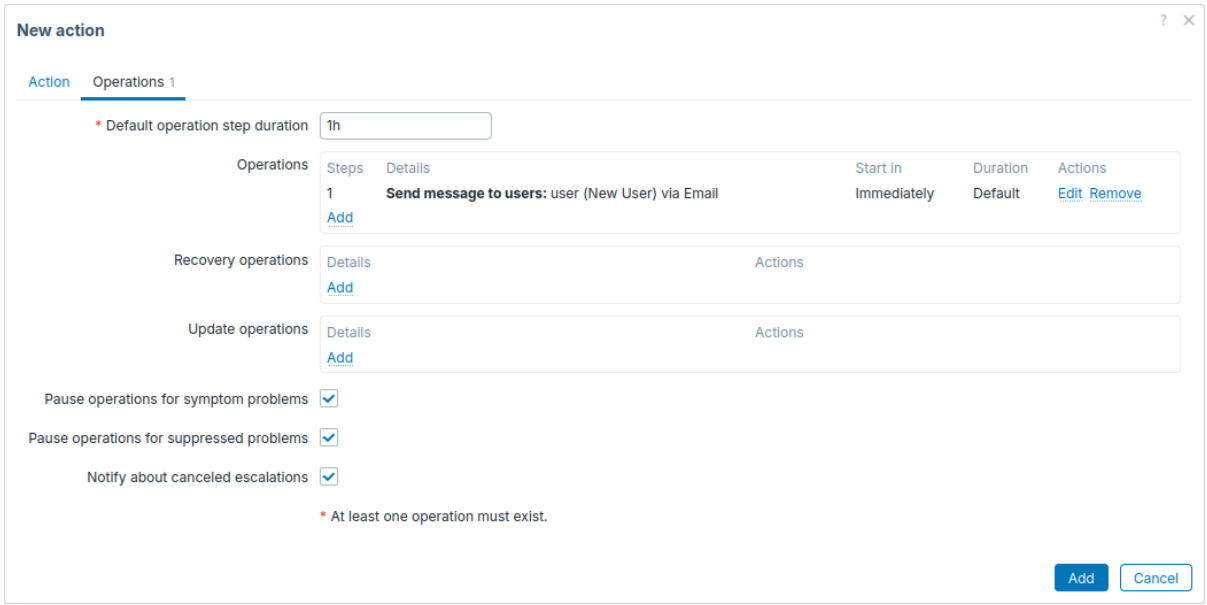
Dans le cas le plus simple, si nous n'ajoutons aucune condition plus spécifique, l'action sera exécutée lors de tout changement de déclencheur de 'Ok' à 'Problem'.

Nous devons encore définir ce que l'action doit faire - cela se configure dans l'onglet *Opérations*. Cliquez sur *Ajouter* dans le bloc *Opérations*, ce qui ouvre un nouveau formulaire d'opération.



Tous les champs obligatoires sont marqués d'un astérisque rouge.

Ici, cliquez sur *Sélectionner* dans le bloc *Envoyer aux utilisateurs* et sélectionnez l'utilisateur ('user') que nous avons défini. Sélectionnez "Email" comme valeur de *Envoyer vers le type de média*. Une fois cela fait, cliquez sur *Ajouter*, et l'opération doit être ajoutée :



C'est tout pour une configuration simple d'action, alors cliquez sur *Ajouter* dans le formulaire de l'action.

Réception d'une notification

Maintenant que l'envoi des notifications est configuré, il serait intéressant d'en recevoir une. Pour cela, nous pouvons volontairement augmenter la charge sur notre hôte, afin que notre déclencheur se "déclenche" et que nous recevions une notification de problème.

Ouvrez la console sur votre hôte et exécutez :

```
cat /dev/urandom | md5sum
```

Vous pouvez exécuter un ou plusieurs de ces processus.

Allez ensuite dans *Monitoring* > *Latest data* et observez comment les valeurs de 'CPU Load' ont augmenté. N'oubliez pas que, pour que notre déclencheur se "déclenche", la valeur de 'CPU Load' doit dépasser '2' pendant 3 minutes consécutives. Une fois que c'est le cas :

- dans *Monitoring* > *Problems*, vous devriez voir le déclencheur avec un état `Problem` clignotant ;
- vous devriez recevoir une notification de problème dans votre e-mail.

Attention:
Si les notifications ne fonctionnent pas :

- vérifiez une fois de plus que les paramètres de messagerie et l'action ont été correctement configurés
- assurez-vous que l'utilisateur que vous avez créé dispose au minimum des autorisations de lecture sur l'hôte qui a généré l'événement, comme indiqué à l'étape *Ajout d'un utilisateur*. L'utilisateur, faisant partie du groupe d'utilisateurs 'Zabbix administrators', doit avoir au minimum un accès en lecture au groupe d'hôtes 'Linux servers' auquel appartient notre hôte.
- Vous pouvez également consulter le journal des actions en allant dans *Reports* > *Action log*.

Nouveau modèle

Présentation

Dans cette section, vous apprendrez à configurer un modèle.

Précédemment, nous avons appris à configurer un élément et un déclencheur, ainsi qu'à obtenir une notification de problème pour le hôte.

Bien que toutes ces étapes offrent une grande flexibilité en elles-mêmes, elles peuvent sembler nombreuses s'il faut les effectuer pour, par exemple, un millier de hôtes. Une certaine automatisation serait utile.

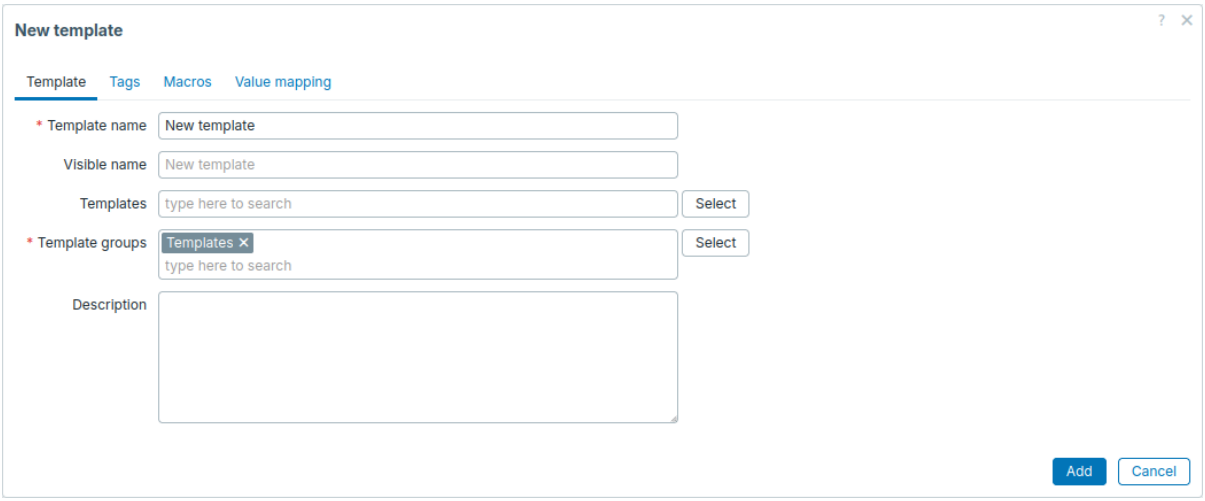
C'est là que les modèles interviennent.

Les modèles vous permettent de regrouper des éléments, des déclencheurs et d'autres entités utiles afin de pouvoir les réutiliser encore et encore en les appliquant aux hôtes en une seule étape.

Lorsqu'un modèle est lié à un hôte, celui-ci hérite de toutes les entités du modèle. Ainsi, un ensemble de vérifications préparé à l'avance peut être appliqué très rapidement.

Ajout d'un modèle

Pour commencer à travailler avec des modèles, nous devons d'abord en créer un. Pour cela, dans *Collecte de données* > *Modèles*, cliquez sur *Créer un modèle*. Cela affichera un formulaire de configuration du modèle.



Tous les champs obligatoires sont marqués d'un astérisque rouge.

Les paramètres requis à saisir ici sont les suivants:

Nom du modèle

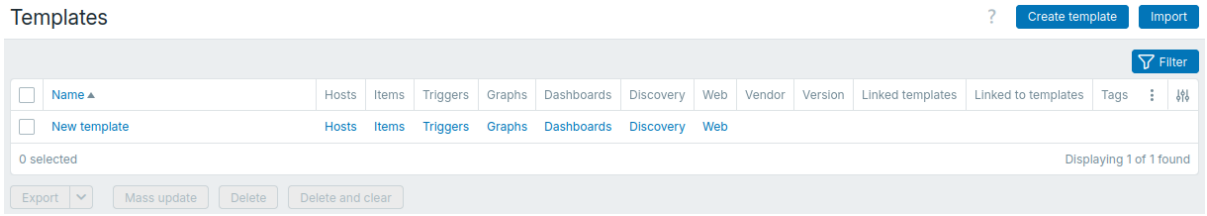
- Saisissez un nom de modèle. Les caractères alphanumériques, les espaces et les underscores sont autorisés.

Groupes de modèles

- Sélectionnez un ou plusieurs groupes en cliquant sur le bouton *Sélectionner*. Le modèle doit appartenir à un groupe.

Note:
Les permissions d'accès aux groupes de modèles sont attribuées dans la configuration du groupe d'utilisateurs, dans l'onglet **Permissions des modèles**, de la même manière que les permissions des hôtes. Toutes les permissions d'accès sont attribuées à des groupes, et non à des modèles individuels, c'est pourquoi l'inclusion du modèle dans au moins un groupe est obligatoire.

Une fois terminé, cliquez sur *Ajouter*. Votre nouveau modèle devrait être visible dans la liste des modèles. Vous pouvez également utiliser le filtre pour trouver votre modèle.

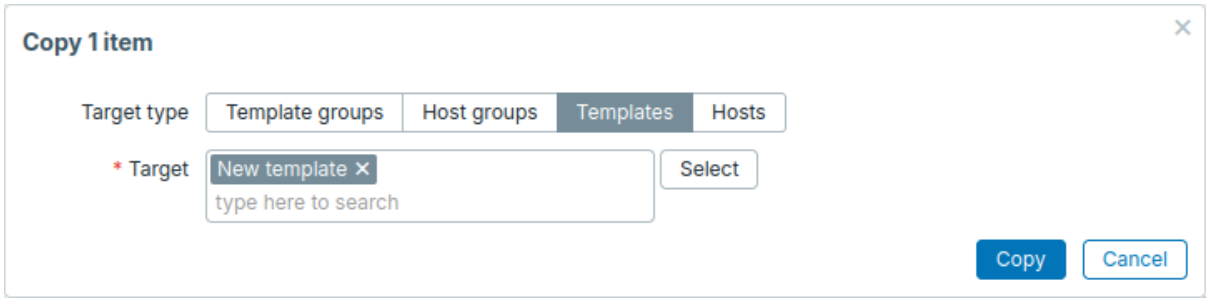


Comme vous pouvez le voir, le modèle est bien présent, mais il ne contient rien - aucun élément, déclencheur ni autre entité.

Ajout d'un élément à un modèle

Pour ajouter un élément au modèle :

1. Accédez à *Collecte de données* > *Hôtes*.
2. Repérez « New host » dans la liste et cliquez sur le lien *Éléments*.
3. Cochez la case de l'élément « CPU Load » dans la liste.
4. Cliquez sur *Copier* sous la liste.
5. Sélectionnez l'onglet *Modèles*.
6. Sélectionnez le modèle vers lequel copier l'élément.



Tous les champs de saisie obligatoires sont marqués d'un astérisque rouge. 7. Cliquez sur *Copier*.

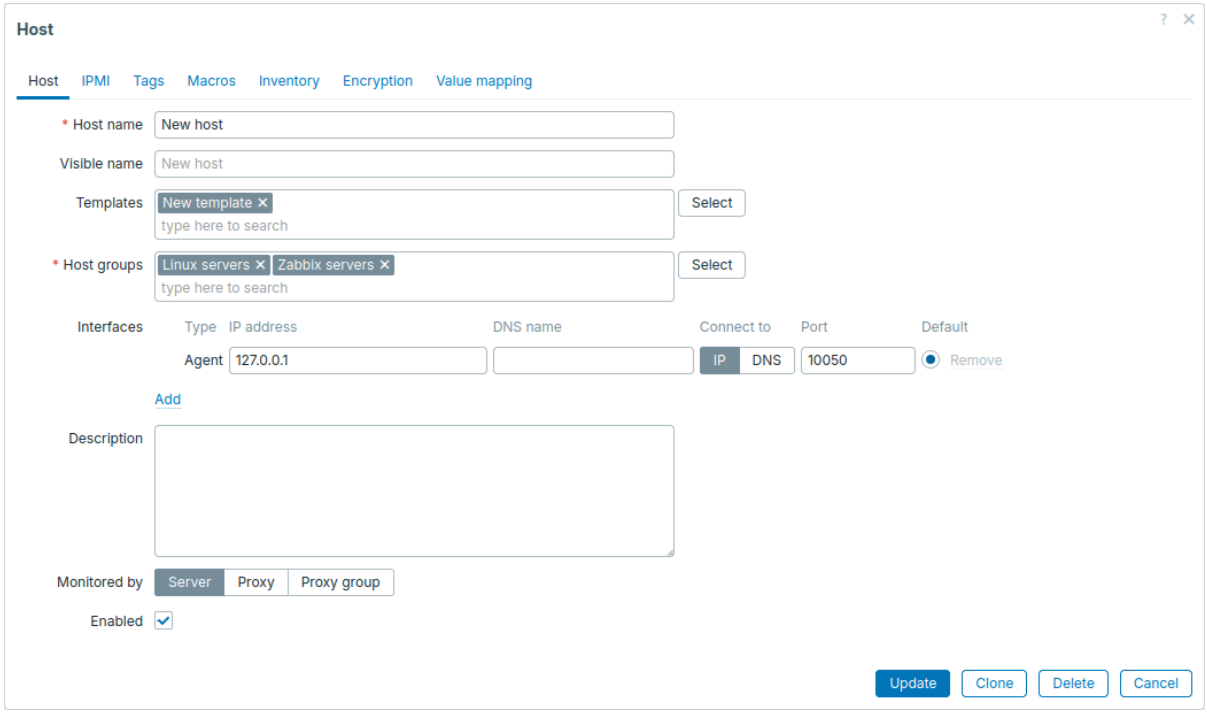
Si vous accédez maintenant à *Collecte de données* > *Modèles*, « New template » devrait contenir un nouvel élément.

Nous nous limiterons pour l'instant à un seul élément, mais vous pouvez ajouter de la même manière d'autres éléments, déclencheurs ou entités au modèle, jusqu'à obtenir un ensemble d'entités suffisamment complet pour un objectif donné (surveillance du système d'exploitation, surveillance d'une application spécifique).

Liaison d'un modèle à un hôte

Une fois le modèle prêt, il ne reste plus qu'à l'ajouter à un hôte. Pour ce faire :

1. Accédez à *Collecte de données* > *Hôtes*.
2. Cliquez sur « Nouvel hôte » pour ouvrir son formulaire de configuration, puis recherchez le champ *Modèles*.
3. Commencez à saisir *Nouveau modèle* dans le champ *Modèles*. Le nom du modèle nouvellement créé devrait apparaître dans la liste déroulante.
4. Faites défiler la liste pour le sélectionner. Vérifiez qu'il apparaît dans le champ *Modèles*.



5. Cliquez sur *Mettre à jour* dans le formulaire pour enregistrer les modifications.

Le modèle est maintenant ajouté à l'hôte, avec toutes les entités qu'il contient.

Il peut ainsi être appliqué à n'importe quel autre hôte. Toute modification apportée aux éléments, déclencheurs et autres entités au niveau du modèle sera propagée aux hôtes auxquels le modèle est lié.

Liaison de modèles prédéfinis aux hôtes

Comme vous l'avez peut-être remarqué, Zabbix est fourni avec un ensemble de modèles prédéfinis pour divers systèmes d'exploitation, périphériques et applications. Pour commencer la supervision très rapidement, vous pouvez associer à un hôte le modèle approprié, mais sachez que ces modèles doivent être ajustés à votre environnement. Certaines vérifications peuvent ne pas être nécessaires, et les intervalles d'interrogation peuvent être beaucoup trop fréquents.

De plus amples informations sur les modèles sont disponibles.

Surveillance du serveur web Apache

Pour commencer à surveiller le serveur web Apache, suivez cette procédure :

1. Assurez-vous d'avoir accès aux statistiques Apache.
2. Appliquez le modèle de surveillance Apache.
3. Consultez les dernières données.

Pour utiliser ce guide, vous devez disposer d'un serveur Zabbix et de l'interface installée. Vous devez être connecté à Zabbix en tant qu'utilisateur *Admin* par défaut.

Accès aux statistiques Apache

L'objectif de cette étape est de s'assurer que les statistiques du serveur web Apache peuvent être lues par Zabbix.

Dans Apache, les informations en temps réel sur les performances et l'activité du serveur sont fournies par le module intégré `mod_status`. Lorsqu'il est activé, il crée une page accessible via le web (généralement à `/server-status`) affichant les données de performances et d'activité.

Pour vérifier que `mod_status` est activé pour le serveur web Apache, exécutez :

Sur Debian/Ubuntu	Sur les systèmes basés sur RHEL
`apache2ctl -M | grep status`	`httpd -M | grep status`

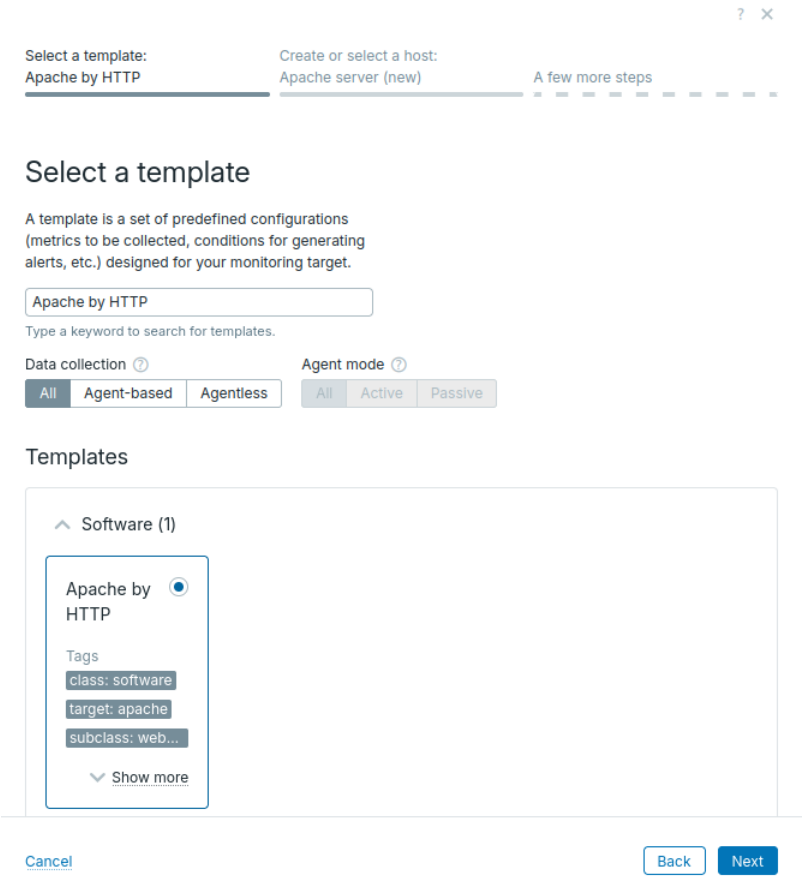
Si vous voyez "`status_module (shared)`" dans la sortie, alors le module de statut est déjà activé. Sinon, vous devez activer le module de statut.

Appliquer le modèle de supervision Apache

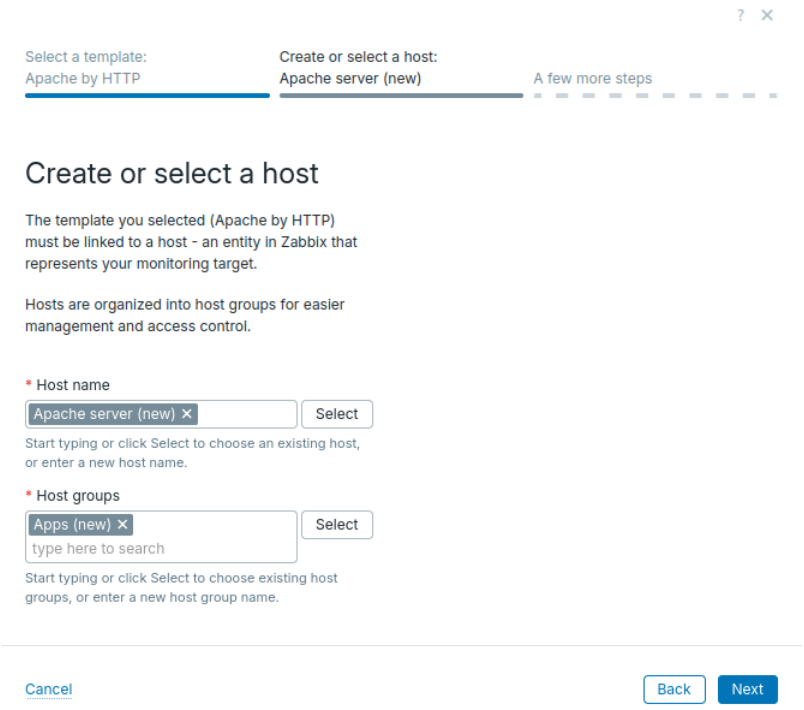
Dans Zabbix, ce modèle est prêt à l'emploi. Il contient des éléments préconfigurés prêts pour la collecte de données, la visualisation et l'analyse.

Pour appliquer le modèle :

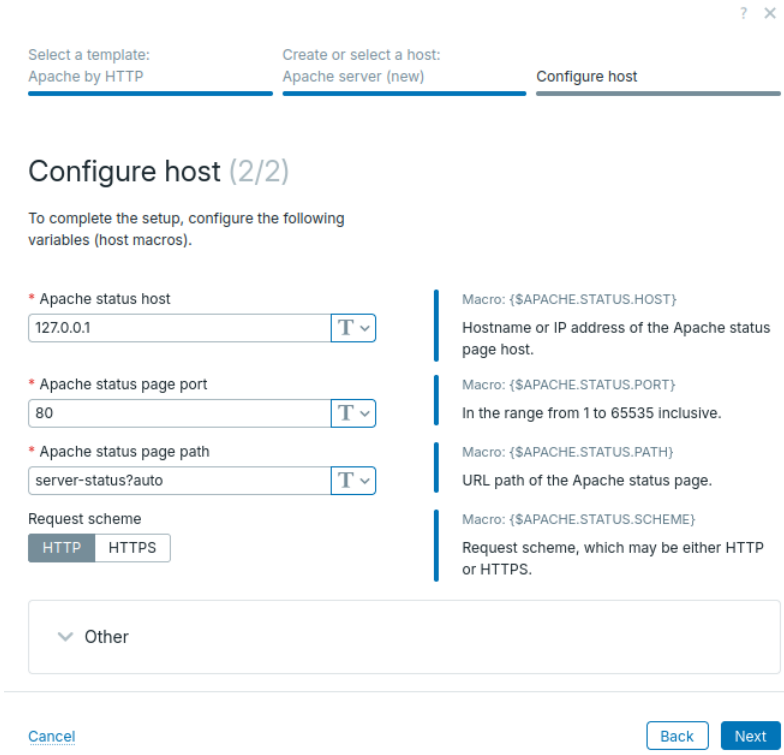
1. Démarrez l'Assistant d'hôte dans *Collecte de données* > *Hôtes*.
2. Sélectionnez le modèle *Apache by HTTP*.



3. Créez un hôte Apache et ajoutez-lui un groupe. Cet hôte virtuel contiendra les données liées à Apache.



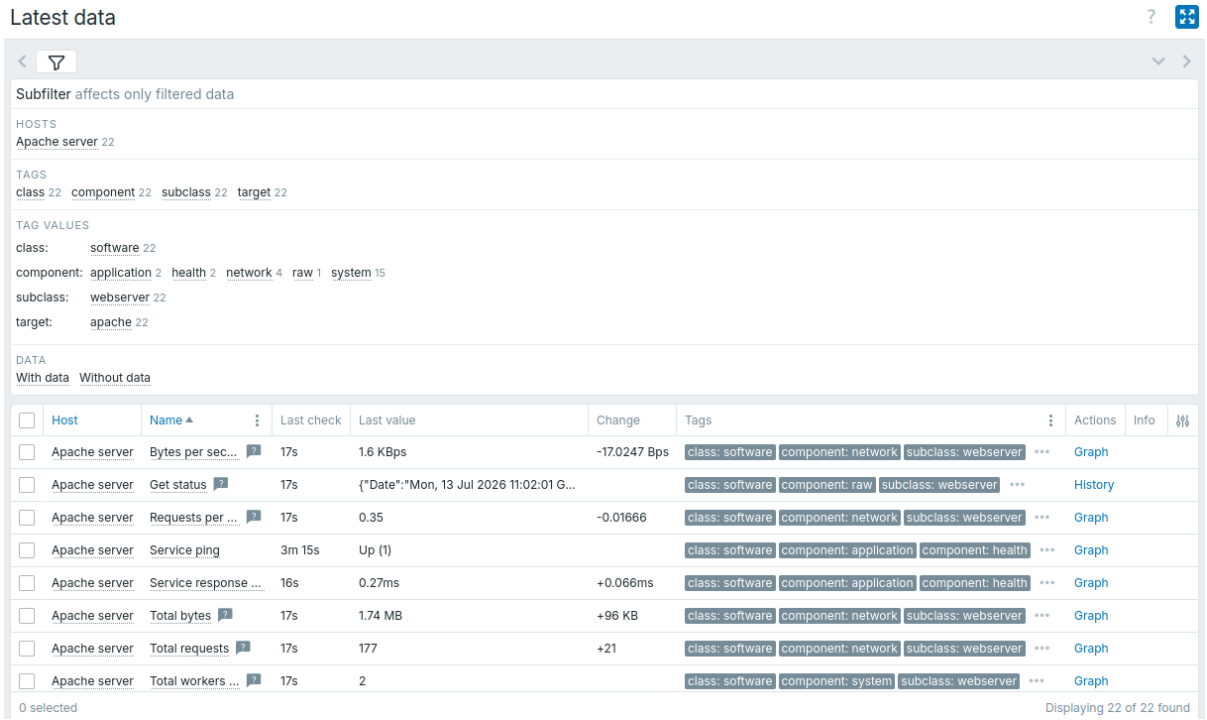
4. Configurez l'hôte. Important : saisissez l'adresse du serveur web Apache (127.0.0.1 sur une machine locale, adresse IP sur une machine distante).



Vous pouvez laisser les autres paramètres avec leurs valeurs par défaut.

5. Pour terminer l'Assistant d'hôte, cliquez sur *Créer* puis sur *Terminer*.

Consultez les dernières données



Félicitations, la supervision du serveur web Apache a démarré !

Zabbix fournit des options de visualisation et envoie des alertes si des problèmes surviennent. Pour configurer les alertes par e-mail, consultez Réception des notifications de problème.

Supervision d'un serveur Linux

Pour commencer à superviser un serveur Linux, suivez cette procédure :

1. Appliquez le modèle de supervision Linux.
2. Installez l'agent Zabbix sur le serveur.

3. Consultez les dernières données.

Pour commencer à superviser un serveur Linux non local, vous devez connaître son adresse IP, ainsi que l'adresse IP du serveur Zabbix.

Un agent Zabbix sera utilisé sur le serveur Linux pour le superviser. Les instructions d'installation de l'agent seront fournies au cours de la procédure.

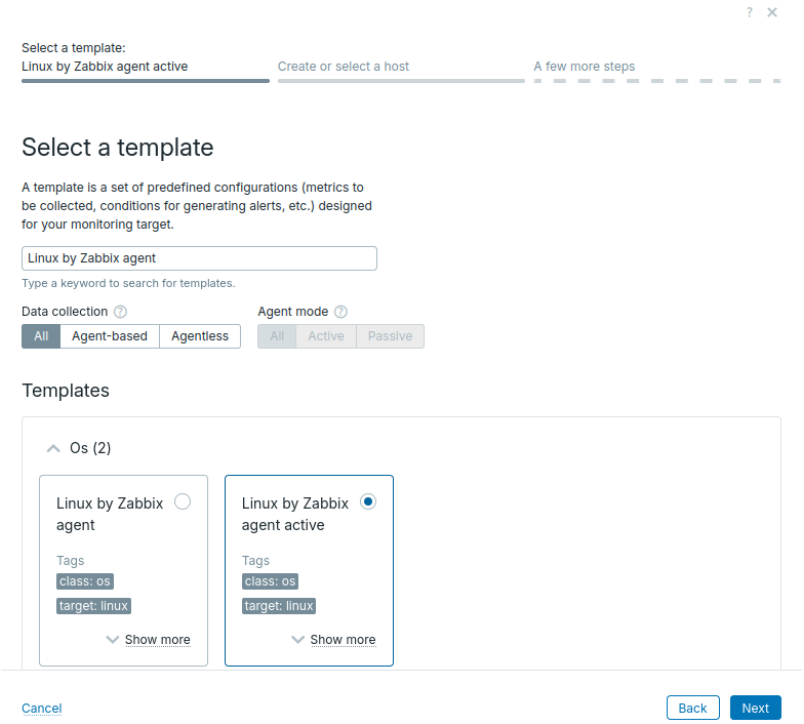
Pour utiliser ce guide, vous devez avoir Zabbix serveur et l'interface installés. Vous devez être connecté à Zabbix en tant qu'utilisateur *Admin* par défaut.

Appliquer le modèle de supervision Linux

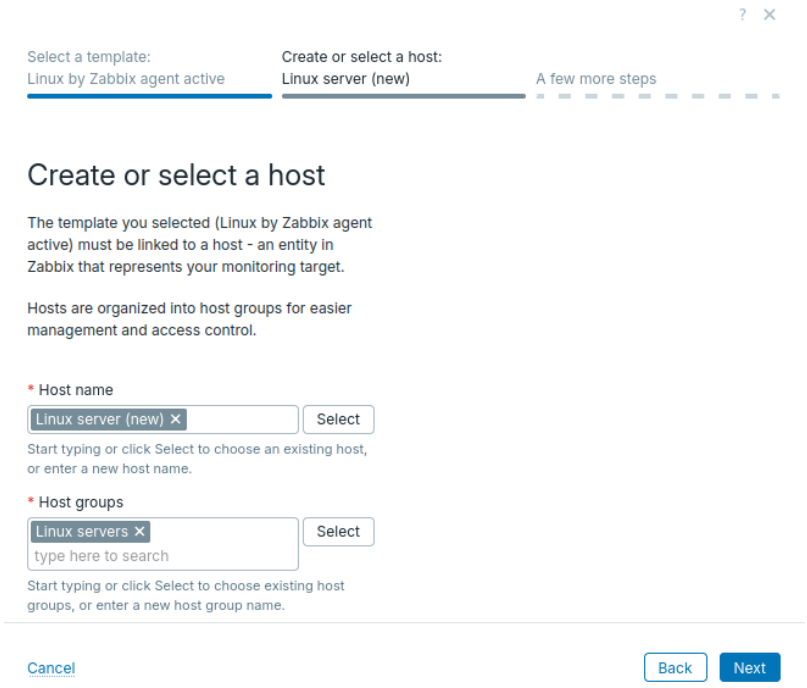
Dans Zabbix, ce modèle est prêt à l'emploi. Il contient des éléments préconfigurés, prêts pour la collecte de données, la visualisation et l'analyse.

Pour appliquer le modèle :

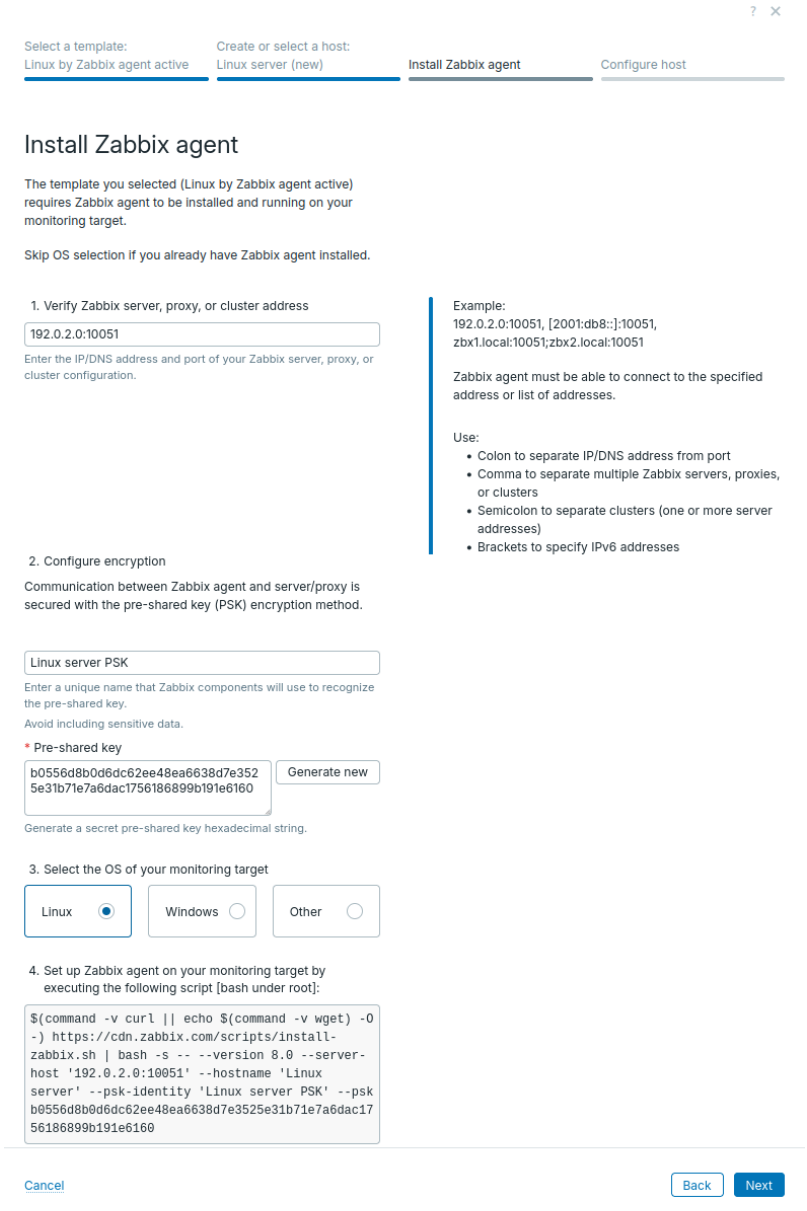
1. Démarrez l'Assistant d'hôte dans *Collecte de données* > *Hôtes*.
2. Sélectionnez le modèle *Linux by Zabbix agent active*.



3. Créez un hôte pour le serveur Linux et ajoutez-lui un groupe. Cet hôte virtuel contiendra toutes les données liées à la supervision du serveur Linux.



4. Configurez et installez l'agent.



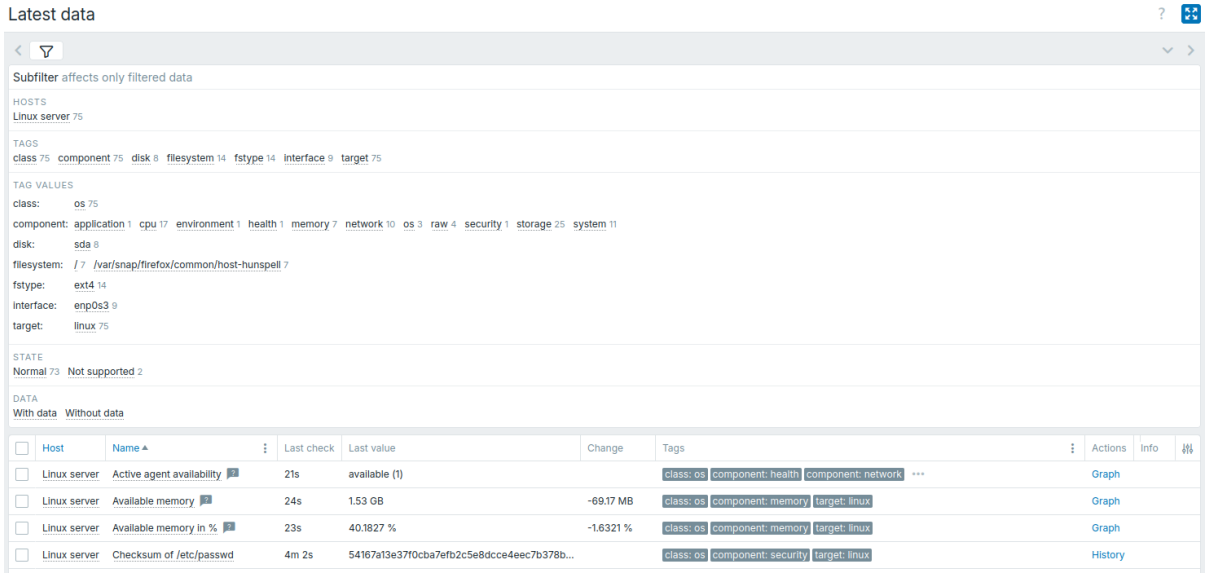
Pour installer l'agent, exécutez la commande fournie sur le serveur Linux. Elle installera l'agent et configurera la connexion de

l'agent au serveur Zabbix avec chiffrement.
 Si l'agent est déjà installé, vous pouvez ignorer cette étape (cliquez sur *Suivant*). Cependant, il vous sera demandé de fournir l'identifiant de la clé prépartagée et la clé prépartagée pour vous connecter à l'agent.

Vous pouvez laisser les autres paramètres sur leurs valeurs par défaut.

5. Terminez la configuration de l'hôte : cliquez sur *Créer* puis sur *Terminer*.

Consultez les dernières données



Félicitations, la supervision du serveur Linux a été lancée !

Zabbix propose des options de visualisation et envoie des alertes si des problèmes surviennent. Pour configurer les alertes par e-mail, consultez Réception des notifications de problème.

Surveillance du serveur MySQL

Pour commencer à surveiller un serveur MySQL, suivez cette procédure :

1. Créez un utilisateur MySQL avec des droits limités.
2. Appliquez le modèle de surveillance MySQL.
3. Consultez les dernières données.

Pour commencer à surveiller un serveur MySQL, vous devez connaître son DSN et l'adresse IP du serveur Zabbix.

Un agent Zabbix sera utilisé pour surveiller le serveur MySQL. Les instructions d'installation de l'agent seront fournies au cours de la procédure.

Pour utiliser ce guide, vous devez avoir installé le serveur Zabbix et l'interface installée. Vous devez être connecté à Zabbix en tant qu'utilisateur *Admin* par défaut.

Créer un utilisateur MySQL avec des privilèges limités

Il est recommandé de créer un utilisateur MySQL dont les privilèges sont limités à ceux nécessaires à la supervision de MySQL. Remplacez <password> par un mot de passe fort et unique.

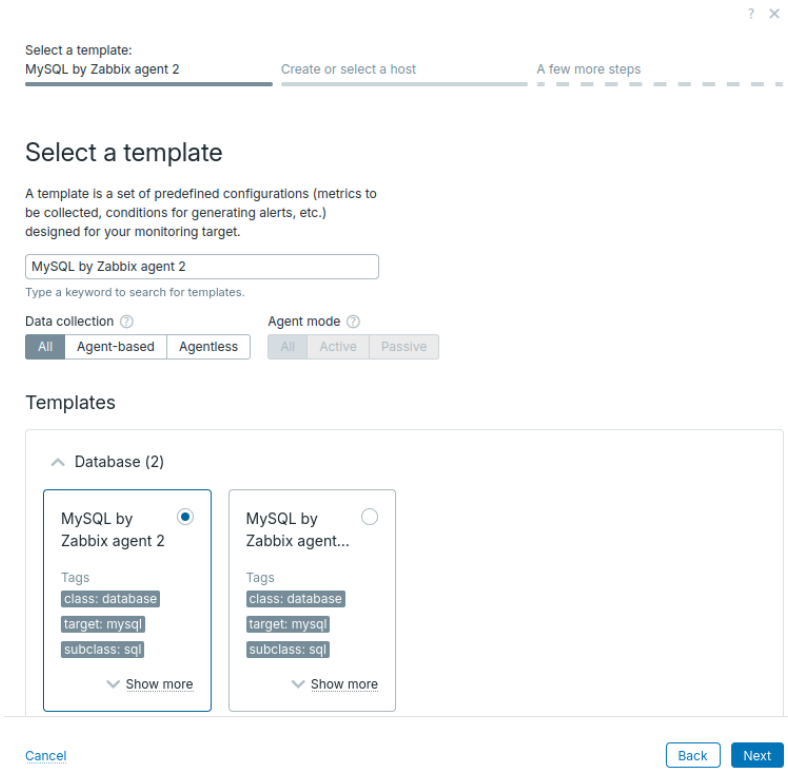
```
mysql> CREATE USER 'zbx_monitor'@'%' IDENTIFIED BY '<password>';
mysql> GRANT REPLICATION CLIENT,PROCESS,SHOW DATABASES,SHOW VIEW ON *.* TO 'zbx_monitor'@'%';
```

Appliquer le modèle de surveillance MySQL

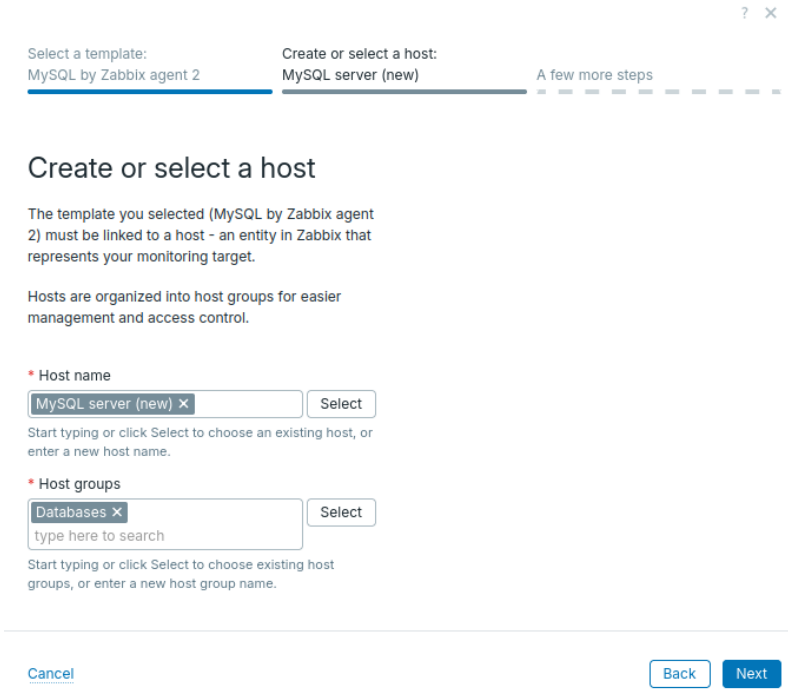
Dans Zabbix, ce modèle est prêt à l'emploi. Il contient des éléments préconfigurés prêts pour la collecte de données, la visualisation et l'analyse.

Pour appliquer le modèle :

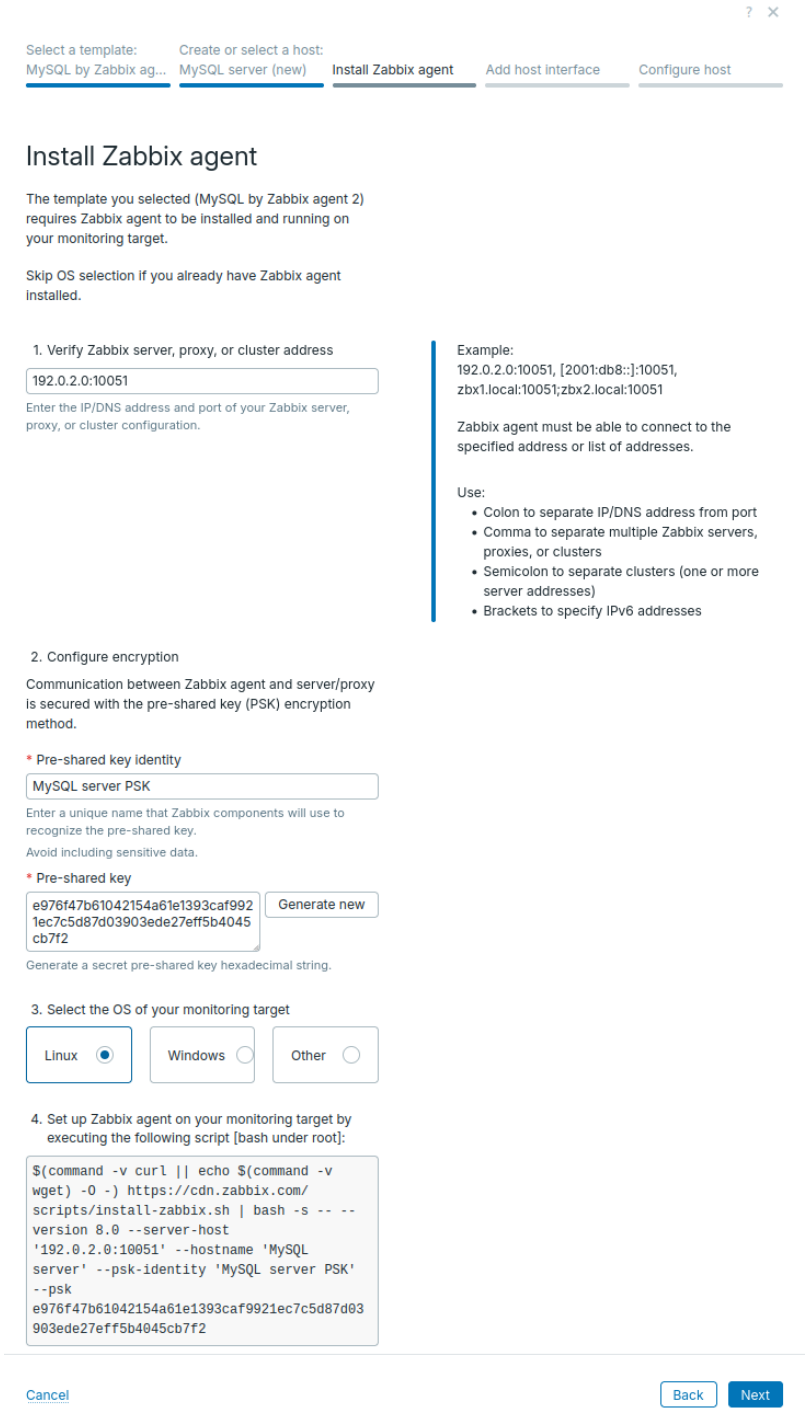
1. Lancez l'Assistant hôte dans *Collecte de données* > *Hôtes*.
2. Sélectionnez le modèle *MySQL by Zabbix agent 2*.



3. Créez un hôte serveur MySQL et ajoutez-lui un groupe. Cet hôte virtuel contiendra toutes les données liées à la surveillance du serveur MySQL.

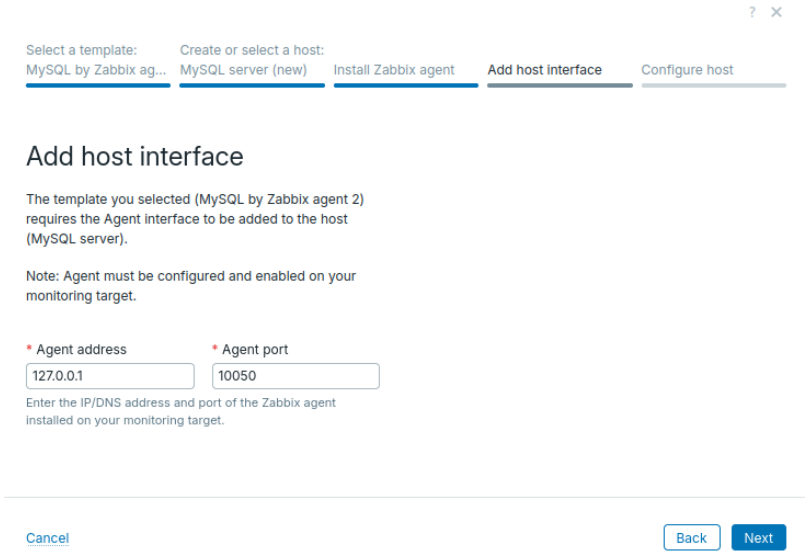


4. Configurez et installez agent 2 (Zabbix agent 2 fournit la surveillance MySQL prête à l’emploi).

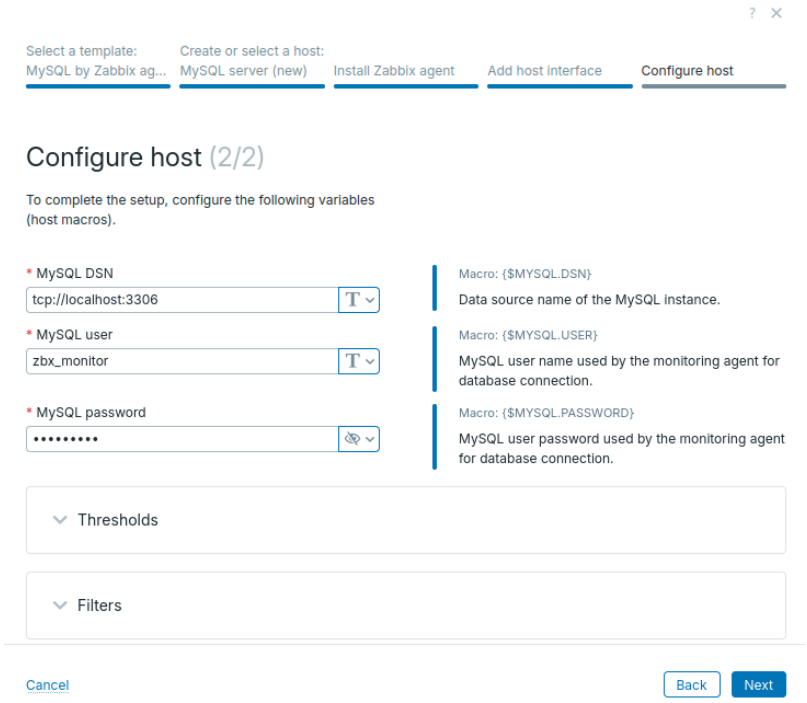


Pour installer l’agent, exécutez la commande fournie sur la machine hébergeant le serveur MySQL. Cela installera l’agent et configurera la connexion de l’agent au serveur Zabbix avec chiffrement.
 Si agent 2 est déjà installé, vous pouvez ignorer cette étape (cliquez sur *Suivant*). Cependant, il vous sera demandé de fournir l’identifiant de la clé prépartagée et la clé prépartagée pour vous connecter à l’agent.

5. Saisissez l’adresse IP et le port de l’agent (serveur Linux).



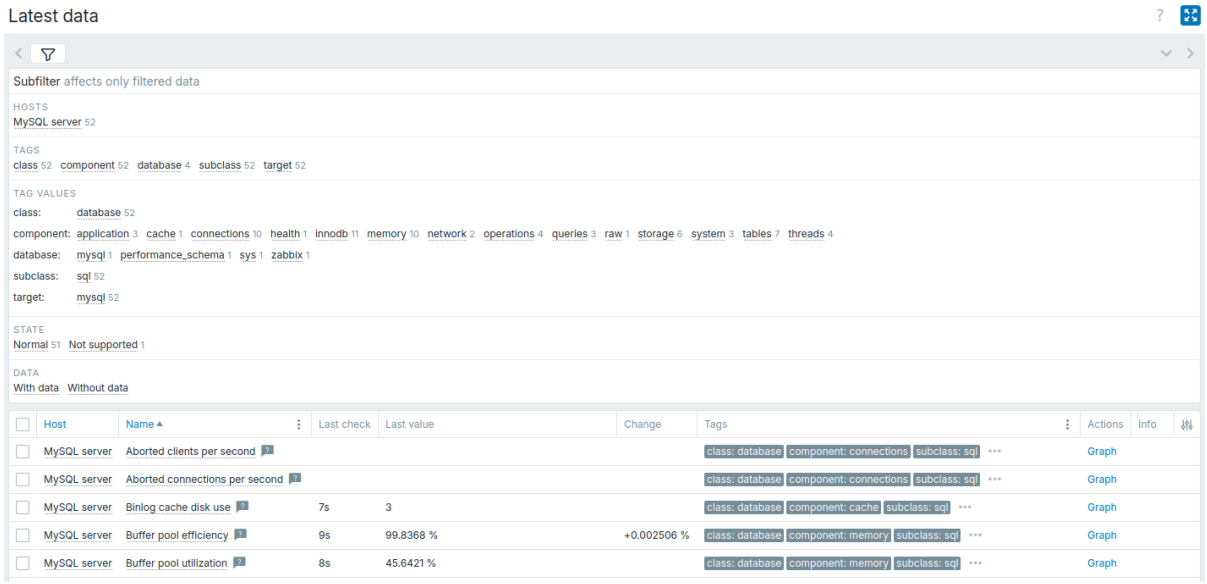
6. Terminez la configuration de l'hôte. Important : il vous sera demandé de saisir le DSN du serveur MySQL, l'utilisateur MySQL et le mot de passe. Fournissez l'utilisateur `zbx_monitor` avec le mot de passe que vous avez créé.



Vous pouvez laisser les autres paramètres à leurs valeurs par défaut.

6. Pour terminer l'Assistant hôte, cliquez sur *Créer* et *Terminer*.

Consultez les dernières données



Félicitations, la supervision du serveur MySQL a été lancée !

Zabbix propose des options de visualisation et envoie des alertes si des problèmes surviennent. Pour configurer les alertes par e-mail, consultez Réception des notifications de problème.

Plus de guides

Cette section de la documentation contient des recettes rapides pour configurer Zabbix afin d'atteindre certains objectifs de supervision couramment requis.

Elle est conçue pour les nouveaux utilisateurs de Zabbix et peut être utilisée comme un guide de navigation à travers d'autres sections de la documentation qui contiennent les informations nécessaires pour résoudre la tâche.

Les guides de référence rapide suivants sont disponibles :

- Surveiller Linux avec Zabbix agent
- Surveiller Windows avec Zabbix agent
- Surveiller Apache via HTTP
- Surveiller MySQL avec Zabbix agent 2
- Surveiller VMware avec Zabbix
- Surveiller le trafic réseau avec Zabbix
- Surveiller le trafic réseau avec Zabbix à l'aide de vérifications actives
- Surveiller des sites web avec des éléments Browser
- Surveiller les certificats de sites web avec Zabbix agent 2 (passif)
- Surveiller un commutateur réseau ou un routeur avec Zabbix
- Surveiller le journal des événements Windows à l'aide de vérifications actives

Surveiller Linux avec agent Zabbix

Introduction

Cette page vous guide à travers les étapes nécessaires pour commencer la surveillance de base des machines Linux avec Zabbix. Les étapes décrites dans ce tutoriel peuvent être appliquées à tout système d'exploitation basé sur Linux.

À qui ce guide s'adresse

Ce guide est conçu pour les nouveaux utilisateurs de Zabbix et contient l'ensemble minimal d'étapes nécessaires pour activer la supervision de base de votre machine Linux. Si vous recherchez des options de personnalisation avancées ou si vous avez besoin d'une configuration plus poussée, consultez la section Configuration du manuel Zabbix.

Prérequis

Avant de poursuivre ce guide, vous devez télécharger et installer le serveur Zabbix et l'interface Zabbix conformément aux instructions correspondant à votre système d'exploitation.

Installer Zabbix agent

Zabbix agent est le processus chargé de collecter les données.

Vérifiez la version de votre serveur Zabbix :

```
zabbix_server -V
```

Installez Zabbix agent de la même version (recommandé) sur la machine Linux que vous souhaitez surveiller. Selon vos besoins de supervision, il peut s'agir de la même machine sur laquelle le serveur Zabbix est installé, ou d'une machine complètement différente.

Choisissez la méthode d'installation la plus adaptée :

- Exécuter dans un conteneur Docker - consultez la liste des images disponibles dans le dépôt Docker Zabbix.
- Installer à partir des packages Zabbix (disponibles pour Alma Linux, CentOS, Debian, Oracle Linux, Raspberry Pi OS, RHEL, Rocky Linux, SUSE Linux Enterprise Server, Ubuntu).
- Compiler à partir des sources.

Configurer Zabbix pour la supervision

L'agent Zabbix peut collecter des métriques en mode actif ou passif (simultanément).

> **Note:**
> Une vérification passive est une simple requête de données. Le serveur Zabbix ou le proxy demande certaines données (par exemple, la charge CPU) et l'agent Zabbix renvoie le résultat au serveur. Les vérifications actives nécessitent un traitement plus complexe. L'agent doit d'abord récupérer depuis le ou les serveurs une liste d'éléments pour un traitement indépendant, puis envoyer les données en bloc. Consultez Vérifications d'agent passives et actives pour plus d'informations.

Les modèles de supervision fournis par Zabbix proposent généralement deux alternatives - un modèle pour l'agent Zabbix et un modèle pour l'agent Zabbix (actif). Avec la première option, l'agent collectera les métriques en mode passif. De tels modèles fourniront des résultats de supervision identiques, mais en utilisant des protocoles de communication différents.

La configuration supplémentaire de Zabbix dépend du fait que vous sélectionniez un modèle pour des vérifications d'agent Zabbix actives ou passives.

Vérifications passives

Agent Zabbix

1. Ouvrez le fichier de configuration de l'agent sur la machine où l'agent est installé.

```
sudo vi /etc/zabbix/zabbix_agentd.conf
```

2. Ajoutez l'adresse IP ou le nom DNS (et le port facultatif) de votre serveur Zabbix au paramètre du serveur. Par exemple :

```
Server=192.0.2.1:10051
```

L'agent Zabbix utilisera cette adresse pour accepter les connexions entrantes uniquement depuis les serveurs Zabbix ou les proxy spécifiés afin de récupérer les données.

3. Redémarrez l'agent Zabbix.

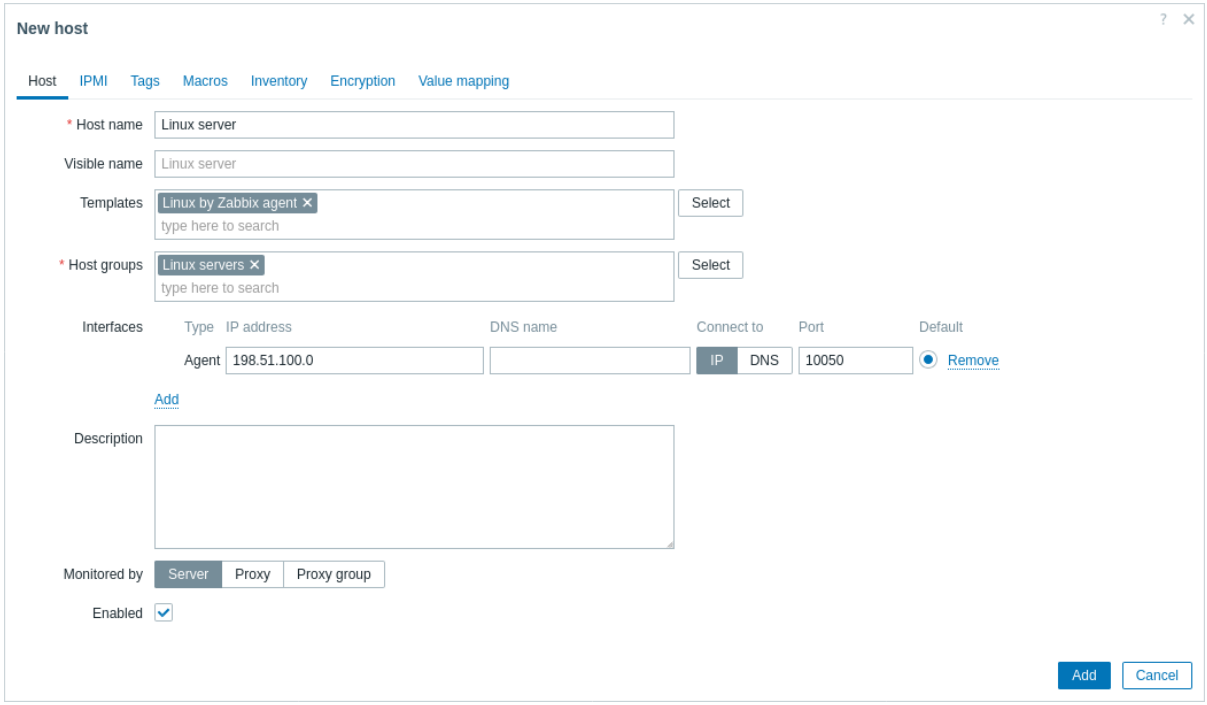
```
systemctl restart zabbix-agent
```

Interface Zabbix

1. Connectez-vous à l'interface Zabbix.

2. Créez un hôte dans l'interface web Zabbix.

- Dans le champ *Host name*, saisissez un nom d'hôte (par exemple, "Linux server").
- Dans le champ *Templates*, saisissez ou sélectionnez le modèle "Linux by Zabbix agent" qui sera lié à l'hôte.
- Dans le champ *Host groups*, saisissez ou sélectionnez un groupe d'hôtes (par exemple, "Linux servers").
- Dans le paramètre *Interfaces*, ajoutez une interface *Agent* et indiquez l'adresse IP ou le nom DNS de la machine Linux sur laquelle l'agent est installé.



3. Cliquez sur *Add* pour ajouter l'hôte. Cet hôte représentera la machine Linux surveillée.

Vérifications actives

Agent Zabbix

1. Ouvrez le fichier de configuration de l'agent sur la machine où l'agent est installé :

```
sudo vi /etc/zabbix/zabbix_agentd.conf
```

2. Ajoutez l'adresse IP ou le nom DNS (et le port facultatif) de votre serveur Zabbix au paramètre ServerActive. Par exemple :

```
ServerActive=192.0.2.1:10051
```

L'agent Zabbix utilisera cette adresse pour se connecter au port trapper du serveur Zabbix (par défaut : 10051) et demander les données de configuration pour les contrôles actifs.

3. Définissez le paramètre Hostname, qui doit correspondre au nom d'hôte qui sera défini dans l'interface Zabbix. Dans notre exemple, il s'agit de :

```
Hostname=Linux server
```

La valeur de Hostname doit correspondre, car pour les contrôles actifs, l'agent Zabbix l'utilise pour récupérer la configuration correcte de l'hôte depuis le serveur. Plus précisément, l'agent établit une connexion avec le serveur et s'identifie à l'aide de la valeur Hostname. Le serveur fournit ensuite la configuration de supervision de cet hôte. Si ces valeurs diffèrent, l'agent ne recevra pas la configuration appropriée, ce qui entraînera l'absence de métriques ou des problèmes de supervision.

4. Redémarrez l'agent Zabbix.

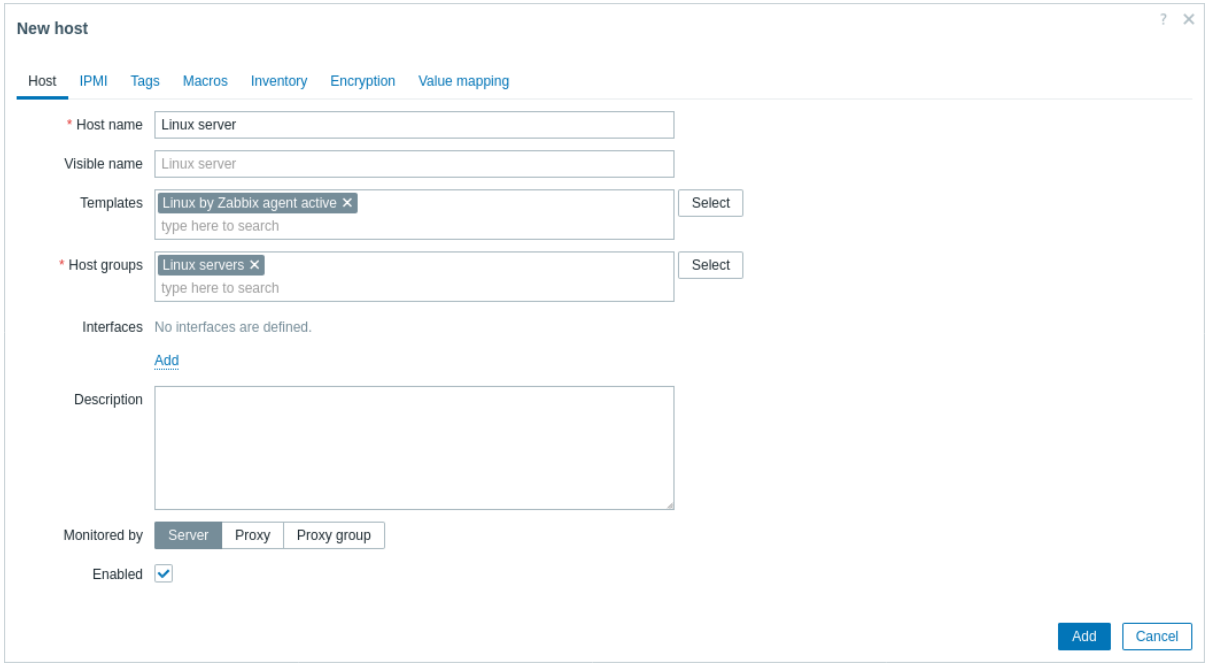
```
systemctl restart zabbix-agent
```

Interface Zabbix

1. Connectez-vous à l'interface Zabbix.

2. Créez un hôte dans l'interface web Zabbix.

- Dans le champ *Host name*, saisissez un nom d'hôte (par exemple, "Linux server") qui correspond à la valeur du paramètre *Hostname* définie précédemment dans le fichier de configuration de l'agent.
- Dans le champ *Templates*, saisissez ou sélectionnez le modèle "Linux by Zabbix agent active" qui sera lié à l'hôte.
- Dans le champ *Host groups*, saisissez ou sélectionnez un groupe d'hôtes (par exemple, "Linux servers").
- Laissez *Interfaces* non défini ; une interface n'est pas nécessaire pour les vérifications actives, car l'agent initie la connexion au serveur au lieu d'attendre une connexion du serveur.



3. Cliquez sur *Add* pour ajouter l'hôte. Cet hôte représentera la machine Linux surveillée.

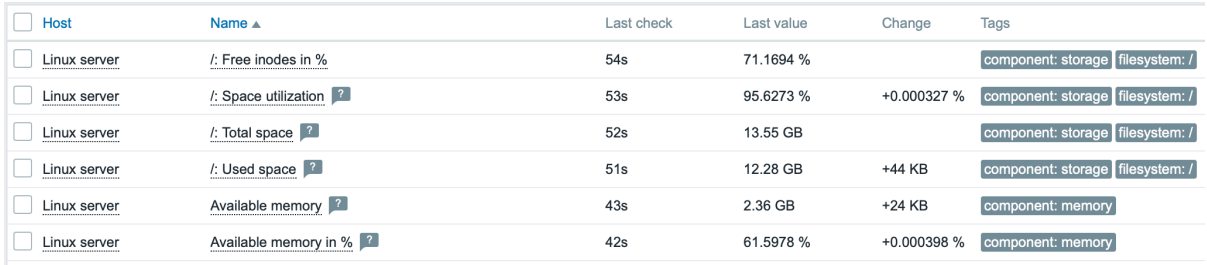
Afficher les métriques collectées

Félicitations ! À ce stade, Zabbix surveille déjà votre machine Linux.

Pour afficher les métriques collectées, ouvrez la section de menu *Monitoring* > *Hosts* menu section et cliquez sur *Latest data* à côté de l'hôte.



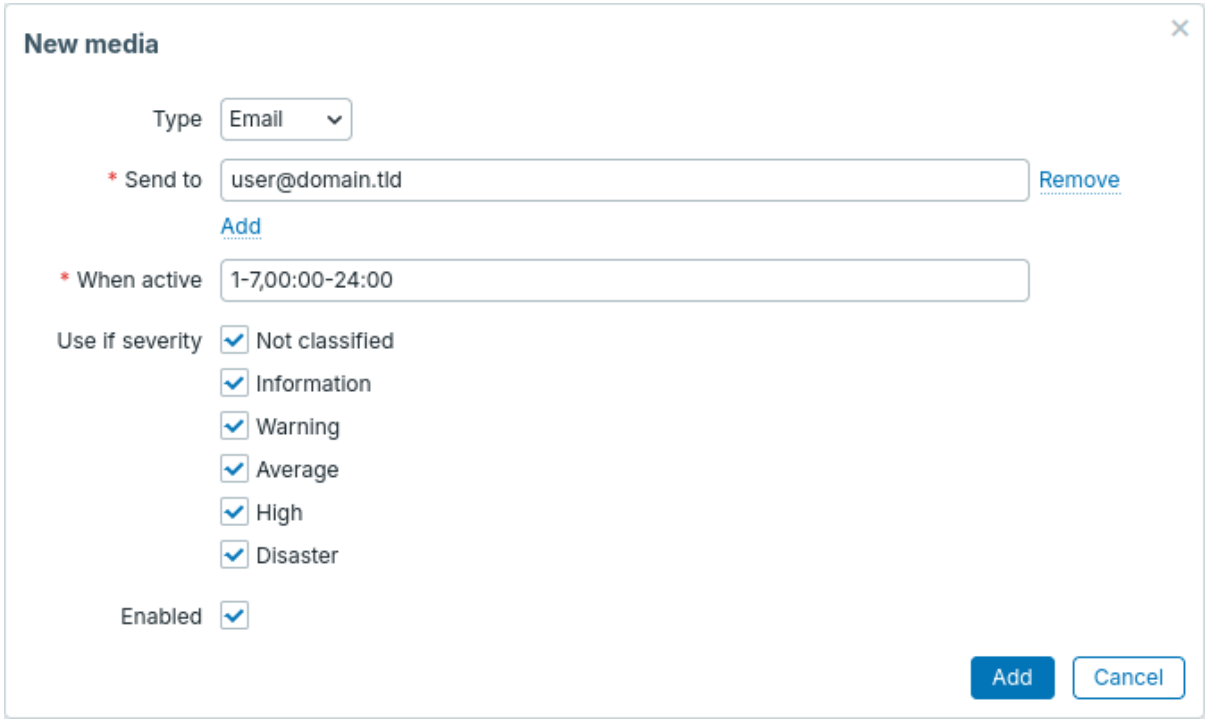
Cette action ouvrira une liste de toutes les dernières métriques collectées depuis l'hôte du serveur Linux.



Configurer les alertes de problème

Zabbix peut vous avertir d'un problème sur votre infrastructure à l'aide de diverses méthodes.
Ce guide fournit les étapes de configuration pour l'envoi d'alertes par e-mail.

1. Accédez à *Paramètres utilisateur* > *Profil*, puis ouvrez l'onglet *Média* et ajoutez votre e-mail.



2. Suivez le guide pour Recevoir une notification de problème.

La prochaine fois que Zabbix détectera un problème, vous devriez recevoir une alerte par e-mail.

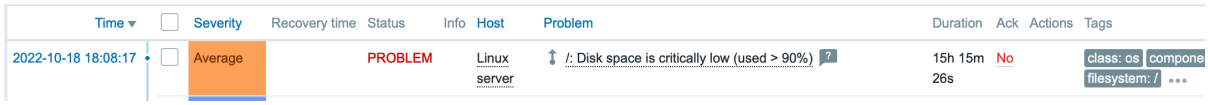
Testez votre configuration

Sous Linux, vous pouvez simuler une charge CPU élevée et, par conséquent, recevoir une alerte de problème en exécutant :

```
cat /dev/urandom | md5sum
```

Vous devrez peut-être lancer plusieurs processus md5sum pour que la charge CPU dépasse le seuil.

Lorsque Zabbix détecte le problème, il apparaîtra dans la section *Monitoring* > *Problems*.



Si les alertes sont configurées, vous recevrez également la notification de problème.

Voir aussi :

- Créer un élément - comment commencer à surveiller des métriques supplémentaires (surveillance personnalisée sans modèles).
- Éléments de l'agent Zabbix, Éléments de l'agent Zabbix pour Windows - liste complète des métriques que vous pouvez surveiller à l'aide de l'agent Zabbix sous Windows.
- Escalades de problème - comment créer des scénarios d'alerte en plusieurs étapes (par exemple, envoyer d'abord un message à l'administrateur système, puis, si un problème n'est pas résolu dans les 45 minutes, envoyer un message au responsable du centre de données).
- Installation à partir de paquets - comment installer les composants Zabbix à l'aide des paquets officiels RPM et DEB pour diverses distributions Linux, en garantissant l'accès aux dernières fonctionnalités et corrections de bogues.

Surveiller Windows avec Zabbix agent

Introduction

Cette page vous guide à travers les étapes nécessaires pour commencer la supervision de base des machines Windows avec Zabbix.

À qui s'adresse ce guide

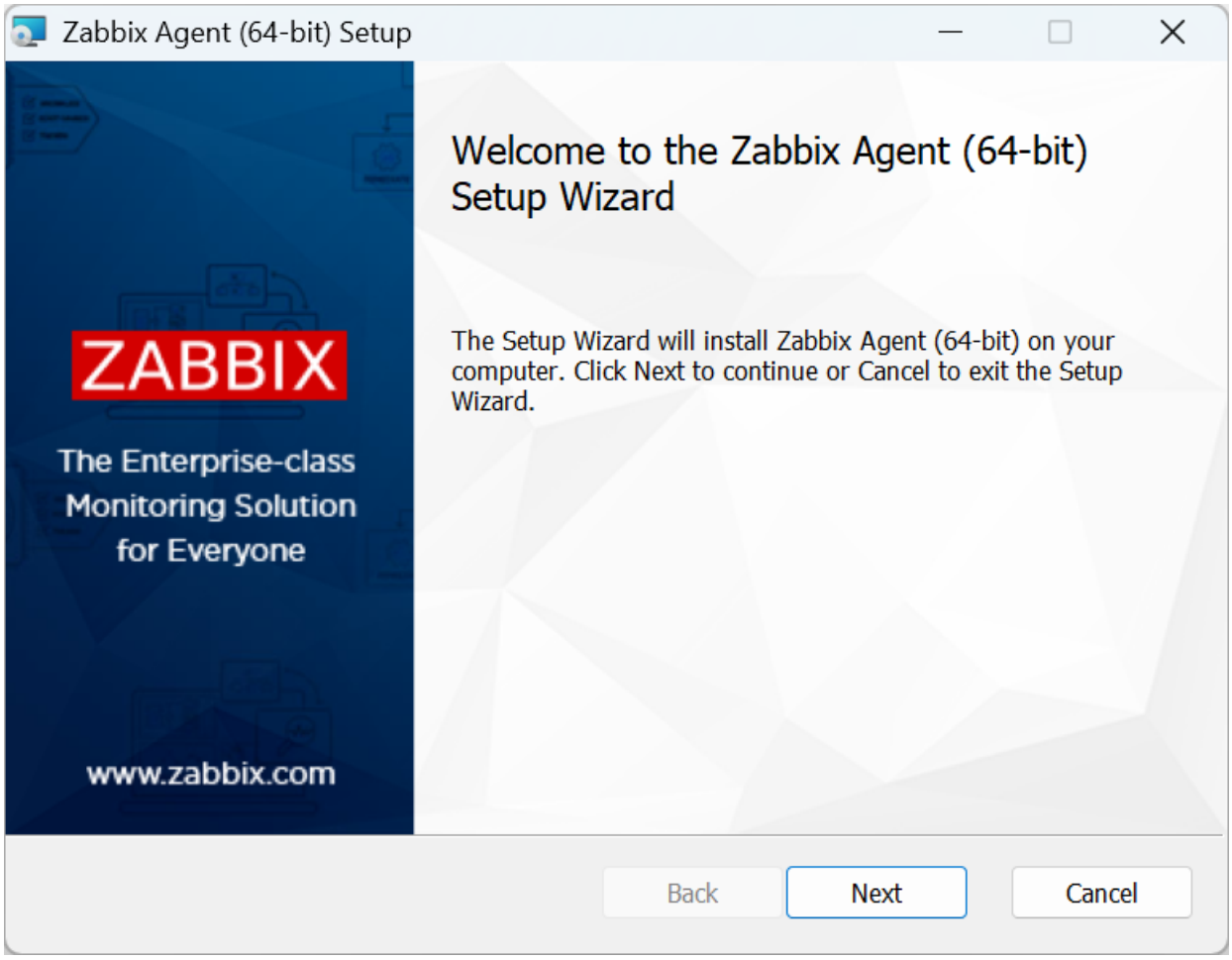
Ce guide est destiné aux nouveaux utilisateurs de Zabbix et contient l'ensemble minimal d'étapes nécessaires pour activer la supervision de base de votre machine Windows. Si vous recherchez des options de personnalisation avancées ou si vous avez besoin d'une configuration plus poussée, consultez la section Configuration du manuel Zabbix.

Prérequis

Avant de suivre ce guide d'installation, vous devez télécharger et installer le serveur Zabbix et l'interface Zabbix conformément aux instructions correspondant à votre système d'exploitation.

Installer Zabbix agent

Zabbix agent est le processus responsable de la collecte des données. Vous devez l'installer sur la machine Windows que vous souhaitez surveiller. Suivez les instructions d'installation de Zabbix agent pour Windows.



Configurer Zabbix pour la supervision

L'agent Zabbix peut collecter des métriques en mode actif ou passif (simultanément).

Note:
Une vérification passive est une simple requête de données. Le serveur Zabbix ou le proxy demande certaines données (par exemple, la charge CPU) et l'agent Zabbix renvoie le résultat au serveur. Les vérifications actives nécessitent un traitement plus complexe. L'agent doit d'abord récupérer depuis le ou les serveurs une liste d'éléments pour un traitement indépendant, puis envoyer les données en bloc. Voir Vérifications passives et actives de l'agent pour plus d'informations.

Les modèles de supervision fournis par Zabbix proposent généralement deux □□□□□□ - un modèle pour l'agent Zabbix et un modèle pour l'agent Zabbix (actif). Avec la première option, l'agent collectera les métriques en mode passif. Ces modèles fourniront des résultats de supervision identiques, mais en utilisant des protocoles de communication différents.

La configuration Zabbix supplémentaire dépend du fait que vous sélectionniez un modèle pour des vérifications de l'agent Zabbix actives ou passives.

Vérifications passives

Interface Zabbix

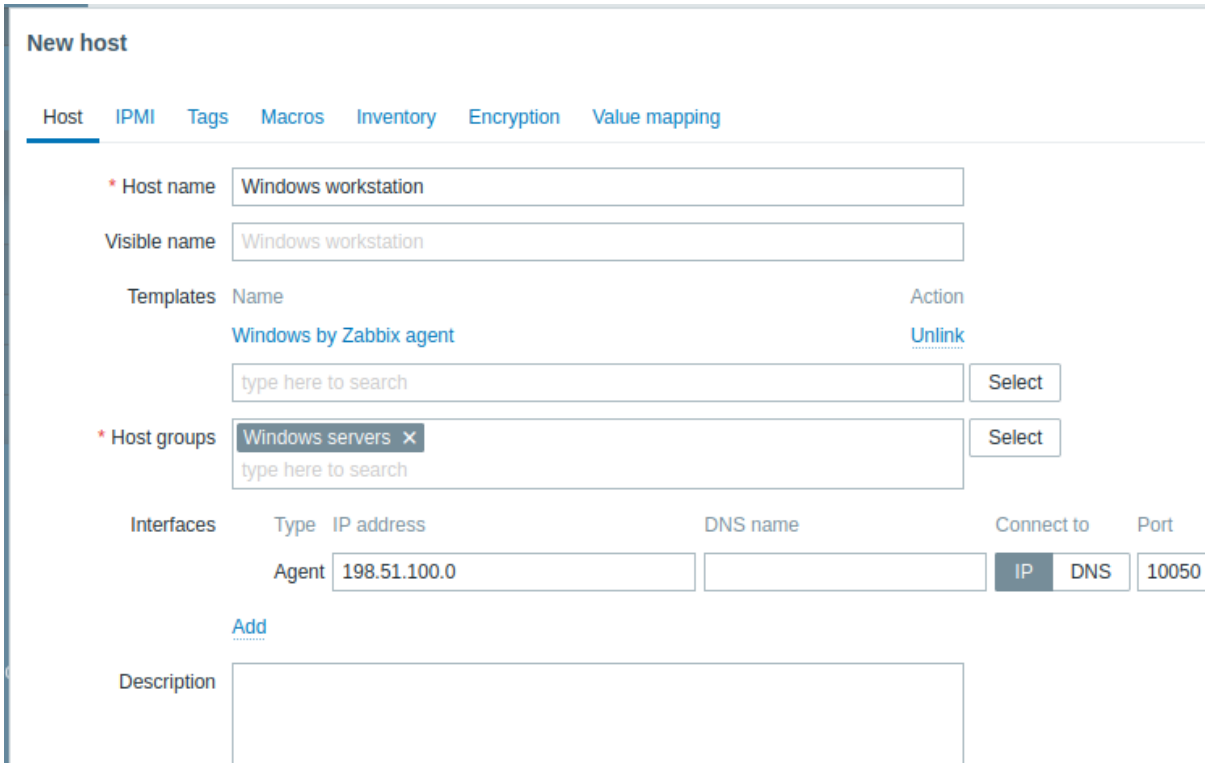
1. Connectez-vous à l'interface Zabbix.

2. Créez un hôte dans l'interface web Zabbix.

Cet hôte représentera votre machine Windows.

3. Dans le paramètre *Interfaces*, ajoutez une interface *agent* et indiquez l'adresse IP ou le nom DNS de la machine Windows sur laquelle l'agent est installé.

4. Dans le paramètre *Modèles*, saisissez ou sélectionnez *Windows by Zabbix agent*.



Agent Zabbix

Pour les vérifications passives, l'agent Zabbix doit connaître l'adresse IP ou le nom DNS du serveur Zabbix. Si vous avez fourni les informations correctes lors de l'installation de l'agent, le fichier de configuration a déjà été mis à jour. Sinon, vous devez les spécifier manuellement. Ouvrez le fichier de configuration de l'agent et ajoutez l'adresse IP ou le nom DNS de votre serveur Zabbix au paramètre `Server`.

Exemple :

```
Server=192.0.2.22
```

Vérifications actives

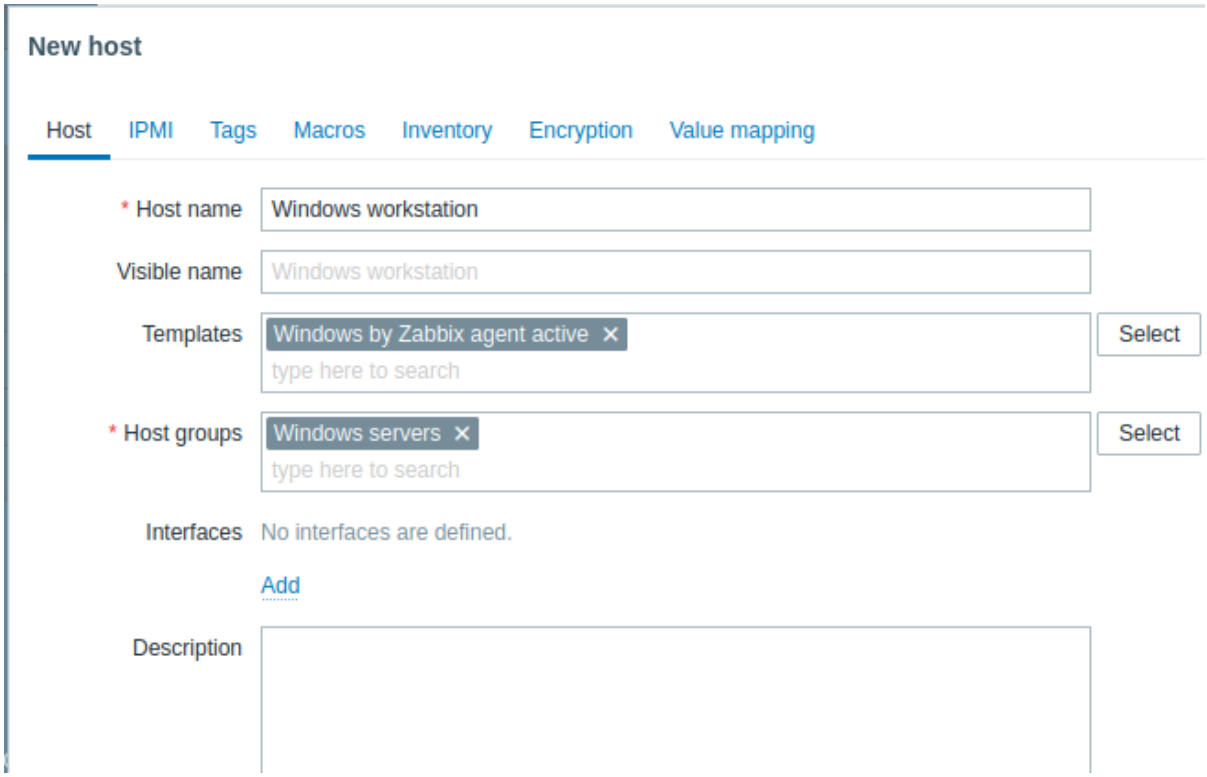
Interface Zabbix

1. Connectez-vous à l'interface Zabbix.

2. Créez un hôte dans l'interface web Zabbix.

Cet hôte représentera votre machine Windows.

3. Dans le paramètre *Templates*, saisissez ou sélectionnez *Windows by Zabbix agent active*.



Agent Zabbix

Dans le dossier `C:\Program files\Zabbix Agent`, ouvrez le fichier `zabbix_agentd.conf` et ajoutez :

- Le nom de l'hôte que vous avez créé dans l'interface web Zabbix au paramètre *Hostname*.
- L'adresse IP ou le nom DNS de votre machine serveur Zabbix au paramètre *ServerActive* (peut être déjà renseigné si vous l'avez fourni lors de la configuration de l'agent Zabbix).

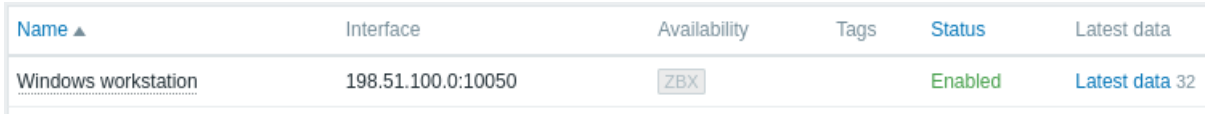
Exemple :

```
ServerActive= 192.0.2.22
Hostname=Windows workstation
```

Afficher les métriques collectées

Félicitations ! À ce stade, Zabbix surveille déjà votre machine Windows.

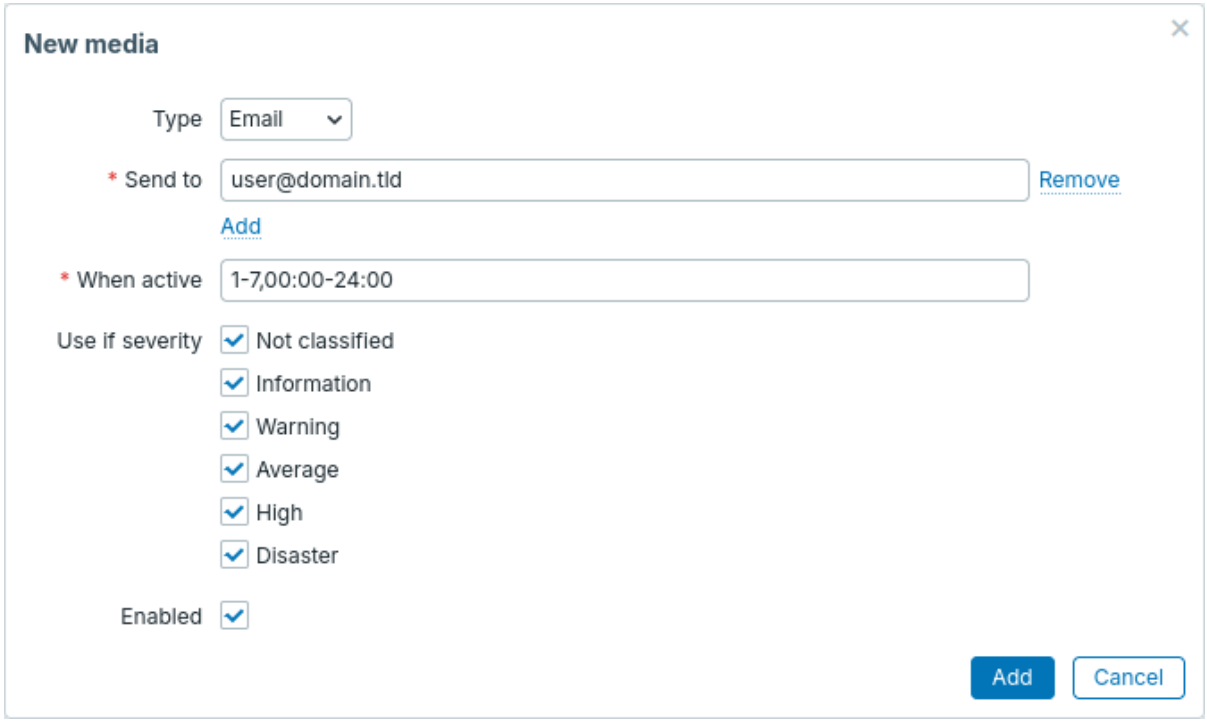
Pour afficher les métriques collectées, ouvrez la section de menu *Monitoring* > *Hosts* menu section et cliquez sur *Latest data* à côté de l'hôte.



Configurer les alertes de problème

Zabbix peut vous avertir d'un problème sur votre infrastructure à l'aide de diverses méthodes. Ce guide fournit les étapes de configuration pour l'envoi d'alertes par e-mail.

1. Allez dans *Paramètres utilisateur* > *Profil*, ouvrez l'onglet *Média* et ajoutez votre e-mail.



2. Suivez le guide pour Recevoir une notification de problème.

La prochaine fois que Zabbix détectera un problème, vous devriez recevoir une alerte par e-mail.

> **Note:**
> Sous Windows, vous pouvez utiliser l'utilitaire CpuStres pour simuler une charge CPU élevée et, par conséquent, recevoir une alerte de problème.

Voir aussi :

- Créer un élément - comment commencer à surveiller des métriques supplémentaires (surveillance personnalisée sans modèles).
- Éléments Zabbix agent, Éléments Zabbix agent pour Windows - liste complète des métriques que vous pouvez surveiller à l'aide de Zabbix agent sous Windows.
- Escalades de problème - comment créer des scénarios d'alerte en plusieurs étapes (par exemple, envoyer d'abord un message à l'administrateur système, puis, si un problème n'est pas résolu en 45 minutes, envoyer un message au responsable du centre de données).

Surveiller Apache via HTTP

Introduction

Cette page présente une méthode rapide et simple pour commencer à surveiller un serveur web Apache sans installer de logiciel supplémentaire.

À qui ce guide est destiné

Ce guide est conçu pour les nouveaux utilisateurs de Zabbix et contient l'ensemble minimal d'étapes nécessaires pour activer la supervision de base de votre installation Apache. Si vous recherchez des options de personnalisation avancées ou avez besoin d'une configuration plus poussée, consultez la section Configuration du manuel Zabbix.

Prérequis

Avant de poursuivre ce guide d'installation, vous devez télécharger et installer le serveur Zabbix et l'interface Zabbix conformément aux instructions correspondant à votre système d'exploitation.

Préparer Apache

1. Vérifiez quelle version d'Apache vous utilisez :

Sur un système basé sur RHEL, exécutez :

```
httpd -v
```

Sur Debian/Ubuntu, exécutez :

```
apache2 -v
```

2. Assurez-vous que le module Status est activé dans votre instance Apache.

Sur un système basé sur RHEL, exécutez :

```
httpd -M | grep status
status_module (shared)
```

Sur Debian/Ubuntu, exécutez :

```
apache2ctl -M | grep status
status_module (shared)
```

Si vous ne voyez pas status_module dans la liste, activez le module en exécutant :

Sur un système basé sur RHEL, exécutez :

```
LoadModule status_module /usr/lib/apache2/modules/mod_status.so
```

Sur Debian/Ubuntu, exécutez :

```
sudo /usr/sbin/a2enmod status
```

3. Modifiez le fichier de configuration Apache pour autoriser l'accès aux rapports d'état depuis l'adresse IP du serveur Zabbix.

Sur un système basé sur RHEL : `/etc/httpd/conf.modules.d/status.conf` :

```
sudo vi /etc/httpd/conf.modules.d/status.conf
```

Sur Debian/Ubuntu : `/etc/apache2/mods-enabled/status.conf` :

```
sudo vi /etc/apache2/mods-enabled/status.conf
```

Ajoutez les lignes suivantes au fichier (**remplacez 198.51.100.255** par l'adresse IP de votre serveur Zabbix) :

- Pour Apache 2.2 :
  <Location /server-status> SetHandler server-status

  Order Deny,Allow Deny from all Allow from 198.51.100.255 </Location>
- Pour Apache 2.4 :
  <Location "/server-status"> SetHandler server-status Require ip 198.51.100.255 </Location>

4. Redémarrez Apache

Sur un système basé sur RHEL, exécutez :

```
sudo systemctl restart httpd
```

Sur Debian/Ubuntu, exécutez :

```
sudo systemctl restart apache2
```

5. Pour vérifier que tout est correctement configuré, exécutez (**remplacez 198.51.100.255** par l'adresse IP de votre serveur Zabbix) :

```
curl 198.51.100.255/server-status
```

La réponse doit contenir les statistiques du serveur web Apache.

Configurer Zabbix pour la surveillance

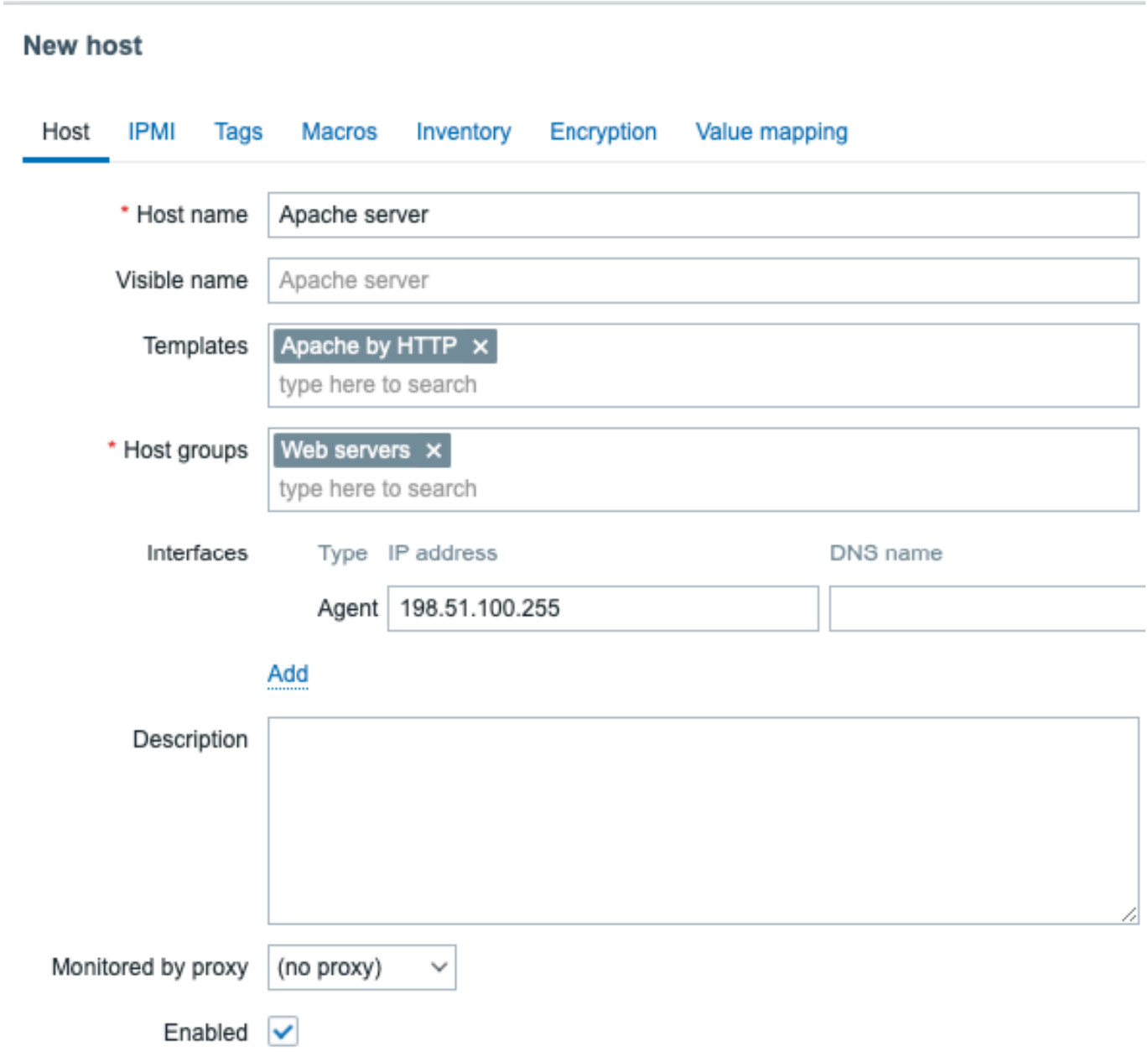
1. Connectez-vous à l'interface Zabbix.

2. Créez un hôte dans l'interface web Zabbix.

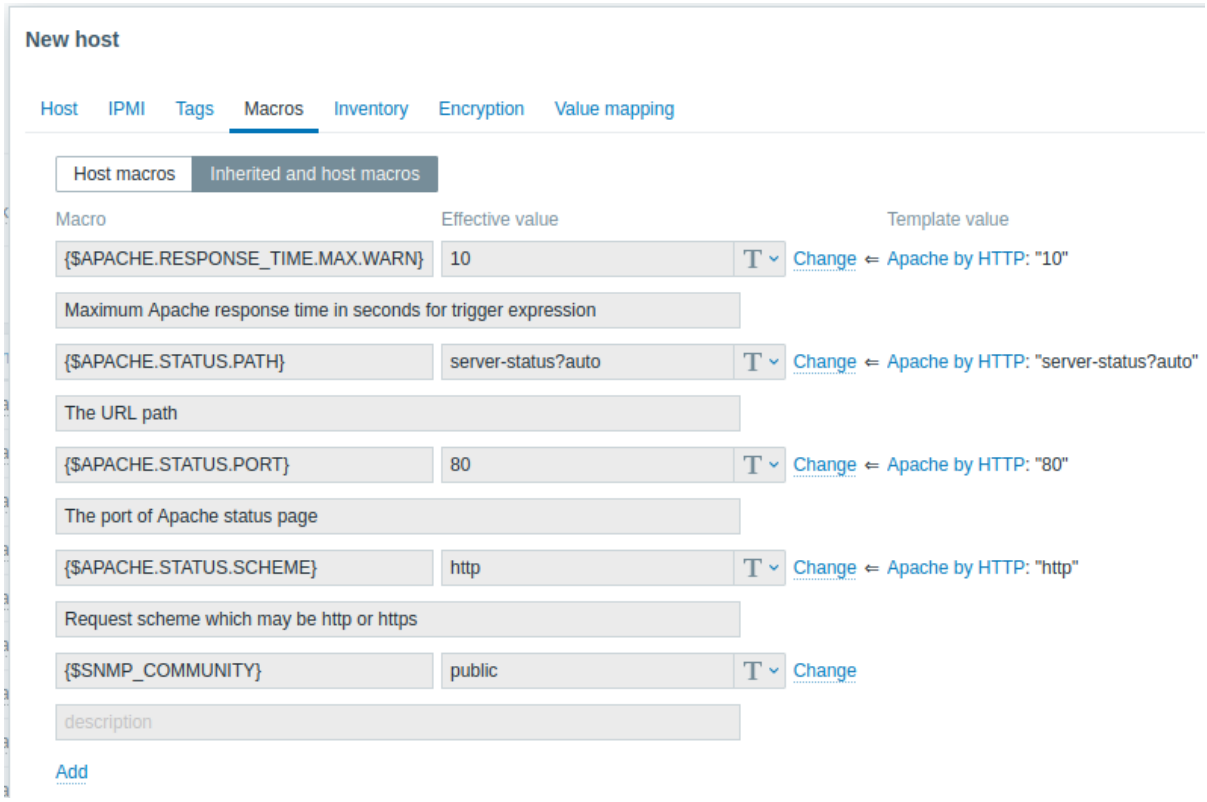
Cet hôte représentera votre serveur Apache.

3. Dans le paramètre *Interfaces*, ajoutez une interface *Agent* et spécifiez l'adresse IP de votre instance Apache. **Vous n'avez pas besoin d'installer l'agent Zabbix sur la machine**, l'interface ne sera utilisée que pour résoudre la macro {HOST.CONN}. Cette macro est utilisée dans les éléments du modèle pour localiser l'instance Apache.

4. Dans le paramètre *Modèles*, saisissez ou sélectionnez *Apache by HTTP*.



5. Passez à l'onglet **Macros** et sélectionnez le mode *Inherited and host macros*. Vérifiez que les valeurs des macros {$APACHE.STATUS.PORT} et {$APACHE.STATUS.SCHEME} conviennent à la configuration de votre installation. Par défaut, le port est 80 et le schéma est http. Modifiez les valeurs des macros si vous utilisez un port et/ou un schéma différents.



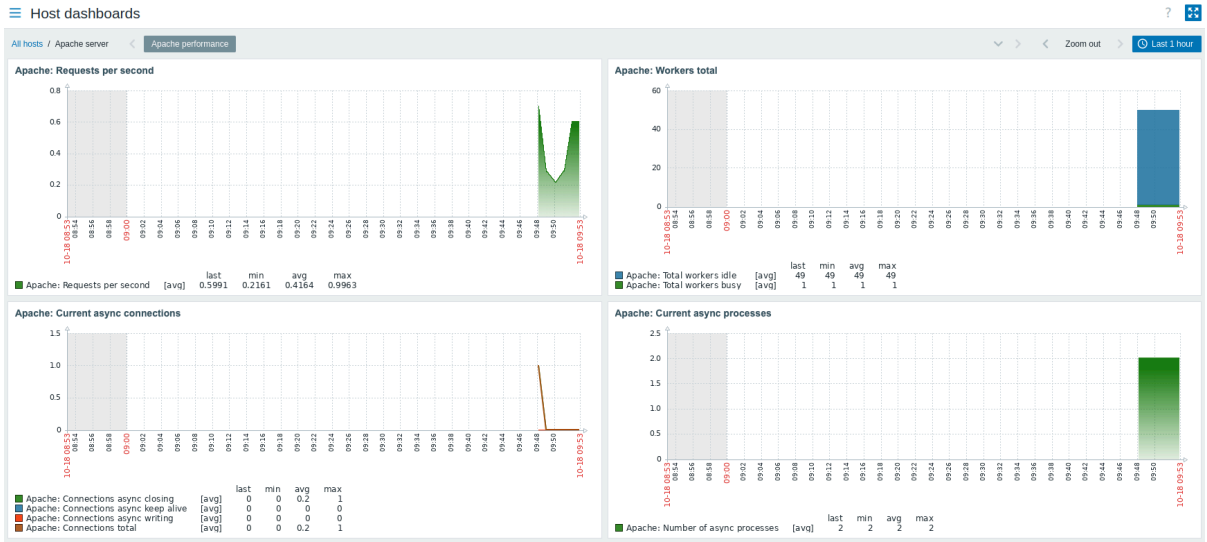
Afficher les métriques collectées

Félicitations ! À ce stade, Zabbix surveille déjà votre serveur web Apache.

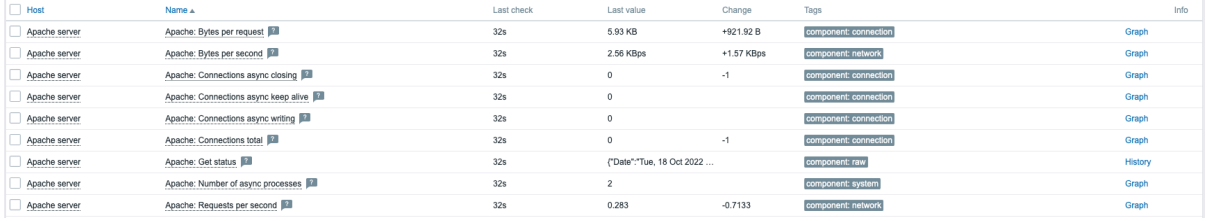
Pour afficher les métriques collectées, ouvrez la section de menu *Monitoring* > *Hosts* menu section et cliquez sur *Dashboards* à côté de l'hôte.



Cette action vous amènera au tableau de bord de l'hôte, avec les métriques les plus importantes collectées depuis la page Apache /server-status.



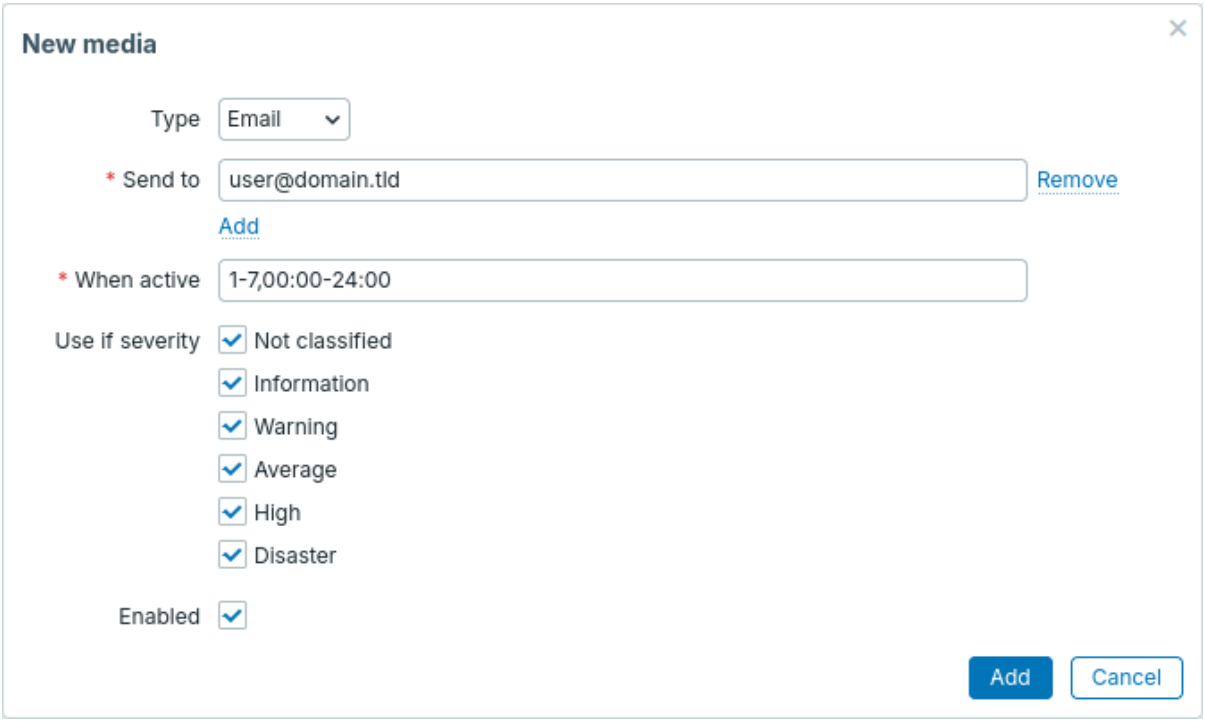
Vous pouvez également, depuis *Monitoring* > *Hosts*, cliquer sur *Latest data* pour afficher toutes les dernières métriques collectées dans une liste.



Configurer les alertes de problème

Zabbix peut vous avertir d'un problème affectant votre infrastructure à l'aide de diverses méthodes.
Ce guide fournit les étapes de configuration pour l'envoi d'alertes par e-mail.

1. Allez dans *Paramètres utilisateur* > *Profil*, basculez vers l'onglet *Media* et ajoutez votre e-mail.



2. Suivez le guide pour Recevoir une notification de problème.

La prochaine fois que Zabbix détectera un problème, vous devriez recevoir une alerte par e-mail.

Testez votre configuration

Pour simuler un vrai problème et recevoir une alerte de test sur un problème :

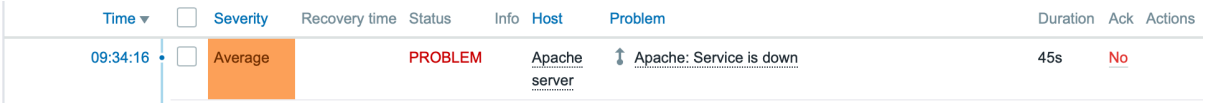
1. Ouvrez la configuration de l'hôte *Apache server* dans Zabbix.

2. Accédez à l'onglet Macros et sélectionnez *Inherited and host macros*.

3. Cliquez sur *Change* à côté de la macro {$APACHE.STATUS.PORT} et définissez un port différent.

4. Cliquez sur *Update* pour enregistrer la configuration de l'hôte.

5. Dans quelques minutes, Zabbix détectera le problème *Apache service is down*, car il ne pourra plus se connecter à l'instance. Il apparaîtra dans la section *Monitoring* > *Problems*.



Si les alertes sont configurées, vous recevrez également la notification du problème.

6. Remettez la valeur de la macro à son état initial pour résoudre le problème et continuer la surveillance d'Apache.

Voir aussi :

- Renforcement de la sécurité du serveur web - paramètres recommandés pour renforcer la sécurité du serveur web.
- Création d'un élément - comment commencer à surveiller des métriques supplémentaires.
- Éléments HTTP - comment surveiller des métriques personnalisées à l'aide de l'agent HTTP.
- Escalades de problèmes - comment créer des scénarios d'alerte en plusieurs étapes (par exemple, envoyer d'abord un message à l'administrateur système, puis, si un problème n'est pas résolu dans les 45 minutes, envoyer un message au responsable du centre de données).

Surveiller MySQL avec Zabbix agent 2

Introduction

Cette page vous guide à travers les étapes nécessaires pour commencer la supervision de base d'un serveur MySQL.

Pour superviser un serveur MySQL, plusieurs approches sont possibles : Zabbix agent, Zabbix agent 2 ou la norme Open Database Connectivity (ODBC). L'objectif principal de ce guide est la supervision d'un serveur MySQL avec Zabbix agent 2, qui est l'approche **recommandée** en raison de sa configuration transparente dans diverses installations. Cependant, cette page fournit également des instructions pour les autres approches, alors n'hésitez pas à choisir celle qui correspond le mieux à vos besoins.

À qui ce guide est destiné

Ce guide est conçu pour les nouveaux utilisateurs de Zabbix et contient l'ensemble minimal d'étapes nécessaires pour activer la supervision de base d'un serveur MySQL. Si vous recherchez des options de personnalisation avancées ou avez besoin d'une configuration plus poussée, consultez la section Configuration du manuel de Zabbix.

Prérequis

Avant de poursuivre ce guide, vous devez télécharger et installer le serveur Zabbix, l'interface Zabbix et l'agent Zabbix 2 conformément aux instructions correspondant à votre système d'exploitation.

Selon votre environnement, certaines étapes de ce guide peuvent légèrement différer. Ce guide est basé sur un environnement exécutant Ubuntu.

Créer un utilisateur MySQL

Pour surveiller un serveur MySQL, Zabbix a besoin d'y accéder ainsi qu'à ses processus. Votre installation MySQL dispose déjà d'un utilisateur avec le niveau d'accès requis (l'utilisateur "zabbix" créé lors de l'installation de Zabbix), cependant, cet utilisateur dispose de plus de privilèges que nécessaire pour une simple surveillance (privilèges permettant de DROP des bases de données, DELETE des entrées de tables, etc.). Par conséquent, il faut créer un utilisateur MySQL destiné *uniquement* à la surveillance du serveur MySQL.

1. Connectez-vous au client MySQL, créez un utilisateur "zbx_monitor" (remplacez *<password>* pour l'utilisateur "zbx_monitor" par un mot de passe de votre choix), et GRANT les privilèges nécessaires à cet utilisateur :

```
mysql -u root -p
# Enter password:

mysql> CREATE USER 'zbx_monitor'@'%' IDENTIFIED BY '<password>';
mysql> GRANT REPLICATION CLIENT,PROCESS,SHOW DATABASES,SHOW VIEW ON *.* TO 'zbx_monitor'@'%';
mysql> quit;
```

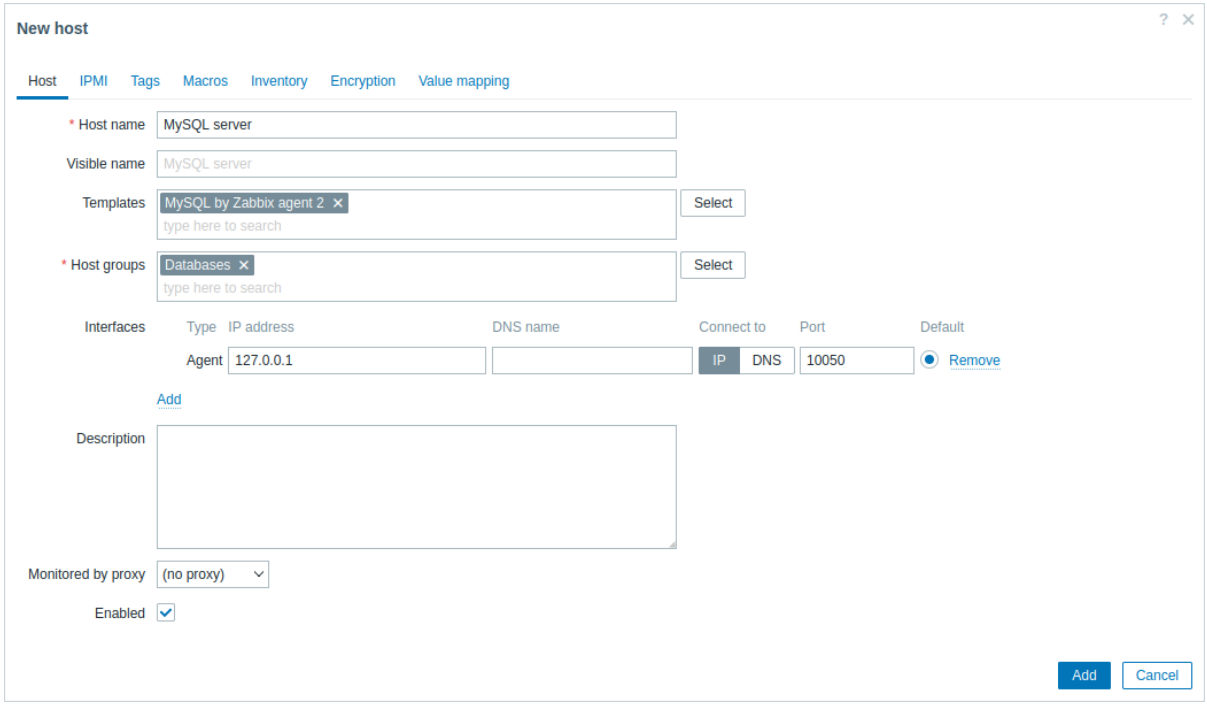
Une fois l'utilisateur créé, vous pouvez passer à l'étape suivante.

Configurer l'interface Zabbix

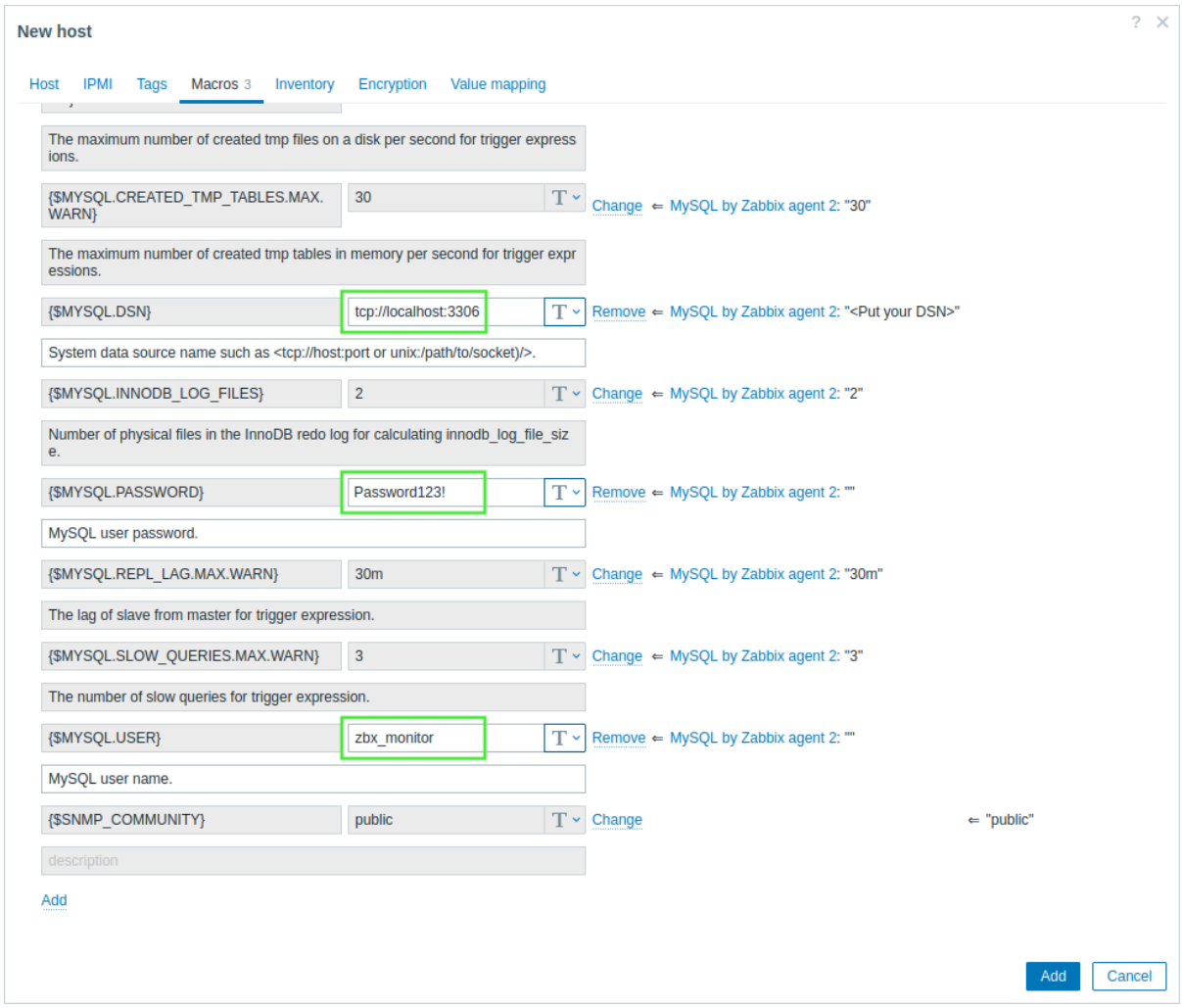
1. Connectez-vous à l'interface Zabbix.

2. Créez un hôte dans l'interface web Zabbix :

- Dans le champ *Host name*, saisissez un nom d'hôte (par exemple, "MySQL server").
- Dans le champ *Templates*, saisissez ou sélectionnez le modèle "MySQL by Zabbix agent 2" qui sera lié à l'hôte.
- Dans le champ *Host groups*, saisissez ou sélectionnez un groupe d'hôtes (par exemple, "Databases").
- Dans le champ *Interfaces*, ajoutez une interface de type "Agent" et indiquez l'adresse IP de votre serveur MySQL. Ce guide utilise "127.0.0.1" (localhost) pour surveiller un serveur MySQL installé sur la même machine que le serveur Zabbix et agent 2 Zabbix.



- Dans l'onglet *Macros*, basculez vers *Inherited and host macros*, recherchez les macros suivantes et cliquez sur *Change* à côté de la valeur de la macro pour la mettre à jour :
  - {$MYSQL.DSN} - définissez la source de données du serveur MySQL (la chaîne de connexion d'une session nommée du fichier de configuration du plugin MySQL de agent 2 Zabbix). Ce guide utilise la source de données par défaut "tcp://localhost:3306" pour surveiller un serveur MySQL installé sur la même machine que le serveur Zabbix et agent 2 Zabbix.
  - {$MYSQL.PASSWORD} - définissez le mot de passe de l'utilisateur MySQL créé précédemment "zbx_monitor".
  - {$MYSQL.USER} - définissez le nom de l'utilisateur MySQL créé précédemment "zbx_monitor".

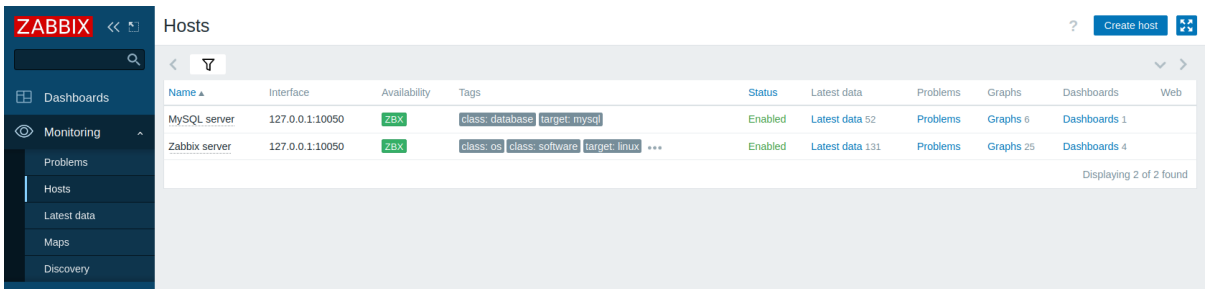


3. Cliquez sur *Add* pour ajouter l'hôte. Cet hôte représentera votre serveur MySQL.

Afficher les métriques collectées

Félicitations ! À ce stade, Zabbix surveille déjà votre serveur MySQL.

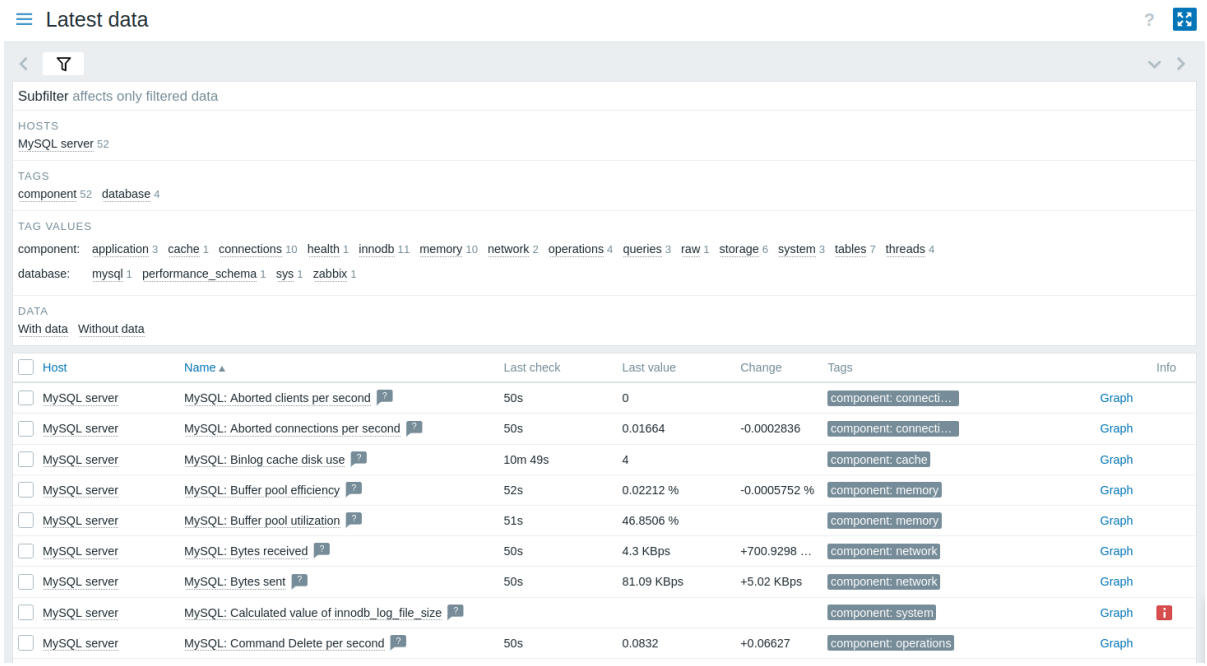
Pour afficher les métriques collectées, accédez à la section de menu *Monitoring > Hosts* et cliquez sur *Dashboards* à côté de l'hôte.



Cette action vous amènera au tableau de bord de l'hôte (configuré au niveau du modèle) avec les métriques les plus importantes collectées depuis le serveur MySQL.



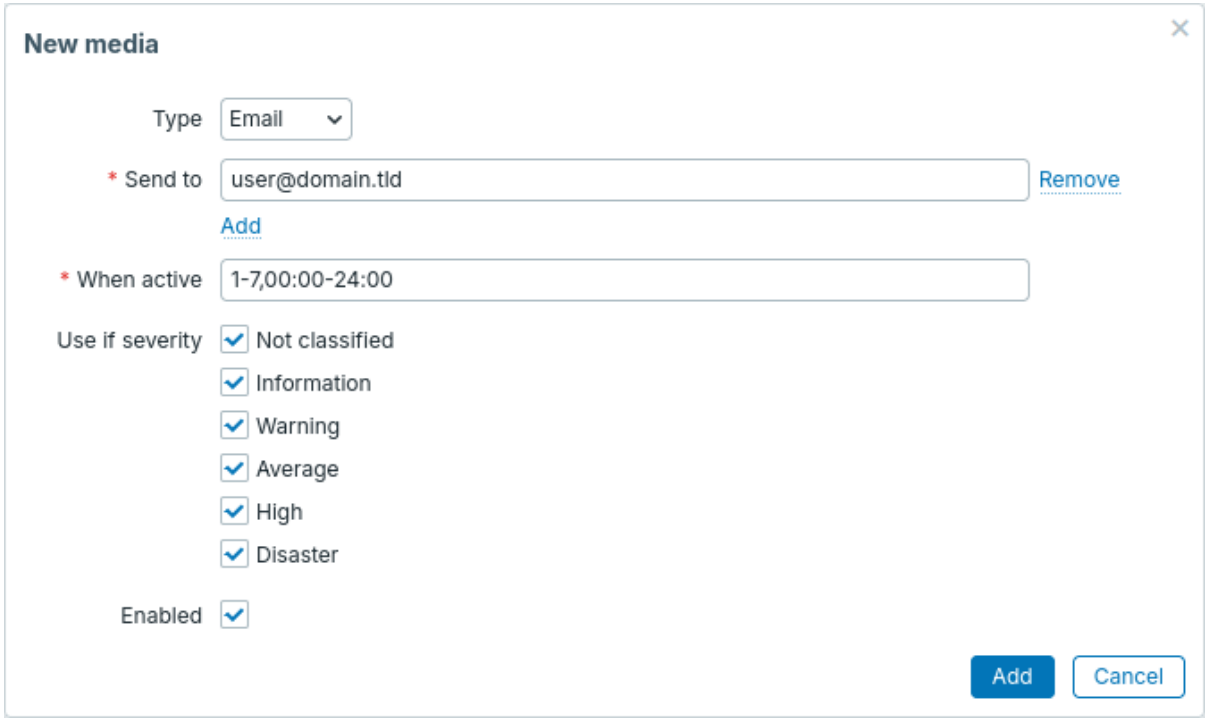
Vous pouvez également, depuis la section de menu *Monitoring > Hosts*, cliquer sur *Latest data* pour afficher toutes les dernières métriques collectées dans une liste. Notez que l'élément *MySQL: Calculated value of innodb_log_file_size* est censé ne contenir aucune donnée, car la valeur sera calculée à partir des données de la dernière heure.



Configurer les alertes de problème

Zabbix peut vous avertir d'un problème avec votre infrastructure en utilisant diverses méthodes. Ce guide fournit les étapes de configuration de base pour l'envoi d'alertes par e-mail.

1. Accédez à *Paramètres utilisateur > Profil*, basculez vers l'onglet *Média* et ajoutez votre e-mail.



2. Suivez le guide pour Recevoir une notification de problème.

La prochaine fois que Zabbix détectera un problème, vous devriez recevoir une alerte par e-mail.

Testez votre configuration

Pour tester votre configuration, nous pouvons simuler un problème réel en mettant à jour la configuration de l'hôte dans l'interface Zabbix.

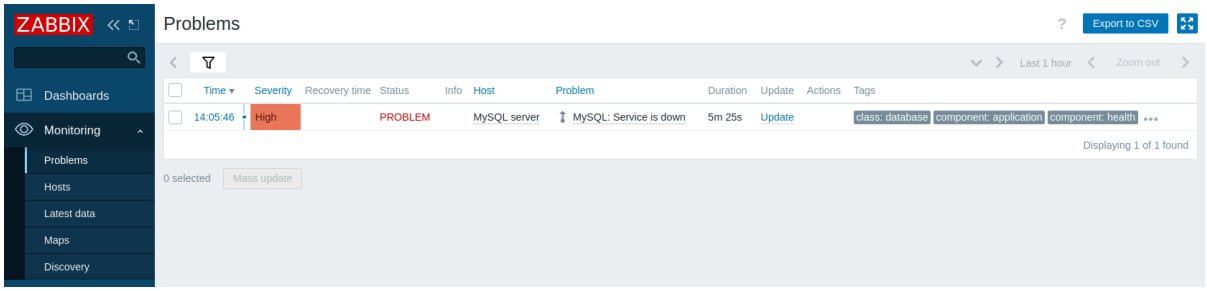
1. Ouvrez la configuration de votre hôte de serveur MySQL dans Zabbix.

2. Accédez à l'onglet *Macros* et sélectionnez *Inherited and host macros*.

3. Cliquez sur *Change* à côté, par exemple, de la valeur de macro {$MYSQL.USER} précédemment configurée, puis définissez un autre nom d'utilisateur MySQL.

4. Cliquez sur *Update* pour mettre à jour la configuration de l'hôte.

5. Dans quelques instants, Zabbix détectera le problème "MySQL: Service is down", car il ne pourra pas se connecter au serveur MySQL. Le problème apparaîtra dans *Monitoring > Problems*.



Si les alertes sont configurées, vous recevrez également la notification du problème.

6. Remettez la valeur de la macro à sa valeur précédente pour résoudre le problème et continuer à surveiller le serveur MySQL.

Autres approches pour surveiller MySQL

Au lieu de surveiller un serveur MySQL avec Zabbix agent 2, vous pouvez également utiliser Zabbix agent ou la norme Open Database Connectivity (ODBC). Bien que l'utilisation de Zabbix agent 2 soit recommandée, certaines configurations peuvent ne pas prendre en charge Zabbix agent 2 ou nécessiter une approche personnalisée.

La principale différence entre Zabbix agent et ODBC réside dans la méthode de collecte des données - Zabbix agent est installé directement sur le serveur MySQL et collecte les données à l'aide de ses fonctionnalités intégrées, tandis qu'ODBC s'appuie sur un pilote ODBC pour établir une connexion au serveur MySQL et récupérer les données à l'aide de requêtes SQL.

Bien que de nombreuses étapes de configuration soient similaires à celles de la surveillance d'un serveur MySQL avec Zabbix agent 2, il existe quelques différences importantes - vous devez configurer Zabbix agent ou ODBC pour pouvoir surveiller un serveur MySQL. Les instructions suivantes vous guideront à travers ces **différences**.

Surveiller MySQL avec Zabbix agent

Pour surveiller un serveur MySQL avec Zabbix agent, vous devez télécharger et installer Zabbix serveur, l'interface Zabbix et Zabbix agent conformément aux instructions de votre système d'exploitation.

Une fois les composants Zabbix requis correctement installés, vous devez créer un utilisateur MySQL comme décrit dans la section *Créer un utilisateur MySQL*.

Après avoir créé l'utilisateur MySQL, vous devez configurer Zabbix agent afin qu'il puisse établir une connexion avec le serveur MySQL et le surveiller. Cela inclut la configuration de plusieurs paramètres utilisateur pour exécuter des vérifications personnalisées de agent, ainsi que la fourniture à Zabbix agent des identifiants nécessaires pour se connecter au serveur MySQL en tant qu'utilisateur "zbx_monitor" créé précédemment.

Configurer l'agent Zabbix

1. Accédez au répertoire des configurations supplémentaires de l'agent Zabbix.

```
cd /etc/zabbix/zabbix_agentd.d
```

> **Attention:**
> Le répertoire des configurations supplémentaires de l'agent Zabbix doit se trouver dans le même répertoire que votre fichier de configuration de l'agent Zabbix ; le chemin par défaut dépend du système d'exploitation et de la méthode d'installation de Zabbix.

Au lieu de définir tous les paramètres utilisateur nécessaires à la supervision du serveur MySQL dans le fichier de configuration de l'agent Zabbix, ces paramètres seront définis dans un fichier distinct du répertoire des configurations supplémentaires.

2. Créez un fichier `template_db_mysql.conf` dans le répertoire des configurations supplémentaires de l'agent Zabbix.

```
vi template_db_mysql.conf
```

3. Copiez le contenu du fichier `template_db_mysql.conf` (situé dans le dépôt Zabbix) dans le fichier `template_db_mysql.conf` que vous avez créé, puis enregistrez-le.

4. Redémarrez l'agent Zabbix pour mettre à jour sa configuration.

```
systemctl restart zabbix-agent
```

Une fois les paramètres utilisateur de l'agent Zabbix configurés, vous pouvez passer à la configuration des identifiants qui permettront à l'agent Zabbix d'accéder au serveur MySQL.

5. Accédez au répertoire personnel de l'agent Zabbix (s'il n'existe pas sur votre système, vous devez le créer ; par défaut : */var/lib/zabbix*).

```
cd /var/lib/zabbix
```

6. Créez un fichier *.my.cnf* dans le répertoire personnel de l'agent Zabbix.

```
vi .my.cnf
```

7. Copiez le contenu suivant dans le fichier *.my.cnf* (remplacez *<password>* par le mot de passe de l'utilisateur "zbx_monitor").

```
[client]
user='zbx_monitor'
password='<password>'
```

Configurer l'interface Zabbix et tester votre configuration

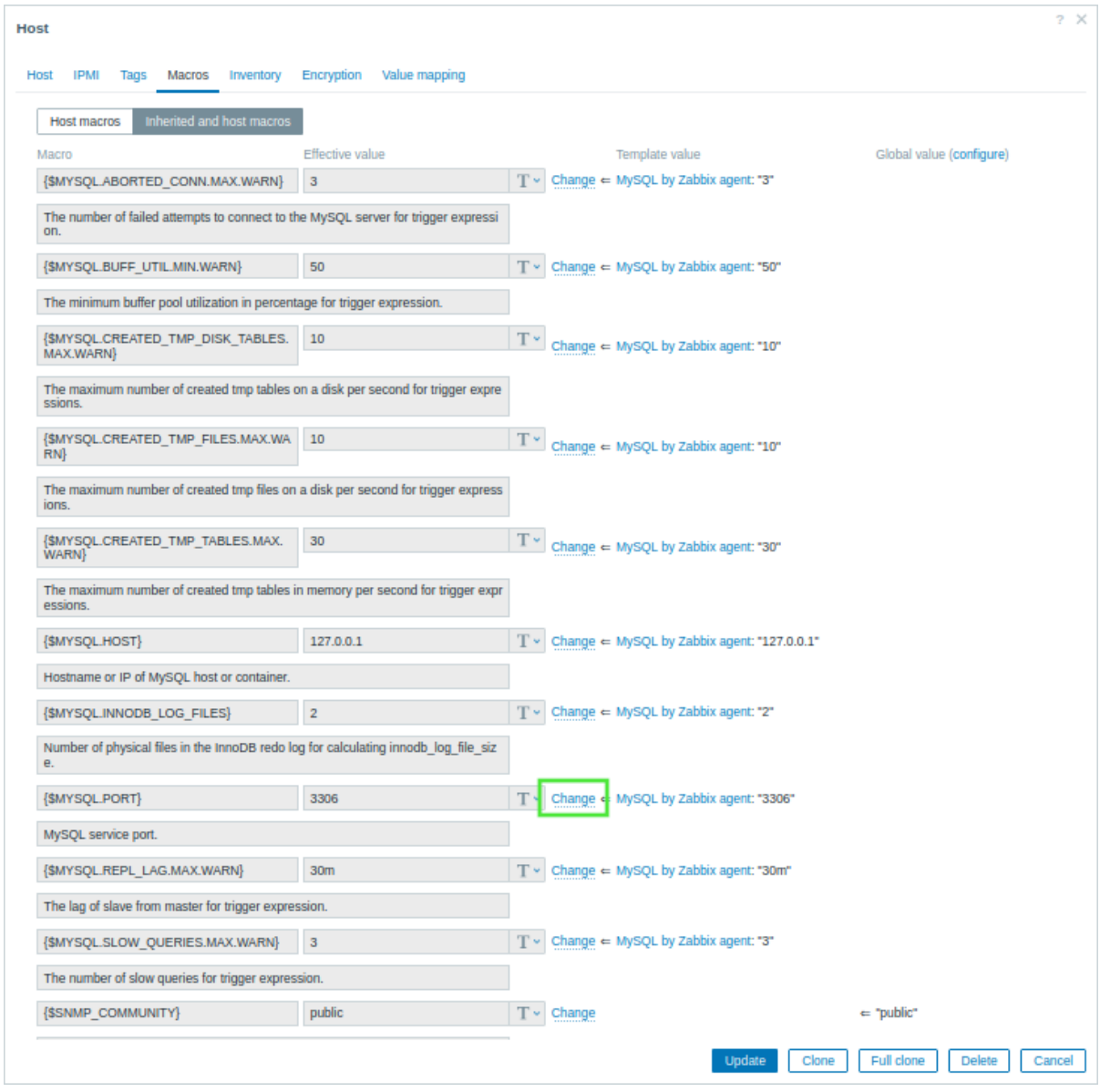
Pour configurer l'interface Zabbix, suivez les instructions de la section *Configurer l'interface Zabbix* avec les ajustements suivants :

- Dans le champ *Modèles*, saisissez ou sélectionnez le modèle "MySQL by Zabbix agent" qui sera lié à l'hôte.
- La configuration des *Macros* n'est pas requise.

Une fois l'interface Zabbix configurée, vous pouvez afficher les métriques collectées et mettre en place des alertes de problème.

Pour tester votre configuration, suivez les instructions de la section *Tester votre configuration* avec les ajustements suivants :

- Dans la section *Macros héritées et macros de l'hôte* de la configuration de l'hôte du serveur MySQL, cliquez sur *Modifier* à côté de la valeur de la macro {$MYSQL.PORT} et définissez un port différent (par exemple, "6033").



Surveiller MySQL avec ODBC

Pour surveiller un serveur MySQL avec ODBC, vous devez télécharger et installer Zabbix serveur et l'interface Zabbix.

Une fois les composants Zabbix requis correctement installés, vous devez créer un utilisateur MySQL comme décrit dans la section *Créer un utilisateur MySQL*.

Après avoir créé l'utilisateur MySQL, vous devez configurer ODBC. Cela inclut l'installation de l'une des implémentations open source les plus couramment utilisées de l'API ODBC - unixODBC - ainsi que d'un pilote unixODBC, et la modification du fichier de configuration du pilote ODBC.

Configurer ODBC

1. Installez unixODBC. La méthode recommandée pour installer unixODBC consiste à utiliser les dépôts de paquets par défaut du système d'exploitation Linux.

```
apt install unixodbc
```

2. Installez le pilote de base de données MariaDB unixODBC. Bien que vous disposiez d'une base de données MySQL, le pilote MariaDB unixODBC est utilisé pour des raisons de compatibilité.

```
apt install odbc-mariadb
```

3. Vérifiez l'emplacement des fichiers de configuration ODBC *odbcinst.ini* et *odbc.ini*.

```
odbcinst -j
```

Le résultat de l'exécution de cette commande devrait être similaire à ce qui suit.

```
unixODBC 2.3.9
DRIVERS............: /etc/odbcinst.ini
SYSTEM DATA SOURCES: /etc/odbc.ini
FILE DATA SOURCES..: /etc/ODBCDataSources
...
```

4. Pour configurer le pilote ODBC pour la surveillance d'une base de données MySQL, vous avez besoin du nom du pilote, qui se trouve dans le fichier *odbcinst.ini*. Dans l'exemple de fichier *odbcinst.ini* suivant, le nom du pilote est "MariaDB Unicode".

```
[MariaDB Unicode]
Driver=libmaodbc.so
Description=MariaDB Connector/ODBC(Unicode)
Threading=0
UsageCount=1
```

5. Copiez le contenu suivant dans le fichier *odbc.ini* (remplacez <*password*> par le mot de passe de l'utilisateur "zbx_monitor"). Ce guide utilise "127.0.0.1" (localhost) comme adresse du serveur MySQL pour surveiller un serveur MySQL installé sur la même machine que le pilote ODBC. Notez le nom de la source de données (DSN) "test", qui sera requis lors de la configuration de l'interface Zabbix.

```
[test]
Driver=MariaDB Unicode
Server=127.0.0.1
User=zbx_monitor
Password=<password>
Port=3306
Database=zabbix
```

Configurer l'interface Zabbix et tester votre configuration

Pour configurer l'interface Zabbix, suivez les instructions de la section *Configurer l'interface Zabbix* avec les ajustements suivants :

- Dans le champ *Modèles*, saisissez ou sélectionnez le modèle "MySQL by ODBC" qui sera lié à l'hôte.
- La configuration des *Interfaces* n'est pas requise.
- La valeur de la macro {$MYSQL.DSN} dans la section *Macros héritées et macros de l'hôte* de la configuration de l'hôte du serveur MySQL doit être définie sur le nom DSN du fichier *odbc.ini*.

Une fois l'interface Zabbix configurée, vous pouvez consulter les métriques collectées, configurer les alertes de problème et tester votre configuration.

Voir aussi

- Création d'un élément - comment commencer à surveiller des métriques supplémentaires.

- Escalades de problème - comment créer des scénarios d'alerte en plusieurs étapes (par exemple, envoyer d'abord un message à l'administrateur système, puis, si un problème n'est pas résolu en 45 minutes, envoyer un message au responsable du centre de données).
- Surveillance ODBC - comment configurer ODBC sur d'autres distributions Linux, et comment commencer à surveiller des métriques supplémentaires liées aux bases de données avec ODBC.
- Modèle *MySQL by Zabbix agent* - informations supplémentaires sur le modèle *MySQL by Zabbix agent*.
- Modèle *MySQL by Zabbix agent 2* - informations supplémentaires sur le modèle *MySQL by Zabbix agent 2*.
- Modèle *MySQL by ODBC* - informations supplémentaires sur le modèle *MySQL by ODBC*.

Surveiller VMware avec Zabbix

Introduction

Cette page vous guide à travers les étapes nécessaires pour commencer la surveillance de base de VMware.

À qui ce guide est destiné

Ce guide est conçu pour les nouveaux utilisateurs de Zabbix et contient l'ensemble minimal d'étapes nécessaires pour activer la supervision de base de VMware. Si vous recherchez des options de personnalisation avancées ou si vous avez besoin d'une configuration plus poussée, consultez la section Supervision des machines virtuelles ou la section Configuration du manuel de Zabbix.

Prérequis

Avant de poursuivre ce guide, vous devez télécharger et installer le serveur Zabbix et l'interface Zabbix conformément aux instructions correspondant à votre système d'exploitation.

Selon votre environnement, certaines étapes de ce guide peuvent légèrement différer. Ce guide est basé sur un environnement exécutant Ubuntu.

Il est supposé que VMware est déjà configuré. Ce guide ne couvre pas la configuration de VMware.

Configurer le serveur Zabbix

Pour surveiller VMware, les processus Zabbix *vmware collector* doivent être activés. Pour plus d'informations sur le fonctionnement de la surveillance VMware, consultez Surveillance des machines virtuelles.

1. Ouvrez le fichier de configuration du serveur Zabbix.

```
vi /etc/zabbix/zabbix_server.conf
```

2. Recherchez le paramètre `StartVMwareCollectors` dans le fichier de configuration du serveur Zabbix et définissez-le sur 2 ou plus (la valeur par défaut est 0).

```
###### Option: StartVMwareCollectors
#####       Number of pre-forked vmware collector instances.
#####
##### Mandatory: no
##### Range: 0-250
##### Default:
##### StartVMwareCollectors=0

StartVMwareCollectors=2
```

3. Redémarrez le serveur Zabbix.

```
systemctl restart zabbix-server
```

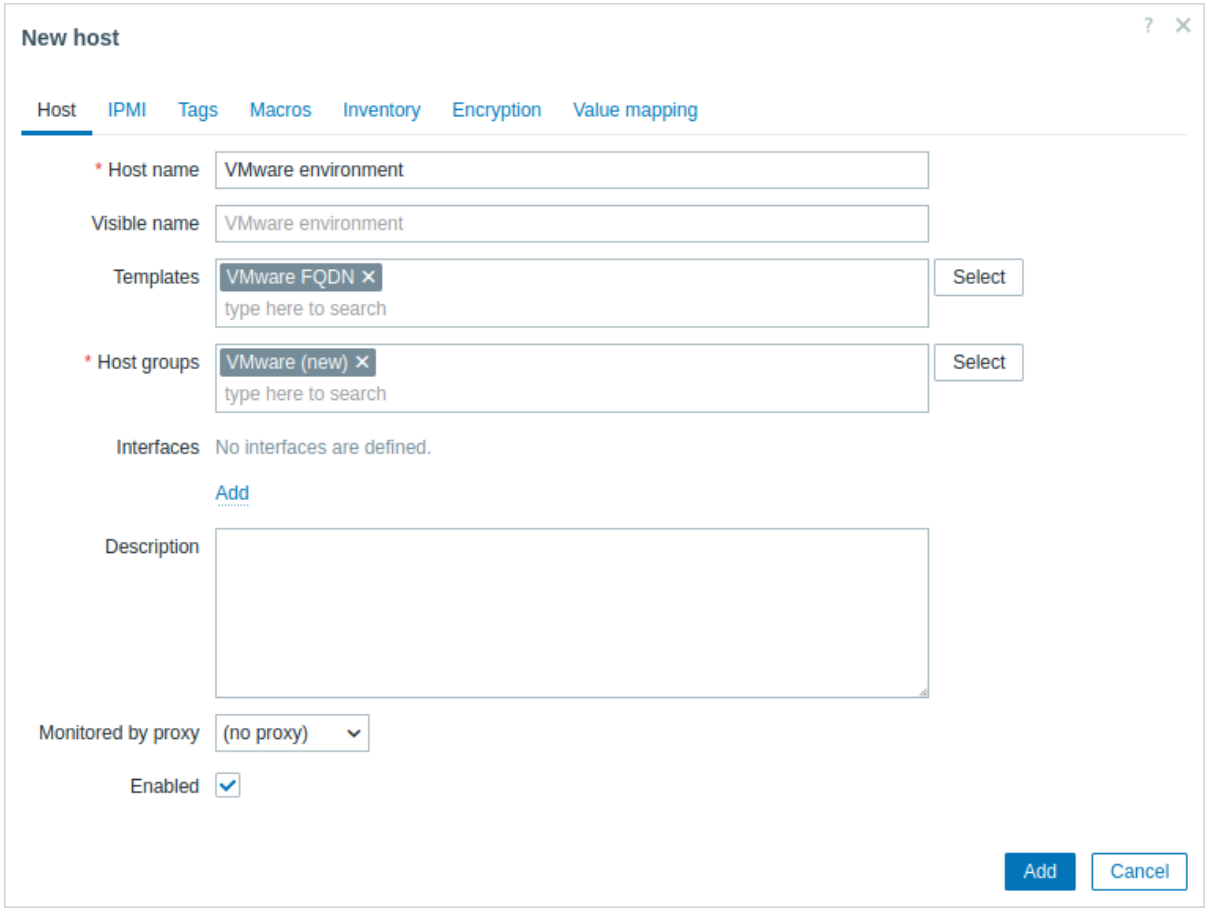
Une fois les processus *vmware collector* démarrés, passez à l'étape suivante.

Configurer l'interface Zabbix

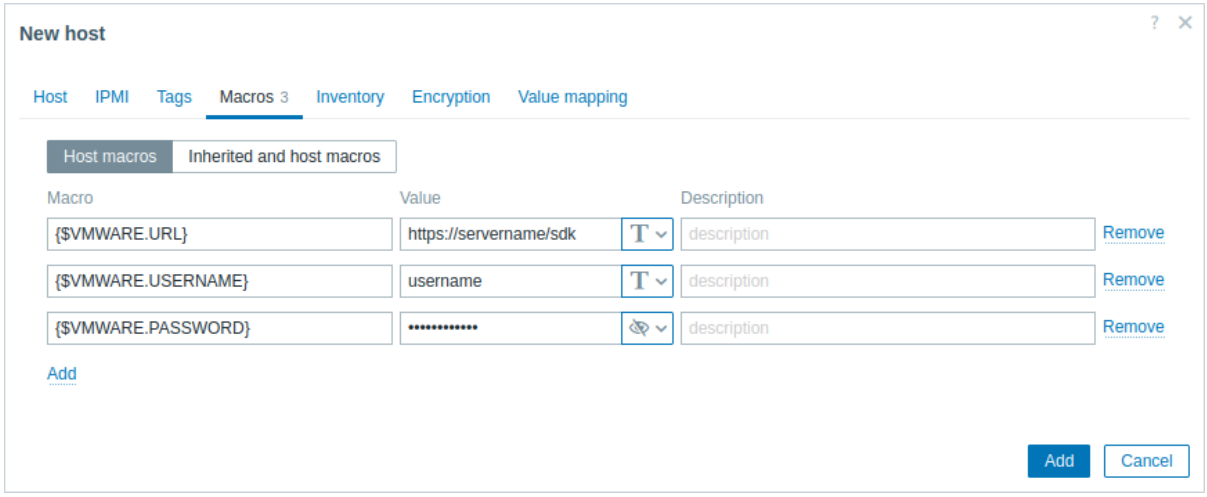
1. Connectez-vous à l'interface Zabbix.

2. Créez un hôte dans l'interface web Zabbix :

- Dans le champ *Host name*, saisissez un nom d'hôte (par exemple, "VMware environment").
- Dans le champ *Templates*, saisissez ou sélectionnez le modèle "VMware FQDN" (ou "VMware"). Pour plus d'informations sur ces modèles, consultez Surveillance des machines virtuelles.
- Dans le champ *Host groups*, saisissez ou sélectionnez un groupe d'hôtes (par exemple, un nouveau groupe d'hôtes "VMware").



- Dans l'onglet *Macros*, définissez les macros d'hôte suivantes :
  - {$VMWARE.URL} - URL SDK du service VMware (vCenter ou hyperviseur ESXi) (https://servername/sdk)
  - {$VMWARE.USERNAME} - nom d'utilisateur du service VMware
  - {$VMWARE.PASSWORD} - mot de passe de l'utilisateur {$VMWARE.USERNAME} du service VMware



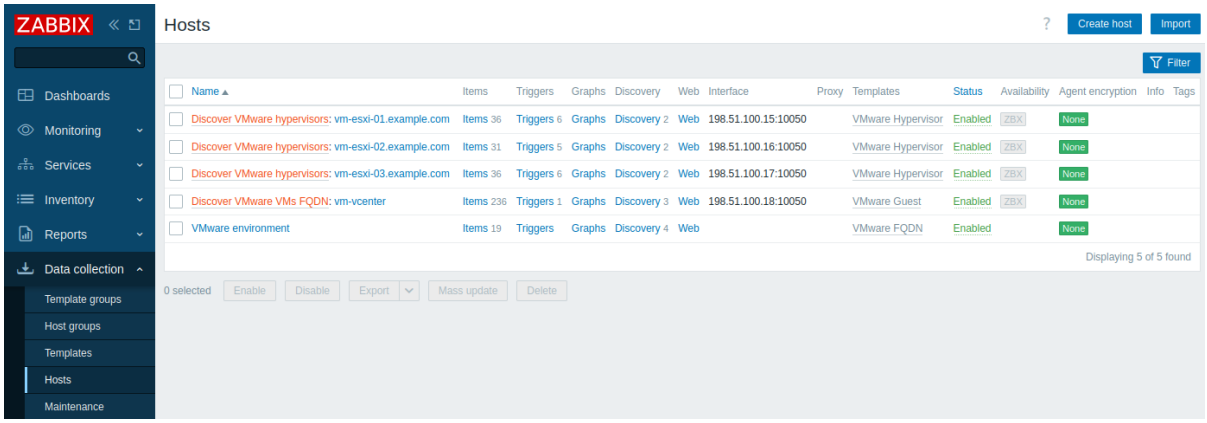
3. Cliquez sur le bouton *Add* pour créer l'hôte. Cet hôte représentera votre environnement VMware.

Afficher les métriques collectées

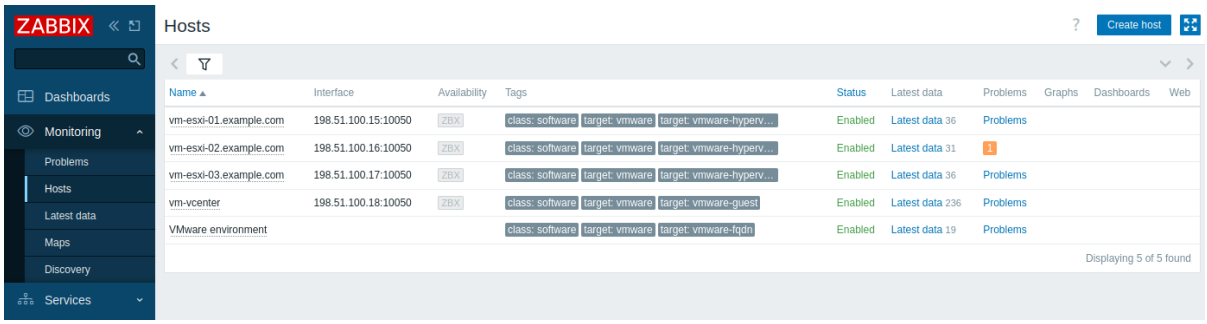
Félicitations ! À ce stade, Zabbix surveille déjà votre environnement VMware.

Selon la configuration de votre environnement VMware, Zabbix peut découvrir puis créer des hôtes pour les entités découvertes. Notez que la découverte et la création d'hôtes peuvent également être exécutées manuellement, si nécessaire.

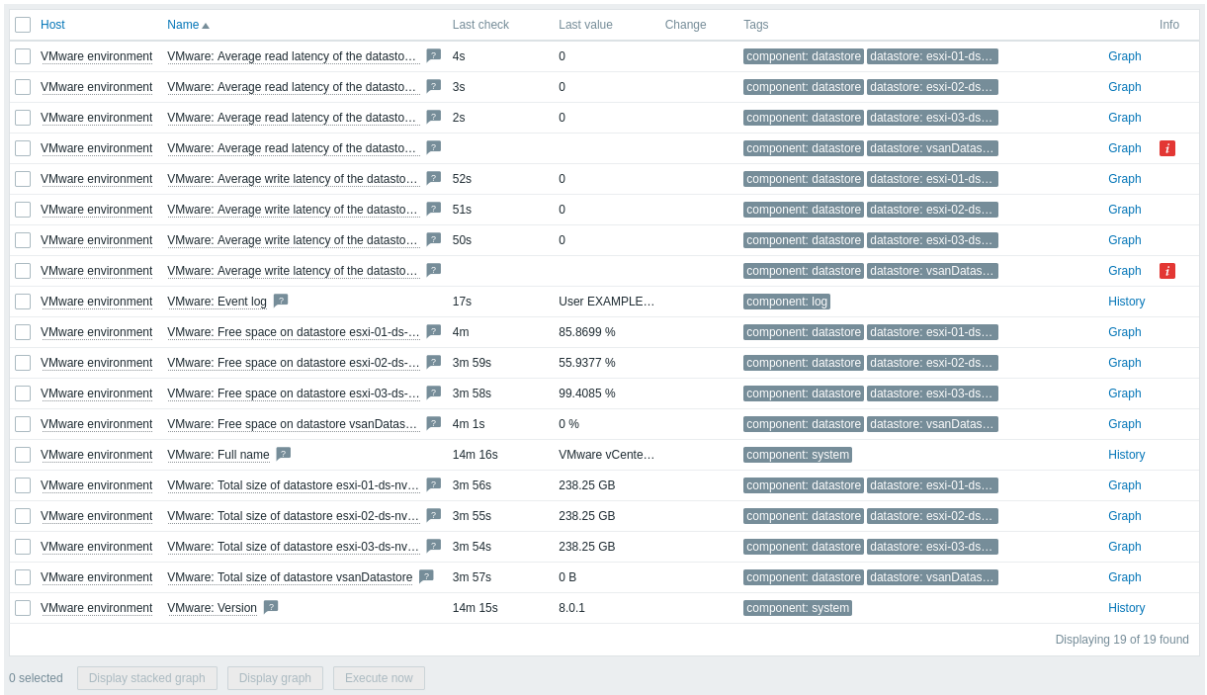
Pour afficher les hôtes créés, accédez à la section de menu *Collecte de données > Hôtes*.



Pour afficher les métriques collectées, accédez à la section de menu *Surveillance > Hôtes* et cliquez sur *Dernières données* à côté de l'hôte créé "VMware environment" ou de l'un des hôtes créés pour les entités découvertes.



Cette action ouvrira une liste de toutes les dernières métriques collectées à partir de l'hôte sélectionné.

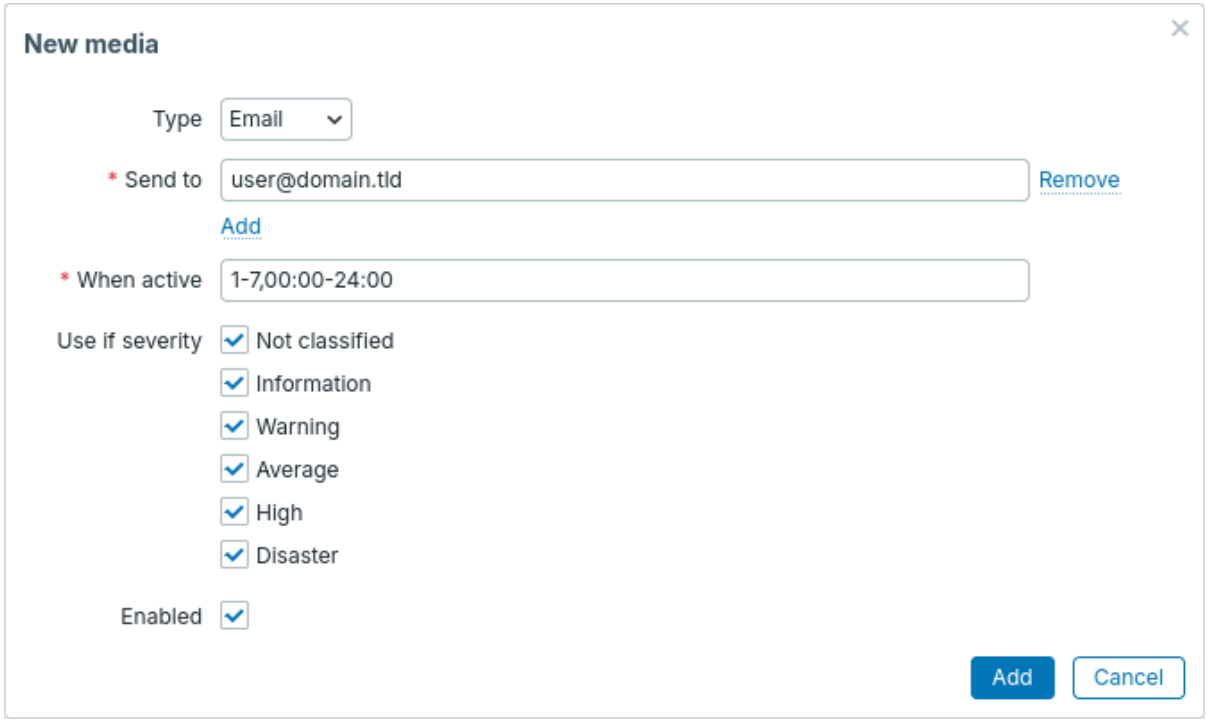


Notez que certains éléments n'ont aucune donnée et affichent l'état *Non pris en charge*. Cela est dû au fait que Zabbix ne peut pas trouver de compteurs de performance valides sur le datastore spécifique, car celui-ci n'est pas activé dans l'environnement VMware surveillé.

Configurer les alertes de problème

Zabbix peut vous avertir d'un problème dans votre infrastructure à l'aide de diverses méthodes. Ce guide fournit les étapes de configuration de base pour l'envoi d'alertes par e-mail.

1. Accédez à *Paramètres utilisateur > Profil*, ouvrez l'onglet *Media* et ajoutez votre e-mail.



2. Suivez le guide pour Recevoir une notification de problème.

La prochaine fois que Zabbix détectera un problème, vous devriez recevoir une alerte par e-mail.

Voir aussi

- Créer un élément - comment commencer à surveiller des métriques supplémentaires.
- Escalades de problème - comment créer des scénarios d'alerte en plusieurs étapes (par exemple, envoyer d'abord un message à l'administrateur système, puis, si un problème n'est pas résolu dans les 45 minutes, envoyer un message au responsable du centre de données).
- Supervision des machines virtuelles - informations supplémentaires sur la supervision VMware (processus de collecte des données, options de configuration du serveur, conseils de dépannage, etc.).
- Clés d'élément de supervision VMware - liste complète des métriques VMware pouvant être surveillées à l'aide de Zabbix.
- Modèle *VMware* - informations supplémentaires sur le modèle *VMware*.
- Modèle *VMware FQDN* - informations supplémentaires sur le modèle *VMware FQDN*.

Surveiller le trafic réseau avec Zabbix

Introduction

Cette page vous guide à travers les étapes nécessaires pour commencer la supervision de base du trafic de votre réseau avec Zabbix.

À qui ce guide est destiné

Ce guide est conçu pour les nouveaux utilisateurs de Zabbix et contient l'ensemble minimal d'étapes nécessaires pour activer la supervision de base de votre trafic réseau. Si vous recherchez des options de personnalisation avancées ou si vous avez besoin d'une configuration plus poussée, consultez la section Configuration du manuel Zabbix.

Prérequis

Avant de poursuivre ce guide, vous devez télécharger et installer le serveur Zabbix, l'interface Zabbix et l'agent Zabbix conformément aux instructions correspondant à votre système d'exploitation. Notez que vous devez installer l'agent Zabbix sur la machine qui nécessite la surveillance du trafic. Il peut s'agir du même hôte où le serveur Zabbix est installé ou d'un hôte différent.

Ce guide fournira les instructions de configuration de la surveillance du trafic réseau de l'interface *eth0* sur une machine distincte nommée *Remote host*.

Configurer Zabbix pour la supervision

L'agent Zabbix peut collecter des métriques en mode actif ou passif (simultanément). Consultez Vérifications passives et actives de l'agent pour plus d'informations. Dans ce guide, la supervision au moyen de vérifications passives sera décrite.

Configuration de l'agent Zabbix

1. Ouvrez le fichier de configuration de l'agent sur la machine où l'agent est installé :

```
sudo vi /etc/zabbix/zabbix_agentd.conf
```

2. Ajoutez l'adresse IP ou le nom DNS de votre serveur Zabbix au paramètre *Server*. Par exemple :

```
Server=192.0.2.22
```

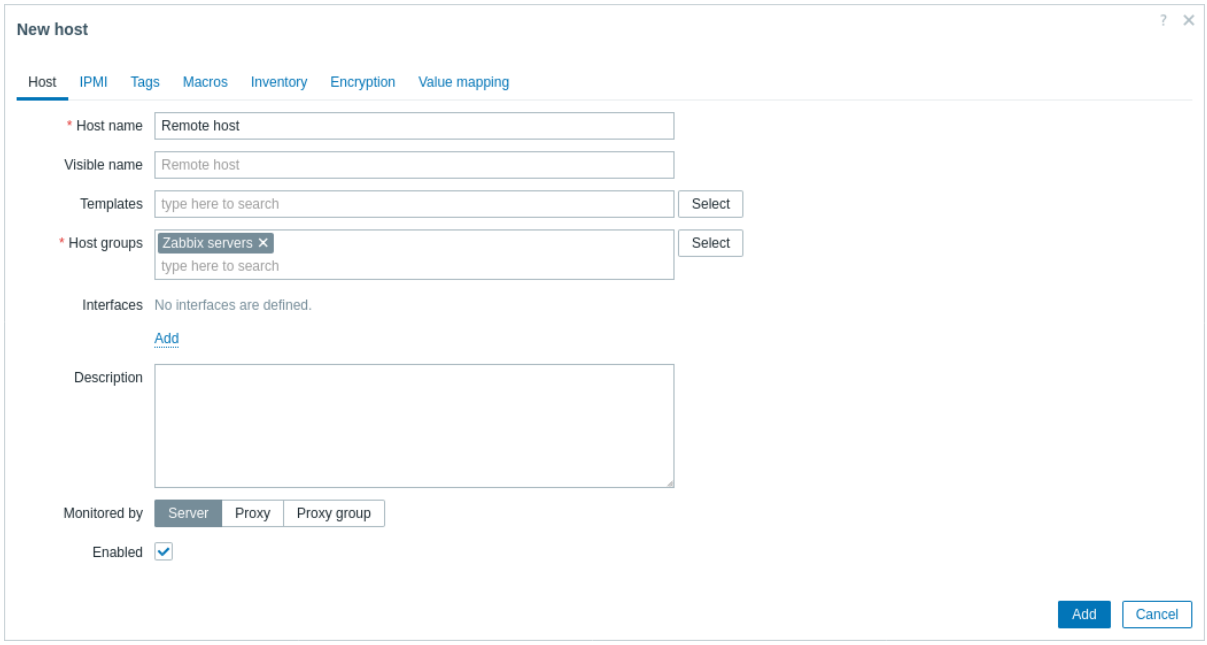
3. Redémarrez l'agent Zabbix :

```
systemctl restart zabbix-agent
```

Interface Zabbix

1. Connectez-vous à l'interface Zabbix.

2. Créez un hôte dans l'interface web Zabbix, en indiquant l'adresse IP ou le nom DNS de la machine sur laquelle l'agent est installé.

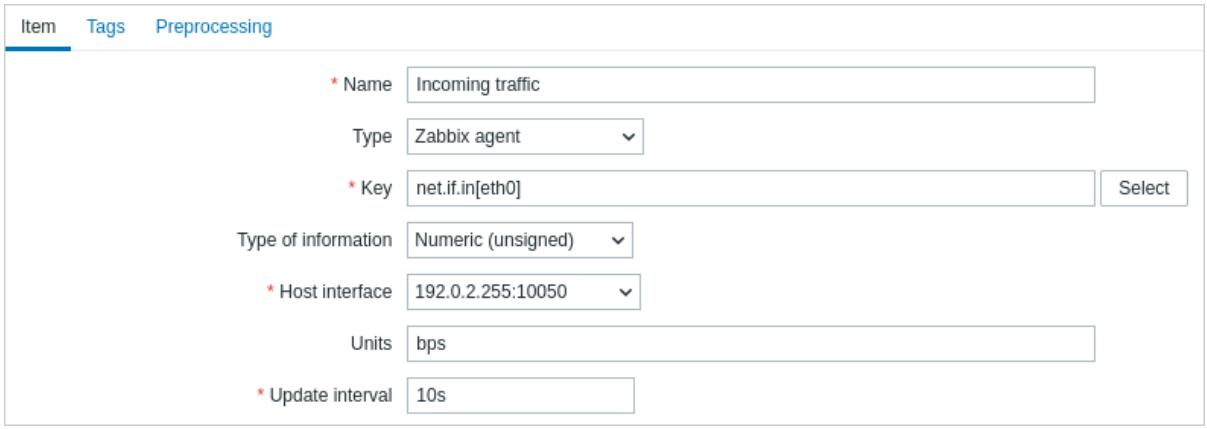


Créer des éléments

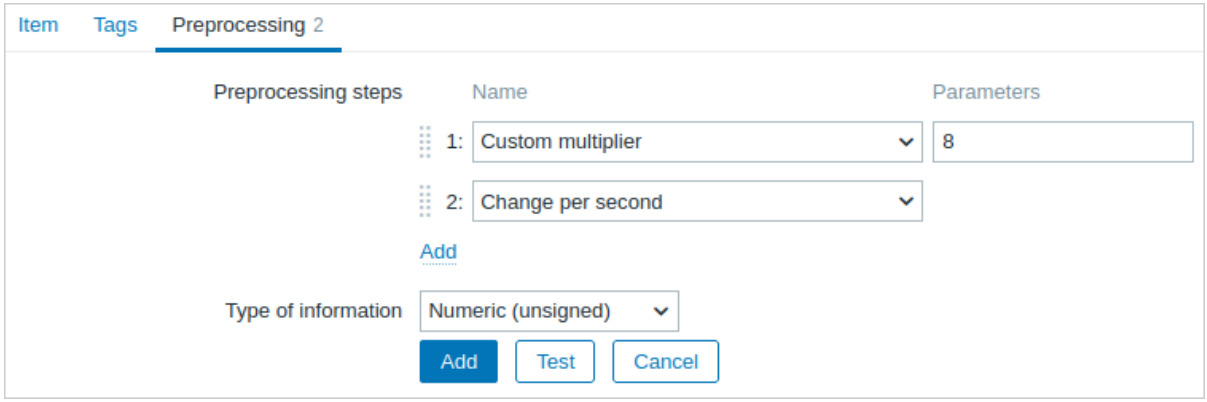
Suivez les instructions de création d'un élément pour ajouter les éléments de surveillance du trafic, à savoir :

- Trafic entrant
- Trafic sortant
- Trafic total

Une configuration simple pour l'élément de surveillance du trafic entrant ressemblerait à ceci :



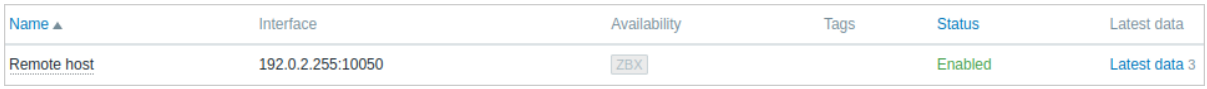
Pour rendre les données collectées adaptées à une utilisation pratique, vous pouvez définir des étapes de prétraitement lors de la création des éléments. Dans le cas présent, il peut s'agir d'une multiplication par 8 (pour convertir les octets en bits) et d'une présentation en variation par seconde.



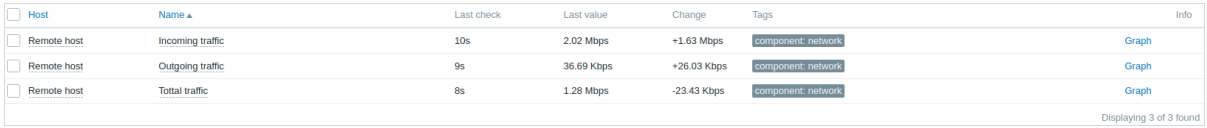
Afficher les données collectées

Félicitations ! À ce stade, Zabbix surveille déjà le trafic de votre réseau.

Pour afficher les métriques collectées, ouvrez la section de menu *Surveillance > Hôtes* et cliquez sur *Dernières données* dans la ligne du hôte.

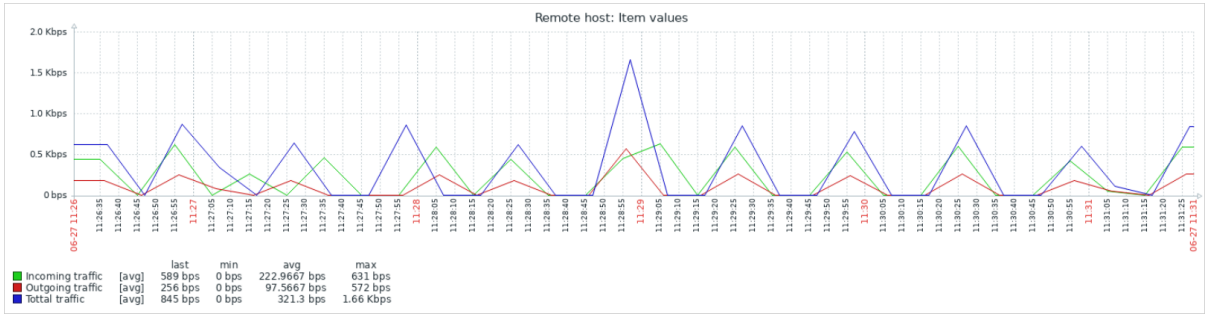


Vous verrez les données de trafic.



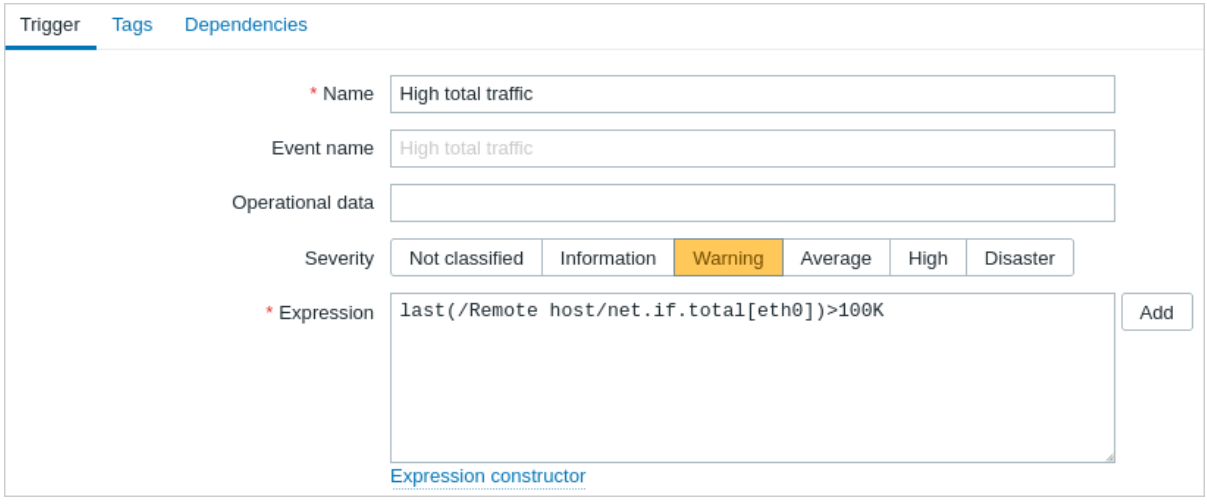
Afficher les graphiques

Les données collectées peuvent être affichées sous forme de graphiques. Pour les afficher, dans la section *Dernières données*, cliquez sur *Graphique* dans la ligne de l'élément ou sélectionnez les éléments requis et cliquez sur *Afficher le graphique* ci-dessous.

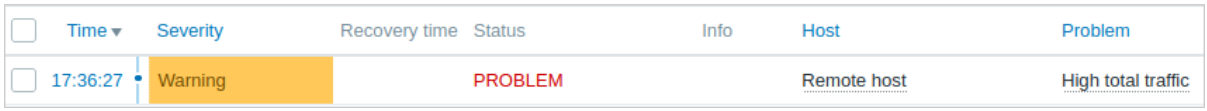


Configurer les déclencheurs

Vous pouvez définir des déclencheurs pour détecter un trafic réseau anormal. Consultez les instructions sur la configuration d'un déclencheur et ajoutez le déclencheur signalant que le trafic total est trop élevé, par exemple :



Faites maintenant dépasser au trafic le seuil que vous avez défini dans l'expression du déclencheur et accédez à *Monitoring* > *Problems* pour vérifier que le problème y est répertorié.



Configurer les alertes de problème

Il existe plusieurs façons de recevoir des notifications concernant le problème. L'e-mail étant la méthode la plus courante, suivez les instructions pour configurer une notification de problème par e-mail. Vous pouvez également choisir d'autres types de média à utiliser pour la diffusion des notifications.

Voir aussi :

- Escalades de problème - comment créer des scénarios d'alerte en plusieurs étapes (par exemple, envoyer d'abord un message à l'administrateur système, puis, si un problème n'est pas résolu dans les 45 minutes, envoyer un message au responsable du centre de données).
- Accusé de réception d'un problème - comment indiquer que le problème est connu, ajouter des commentaires sur sa résolution, supprimer ou fermer le problème.
- Surveiller Linux avec l'agent Zabbix - comment démarrer une surveillance de base des éléments les plus importants en reliant un modèle préconfiguré.

Surveiller le trafic réseau à l'aide de vérifications actives

Introduction

Cette page vous guide à travers les étapes nécessaires pour commencer la surveillance de base du trafic de votre réseau avec Zabbix à l'aide des vérifications actives.

À qui ce guide est destiné

Ce guide est conçu pour les nouveaux utilisateurs de Zabbix et contient l'ensemble minimal d'étapes nécessaires pour activer la supervision de base de votre trafic réseau à l'aide de vérifications actives. Si vous recherchez des options de personnalisation avancées ou avez besoin d'une configuration plus poussée, consultez la section Configuration du manuel Zabbix.

Prérequis

Avant de poursuivre ce guide, vous devez télécharger et installer le serveur Zabbix, l'interface Zabbix et l'agent Zabbix conformément aux instructions correspondant à votre système d'exploitation.
Notez que vous devez installer l'agent Zabbix sur la machine qui nécessite la surveillance du trafic.
Il peut s'agir du même hôte où le serveur Zabbix est installé, ou d'un autre hôte.

Ce guide fournira les instructions de configuration de la surveillance du trafic réseau de l'interface *eth0*
sur une machine distincte nommée *Remote host*.

Configurer Zabbix pour la supervision

L'agent Zabbix peut collecter des métriques en mode actif ou passif (simultanément). Consultez Contrôles d'agent passifs et actifs pour plus d'informations. Dans ce guide, la supervision au moyen de **contrôles actifs** sera décrite.

Configuration de Zabbix agent

1. Ouvrez le fichier de configuration de l'agent sur la machine où l'agent est installé.

Si vous utilisez Zabbix agent :

```
sudo vi /etc/zabbix/zabbix_agentd.conf
```

Si vous utilisez Zabbix agent 2 :

```
sudo vi /etc/zabbix/zabbix_agent2.conf
```

2. Ajoutez l'adresse IP ou le nom DNS (et le port facultatif) de votre serveur Zabbix au paramètre ServerActive. Par exemple :

```
ServerActive=192.0.2.1:10051
```

Zabbix agent utilisera cette adresse pour se connecter au port trapper du serveur Zabbix (par défaut : 10051) et demander les données de configuration pour les vérifications actives.

3. Définissez le paramètre Hostname, qui doit correspondre au nom d'hôte qui sera défini dans l'interface Zabbix. Dans notre exemple, il s'agit de :

```
Hostname=Remote host
```

La valeur de Hostname doit correspondre car, pour les vérifications actives, Zabbix agent l'utilise pour récupérer la configuration correcte de l'hôte depuis le serveur. Plus précisément, l'agent établit une connexion avec le serveur et s'identifie à l'aide de la valeur Hostname. Le serveur fournit ensuite la configuration de supervision de cet hôte. Si ces valeurs diffèrent, l'agent ne recevra pas la configuration appropriée, ce qui entraînera l'absence de métriques ou des problèmes de supervision.

4. Redémarrez Zabbix agent.

Si vous utilisez Zabbix agent :

```
systemctl restart zabbix-agent
```

Si vous utilisez Zabbix agent 2 :

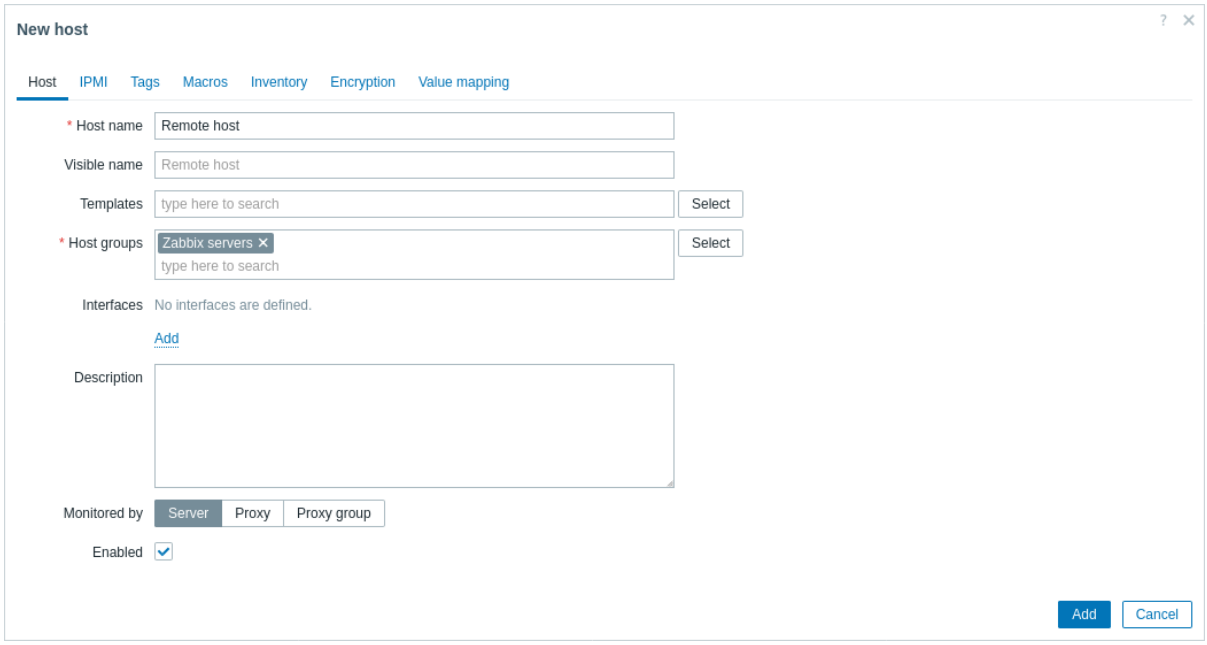
```
systemctl restart zabbix-agent2
```

Interface Zabbix

1. Connectez-vous à l'interface Zabbix.

2. Créez un hôte dans l'interface web Zabbix.

- Dans le champ *Host name*, saisissez un nom d'hôte (par exemple, "Remote host") qui correspond à la valeur du paramètre Hostname définie précédemment dans le fichier de configuration de l'agent.
- Dans le champ *Host groups*, saisissez ou sélectionnez un groupe d'hôtes (par exemple, "Zabbix servers").
- Laissez *Interfaces* non défini ; une interface n'est pas nécessaire pour les vérifications actives, car l'agent initie la connexion au serveur au lieu d'attendre une connexion du serveur.



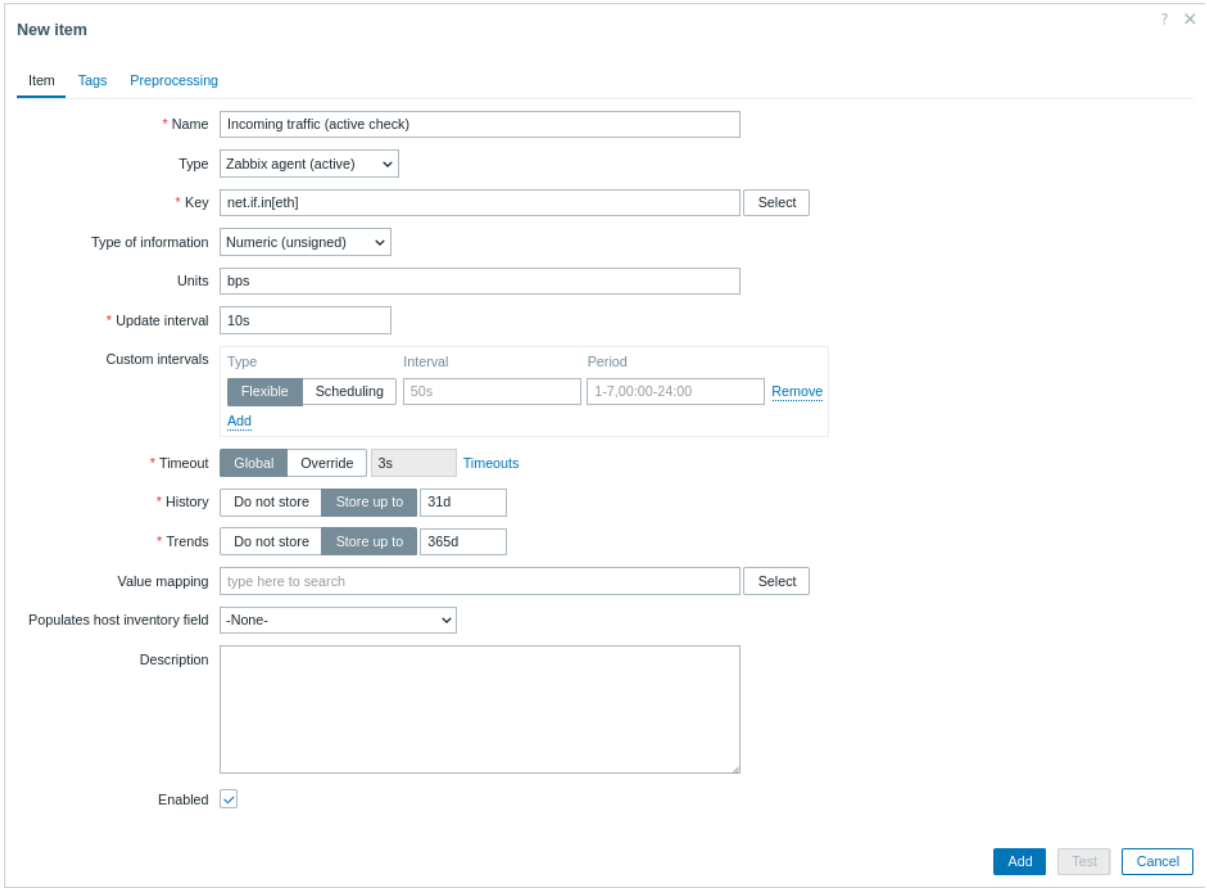
3. Cliquez sur *Add* pour ajouter l'hôte. Cet hôte représentera la machine Linux surveillée.

Créer des éléments

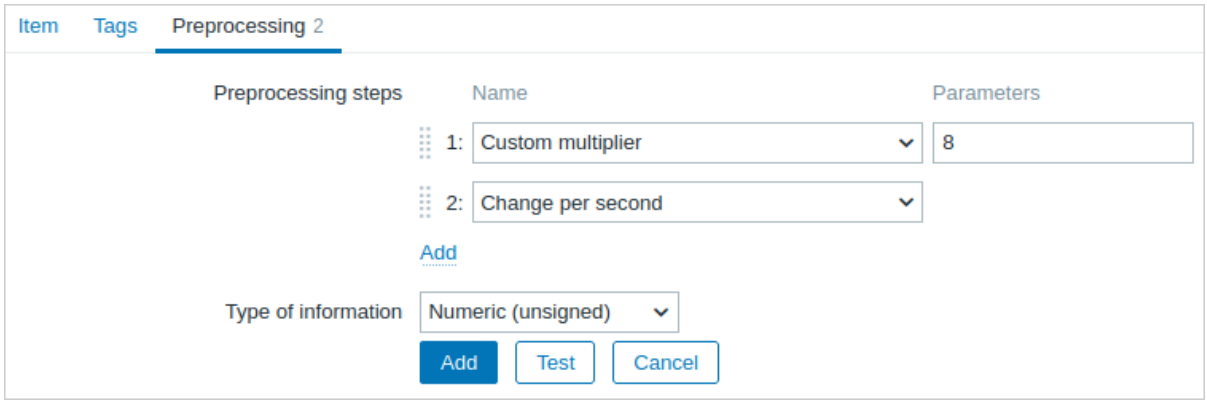
Suivez les instructions de création d'un élément pour ajouter les éléments de surveillance du trafic, à savoir :

- Trafic entrant
- Trafic sortant
- Trafic total

Une configuration simple pour l'élément de surveillance du trafic entrant avec une vérification active ressemblerait à ceci :



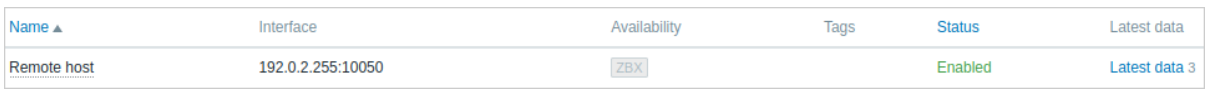
Pour rendre les données collectées adaptées à une utilisation pratique, vous pouvez définir certaines étapes de prétraitement lors de la création des éléments. Dans le cas présent, il peut s'agir d'une multiplication par 8 (pour convertir les octets en bits) et d'une présentation sous forme de variation par seconde.



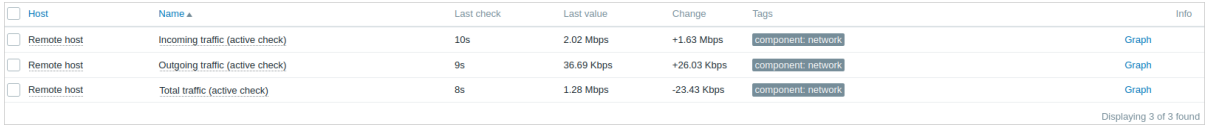
Afficher les données collectées

Félicitations ! À ce stade, Zabbix surveille déjà le trafic de votre réseau.

Pour afficher les métriques collectées, ouvrez la section de menu *Monitoring > Hosts* et cliquez sur *Latest data* dans la ligne de l'hôte.



Vous verrez les données de trafic.



Voir aussi :

- Afficher les graphiques - comment afficher les données collectées sous forme de graphiques.
- Configurer les déclencheurs - comment configurer des déclencheurs pour détecter un trafic réseau anormal.

- Configurer les alertes de problème - comment configurer des notifications concernant les situations problématiques.
- Escalades de problème - comment créer des scénarios d'alerte en plusieurs étapes (par exemple, envoyer d'abord un message à l'administrateur système, puis, si un problème n'est pas résolu en 45 minutes, envoyer un message au responsable du centre de données).
- Accusé de réception d'un problème - comment indiquer que le problème est connu, ajouter des commentaires sur sa résolution, le supprimer ou le fermer.
- Surveiller Linux avec l'agent Zabbix - comment démarrer une surveillance de base des éléments les plus importants en reliant un modèle préconfiguré.
- Installation à partir de paquets - comment installer les composants Zabbix à l'aide des paquets officiels RPM et DEB pour diverses distributions Linux, afin de garantir l'accès aux dernières fonctionnalités et corrections de bogues.

Surveiller des sites web avec des éléments Browser

Introduction

Cette page vous guide à travers les étapes nécessaires pour commencer la surveillance de base des sites web avec des éléments Browser.

À qui ce guide est destiné

Ce guide est conçu pour les nouveaux utilisateurs de Zabbix et contient l'ensemble minimal d'étapes nécessaires pour activer la supervision de base des sites web avec des éléments Browser. Si vous recherchez des options de personnalisation avancées ou avez besoin d'une configuration plus poussée, consultez la page Éléments Browser ou la section Configuration du manuel Zabbix.

Prérequis

Avant de poursuivre ce guide, vous devez télécharger et installer le serveur Zabbix et l'interface Zabbix conformément aux instructions correspondant à votre système d'exploitation.

Selon votre environnement, certaines étapes de ce guide peuvent légèrement différer. Ce guide est basé sur un environnement exécutant Ubuntu.

Configurer WebDriver

Les éléments de navigateur nécessitent un framework d'automatisation (soit Selenium Server, soit un WebDriver simple, par exemple ChromeDriver) comme point de terminaison de test Web qui contrôle un navigateur et interagit avec lui, en exécutant des commandes de test telles que cliquer sur des boutons ou saisir du texte. À titre d'exemple, ce guide utilisera Selenium Server avec Chrome dans un conteneur Docker.

Il est supposé que Docker est déjà configuré. Ce guide ne couvre pas la configuration de Docker. Pour les instructions d'installation, voir Installer Docker Engine sur Ubuntu.

1. Lancez Selenium Server avec Chrome dans un conteneur Docker avec les options suivantes :

- **docker run --name browser** - lance un nouveau conteneur Docker nommé "browser" ;
- **-p 4444:4444** - mappe le port 4444 de votre machine hôte vers le port 4444 du conteneur (c'est le port utilisé par Selenium Server pour accepter les commandes) ;
- **-p 7900:7900** - mappe le port 7900 de votre machine hôte vers le port 7900 du conteneur (c'est le port utilisé par le serveur Virtual Network Computing (VNC), ce qui vous permet de voir l'interface graphique du navigateur à distance ; nécessite un client VNC) ;
- **--shm-size="2g"** - alloue 2 Go de mémoire partagée au conteneur (cela est important pour que Chrome fonctionne correctement, car il peut nécessiter une quantité importante de mémoire partagée pour éviter les plantages) ;
- **-d** - exécute le conteneur en mode détaché, ce qui signifie qu'il s'exécutera en arrière-plan ;
- **selenium/standalone-chrome:latest** - spécifie l'image Docker à utiliser ; dans ce cas, la dernière version de Selenium Server avec Chrome.

```
docker run --name browser \
-p 4444:4444 \
-p 7900:7900 \
--shm-size="2g" \
-d selenium/standalone-chrome:latest
```

2. Vérifiez que le conteneur Docker `browser` est en cours d'exécution et accessible.

- Récupérez l'adresse IP du conteneur (dans cet exemple, 192.0.2.1) :

```
ip addr

##### 1: lo: <LOOPBACK,UP,LOWER_UP>
#####     ...
```

```
##### 3: docker0: <NO-CARRIER,BROADCAST,MULTICAST,UP> ...
#####    inet 192.0.2.1/16 brd 192.0.255.255 scope global docker0
#####    ...
```

- Testez la connexion au conteneur avec Ncat :

```
nc -zv 192.0.2.1 4444

##### Connection to 192.0.2.1 4444 port [tcp/*] succeeded!
```

- Récupérez le contenu de la page Web depuis Selenium Server avec curl :

```
curl -L 192.0.2.1:4444

##### <!DOCTYPE html>
##### <html lang="en">
#####
##### <head>
#####   <meta charset="utf-8"/>
#####   <link href="favicon.svg" rel="icon" type="image/svg">
#####   <meta content="width=device-width, initial-scale=1" name="viewport"/>
#####   <link href="logo192.png" rel="apple-touch-icon"/>
#####   <link href="manifest.json" rel="manifest"/>
#####   <title>Selenium Grid</title>
##### </head>
#####
##### <body>
##### ...
```

Note:
Pour le dépannage, consultez la documentation Docker.

Configurer le serveur Zabbix

Les éléments de navigateur sont exécutés et traités par les processus *browser poller* de Zabbix, qui doivent être activés en ajustant le paramètre de configuration du serveur StartBrowserPollers. De plus, le paramètre WebDriverURL doit spécifier le point de terminaison de test web configuré précédemment.

Par défaut, le paramètre StartBrowserPollers est défini sur 1. Vous devez donc uniquement spécifier le point de terminaison de test web.

1. Ouvrez le fichier de configuration du serveur Zabbix :

```
vi /etc/zabbix/zabbix_server.conf
```

2. Recherchez et définissez le paramètre WebDriverURL dans le fichier de configuration du serveur Zabbix :

```
###### Option: WebDriverURL
#####WebDriver interface HTTP(S) URL. For example, http://localhost:4444 used with Selenium WebDriver star
#####
##### Mandatory: no
##### Default:
##### WebDriverURL=

WebDriverURL=192.0.2.1:4444
```

3. Redémarrez le serveur Zabbix.

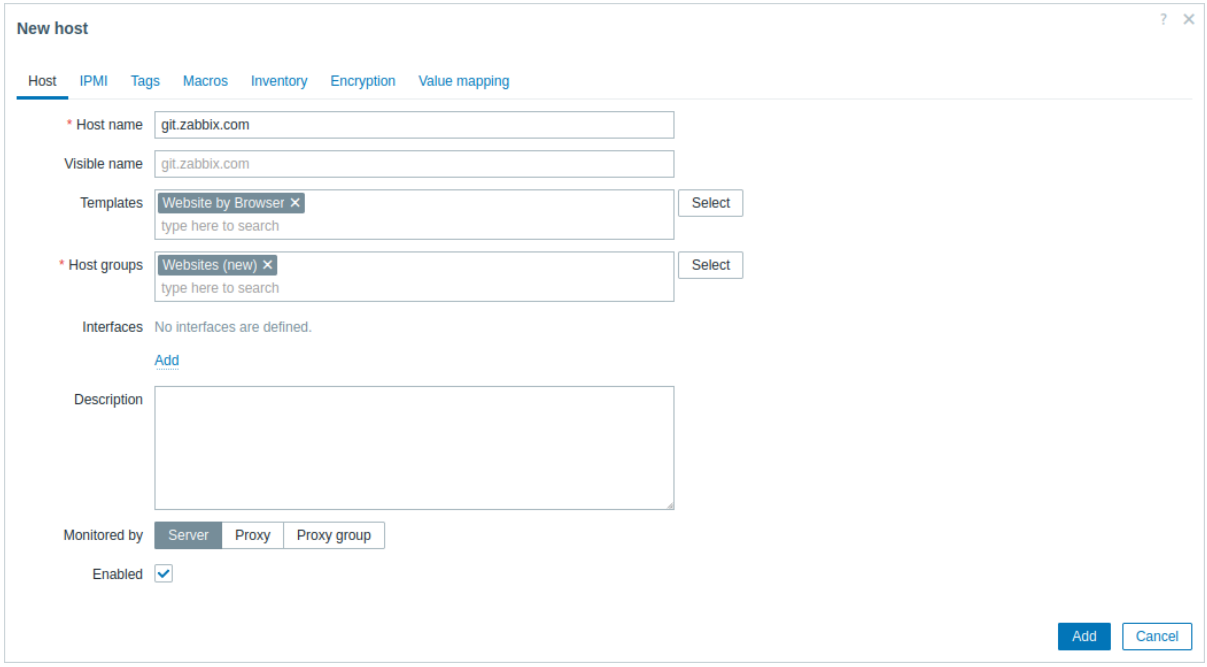
```
systemctl restart zabbix-server
```

Configurer l'interface Zabbix

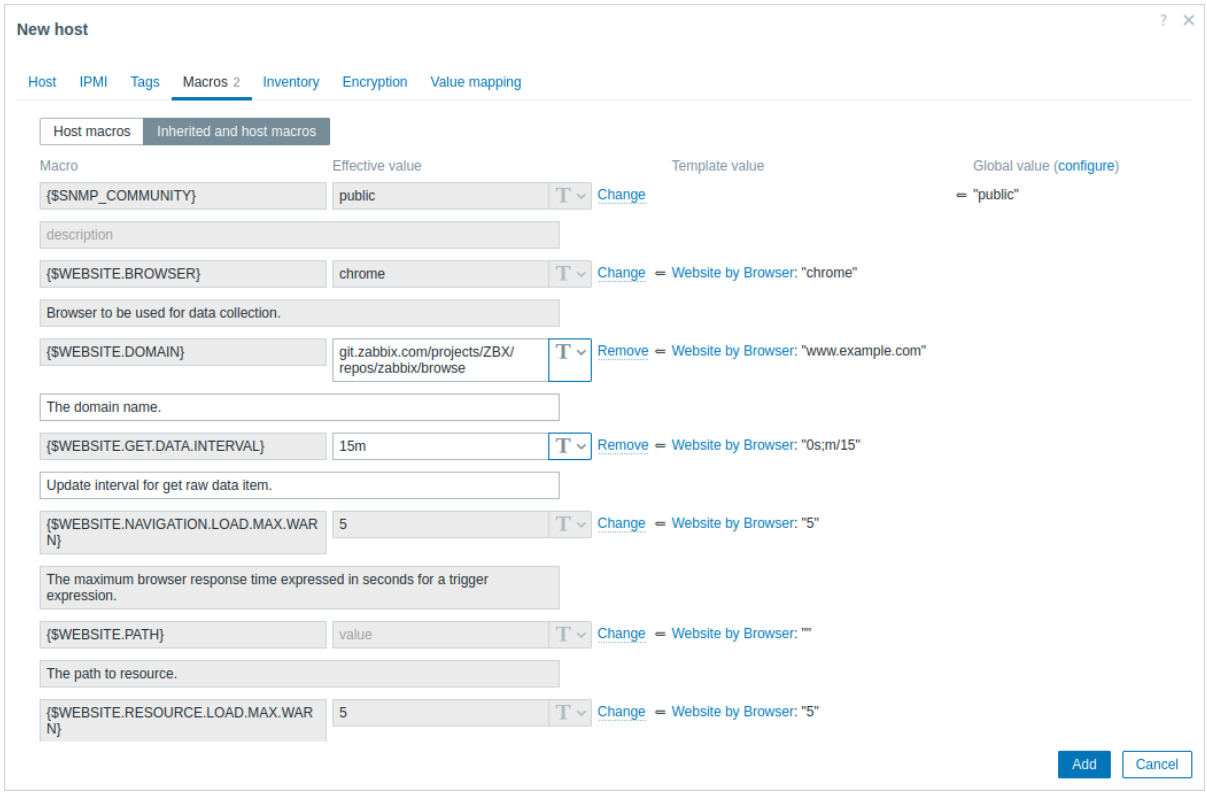
1. Connectez-vous à l'interface Zabbix.

2. Créez un hôte dans l'interface web Zabbix :

- Dans le champ *Host name*, saisissez un nom d'hôte (par exemple, "git.zabbix.com").
- Dans le champ *Templates*, tapez ou sélectionnez le modèle "Website by Browser". Pour plus d'informations sur ce modèle, consultez Website by Browser.
- Dans le champ *Host groups*, tapez ou sélectionnez un groupe d'hôtes (par exemple, un nouveau groupe d'hôtes "Websites").



- Dans l'onglet *Macros*, passez à *Inherited and host macros*, recherchez les macros suivantes et cliquez sur *Change* à côté de la valeur de la macro pour la mettre à jour :
  - {$WEBSITE.DOMAIN} - Nom de domaine (par exemple, git.zabbix.com/projects/ZBX/repos/zabbix/browse)
  - {$WEBSITE.GET.DATA.INTERVAL} - Intervalle de mise à jour des données de l'élément (par exemple, 15m)

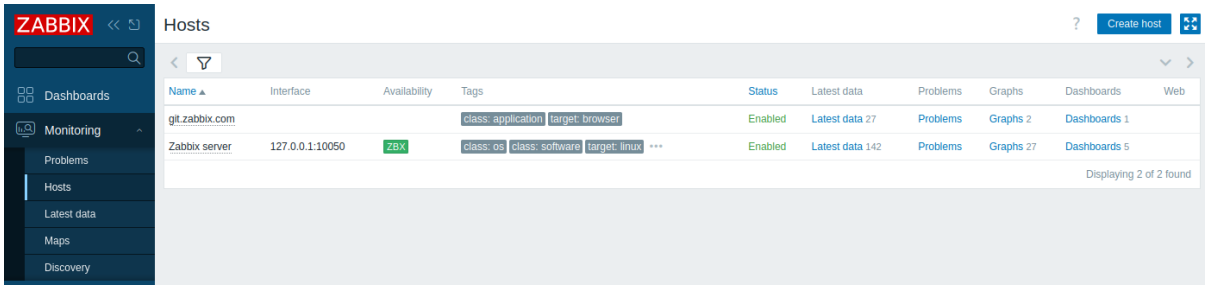


3. Cliquez sur le bouton *Add* pour créer l'hôte. Cet hôte représentera le site web que vous souhaitez surveiller.

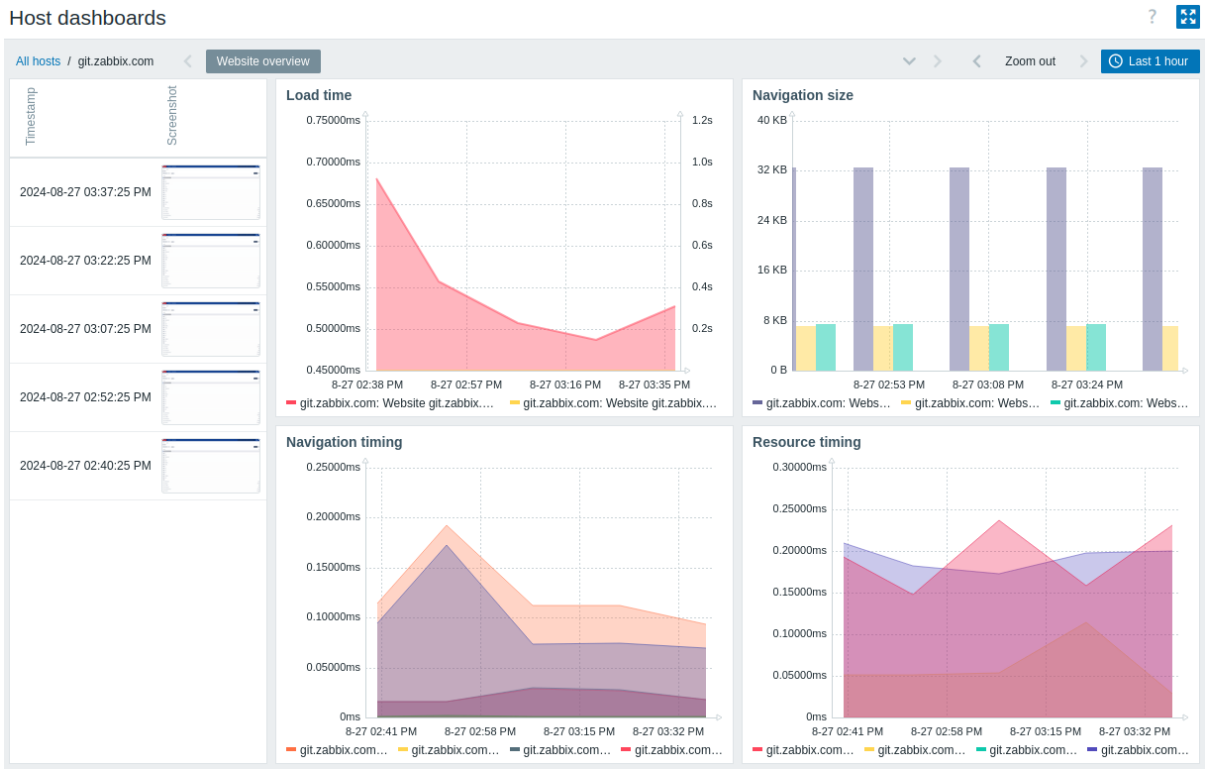
Afficher les métriques collectées

Félicitations ! À ce stade, Zabbix surveille déjà le site web que vous avez spécifié.

Pour afficher les métriques collectées, accédez à la section de menu *Monitoring > Hosts* et cliquez sur *Dashboards* à côté de l'hôte.



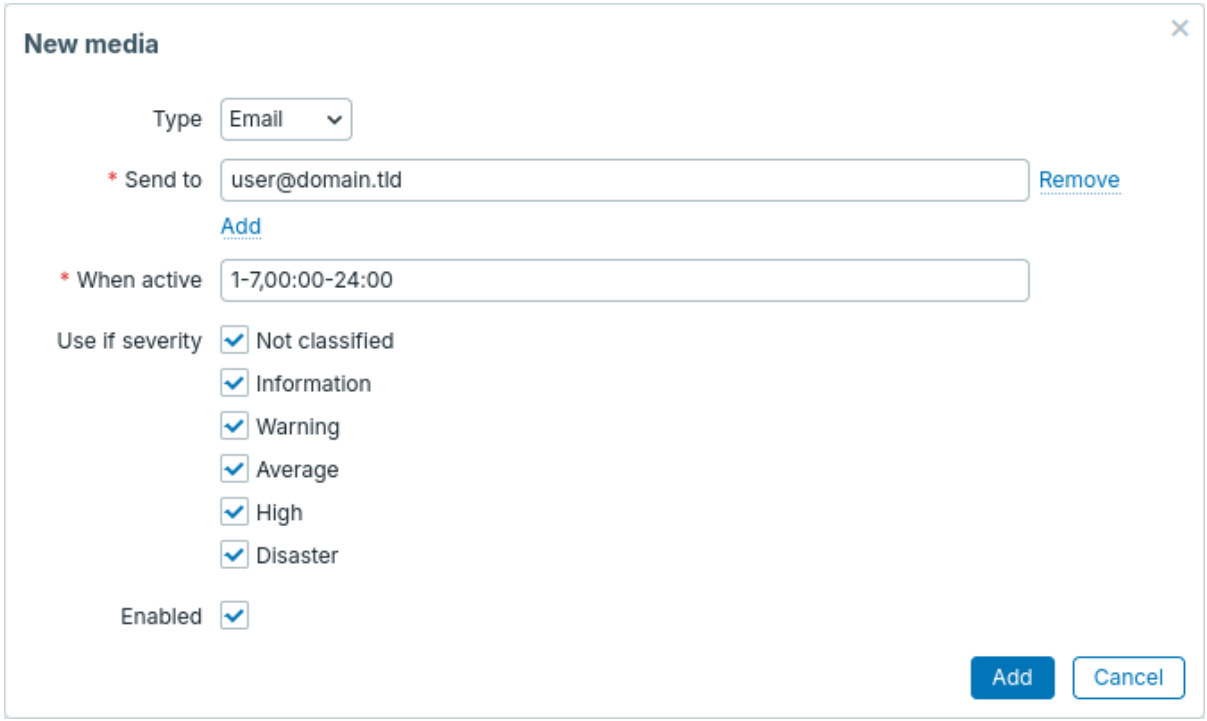
Cette action vous amènera au tableau de bord de l'hôte (configuré au niveau du modèle) avec les métriques les plus importantes collectées à partir du site web.



Configurer les alertes de problème

Zabbix peut vous avertir d'un problème dans votre infrastructure à l'aide de diverses méthodes. Ce guide fournit les étapes de configuration de base pour l'envoi d'alertes par e-mail.

1. Accédez à *Paramètres utilisateur* > *Profil*, ouvrez l'onglet *Média* et ajoutez votre adresse e-mail.



2. Suivez le guide pour Recevoir une notification de problème.

La prochaine fois que Zabbix détectera un problème, vous devriez recevoir une alerte par e-mail.

Testez votre configuration

Pour tester votre configuration, nous pouvons simuler un problème réel en mettant à jour la configuration de l'hôte dans l'interface Zabbix.

1. Ouvrez la configuration de l'hôte de votre site web dans Zabbix.

2. Accédez à l'onglet *Macros* et sélectionnez *Macros héritées et de l'hôte*.

3. Cliquez sur *Modifier* à côté, par exemple, de la valeur de macro {$WEBSITE.DOMAIN} configurée précédemment, puis définissez un nom de domaine incorrect (par exemple, /git.zabbix.com/projects/ZBX/repos/zabbix/browse).

4. Cliquez sur *Mettre à jour* pour mettre à jour la configuration de l'hôte.

5. Au bout de quelques instants, Zabbix détectera le problème "Failed to get JSON of the requested website", car il ne pourra pas se connecter au site web spécifié. Le problème apparaîtra dans *Surveillance > Problèmes*.



Si des alertes sont configurées, vous recevrez également la notification du problème.

6. Remettez la valeur de la macro à sa valeur précédente pour résoudre le problème et continuer à surveiller le site web.

Voir aussi

- Création d'un élément - comment commencer à surveiller des métriques supplémentaires.
- Escalades de problème - comment créer des scénarios d'alerte en plusieurs étapes (par exemple, envoyer d'abord un message à l'administrateur système, puis, si un problème n'est pas résolu dans les 45 minutes, envoyer un message au responsable du centre de données).
- Éléments de navigateur - comment configurer les éléments de navigateur.
- Modèle Website by Browser - informations supplémentaires sur le modèle *Website by Browser*.

Surveiller les certificats de site web avec Zabbix agent 2 (passif)

Introduction

Ce guide fournit un aperçu complet de la configuration et de la surveillance des certificats SSL/TLS à l'aide de la clé `web.certificate.get` dans Zabbix agent 2. Il est conçu pour simplifier la surveillance des certificats pour un ou plusieurs sites web, en permettant aux administrateurs d'identifier rapidement d'éventuels problèmes tels que des certificats expirés ou non valides.

À qui ce guide s'adresse

Ce guide est conçu pour les nouveaux utilisateurs de Zabbix et contient l'ensemble minimal d'étapes nécessaires pour activer la supervision de base des certificats de site web. Si vous recherchez des options de personnalisation avancées ou avez besoin d'une configuration plus poussée, consultez la section Configuration du manuel de Zabbix.

Prérequis

Avant de poursuivre ce guide, vous devez télécharger et installer le serveur Zabbix, l'interface Zabbix et l'agent 2 Zabbix conformément aux instructions pour votre système d'exploitation. Ce tutoriel suppose que le serveur Zabbix et l'agent sont tous deux installés sur la même machine ; par conséquent, `127.0.0.1` est utilisé dans la configuration.

Configurer Zabbix agent 2

1. Ouvrez le fichier de configuration de l'agent Zabbix :

```
sudo vi /etc/zabbix/zabbix_agent2.conf
```

2. Définissez le paramètre Server sur 127.0.0.1, puisque l'agent et le serveur s'exécutent sur la même machine :

```
Server=127.0.0.1
```

3. Enregistrez le fichier et redémarrez le service Zabbix agent 2 :

```
sudo systemctl restart zabbix-agent2
```

4. Après avoir installé et configuré Zabbix agent 2, testez sa disponibilité avec :

```
zabbix_get -s 127.0.0.1 -k web.certificate.get[<website_DNS_name>]
```

Zabbix agent 2 inclut le plugin WebCertificate par défaut ; aucune installation ni configuration distincte n'est donc requise.

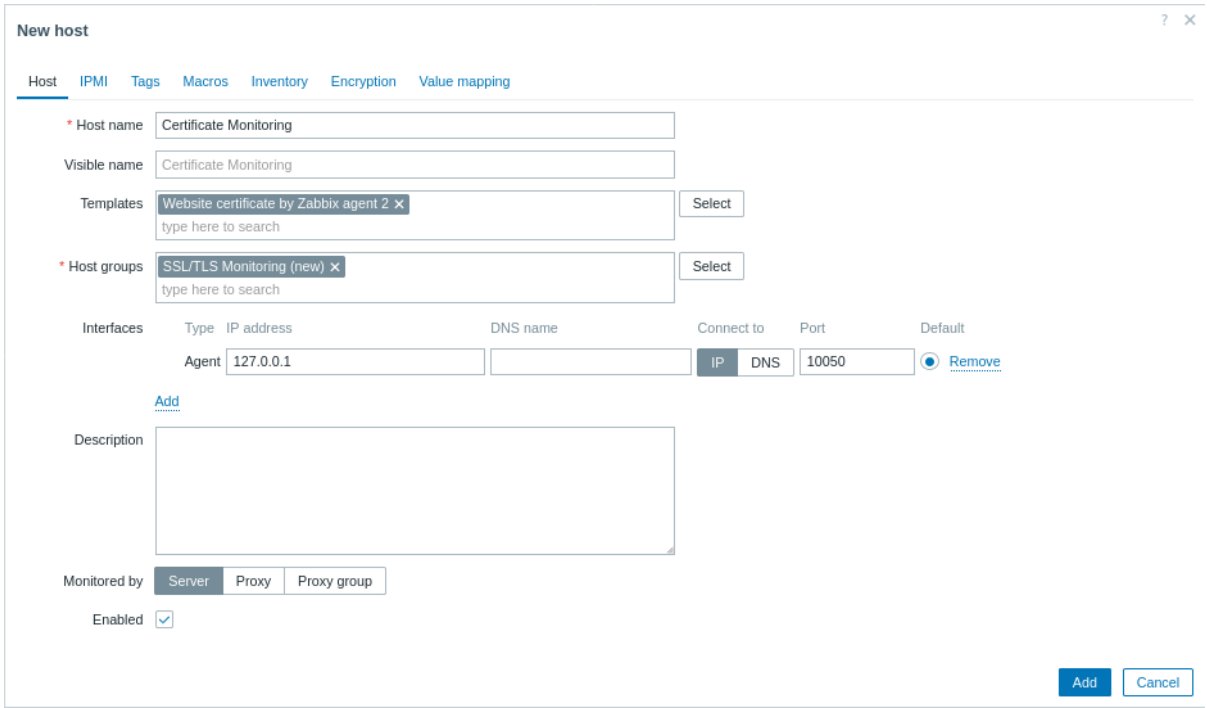
Configurer l'interface Zabbix

1. Connectez-vous à l'interface Zabbix.

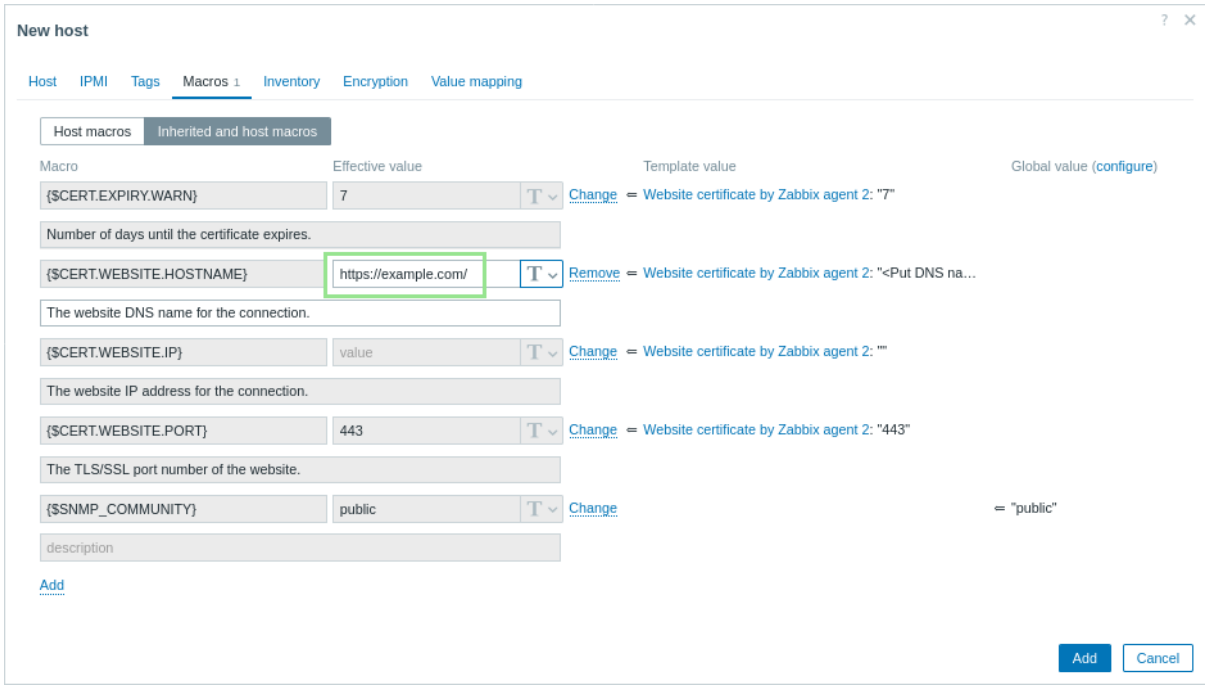
2. Accédez à *Monitoring* > *Hosts*.

3. Cliquez sur un hôte existant sur lequel vous souhaitez surveiller les certificats de site web, ou créez un hôte si nécessaire :

- Dans le champ *Host name*, saisissez un nom d'hôte (par exemple, "Certificate Monitoring").
- Dans le champ *Templates*, saisissez ou sélectionnez le modèle "Website certificate by Zabbix agent 2" qui sera lié à l'hôte.
- Dans le champ *Host groups*, saisissez ou sélectionnez un groupe d'hôtes (par exemple, "SSL/TLS Monitoring").
- Dans le champ *Interfaces*, ajoutez une interface de type "Agent" et spécifiez une adresse IP. Pour cet exemple, "127.0.0.1" est utilisé.



- Dans l'onglet *Macros*, passez à *Inherited and host macros*, recherchez les macros suivantes et cliquez sur *Change* à côté de la valeur de la macro pour la mettre à jour :
  - `{$CERT.WEBSITE.HOSTNAME}` - saisissez comme valeur le nom DNS du site web souhaité.



4. Cliquez sur *Add* pour ajouter l'hôte.

Pour surveiller plusieurs sites web, vous pouvez spécifier une liste de leurs noms d'hôte séparés par des virgules dans la macro `{$CERT.WEBSITE.HOSTNAME}`. En option, vous pouvez également fournir une liste de ports séparés par des virgules dans la macro `{$CERT.WEBSITE.PORT}`, où chaque port correspond au nom d'hôte respectif dans l'ordre. Par exemple, pour surveiller example.com et example.org sur le port 8443 :

- `{$CERT.WEBSITE.HOSTNAME}` : example.com,example.org
- `{$CERT.WEBSITE.PORT}` : 443,8443

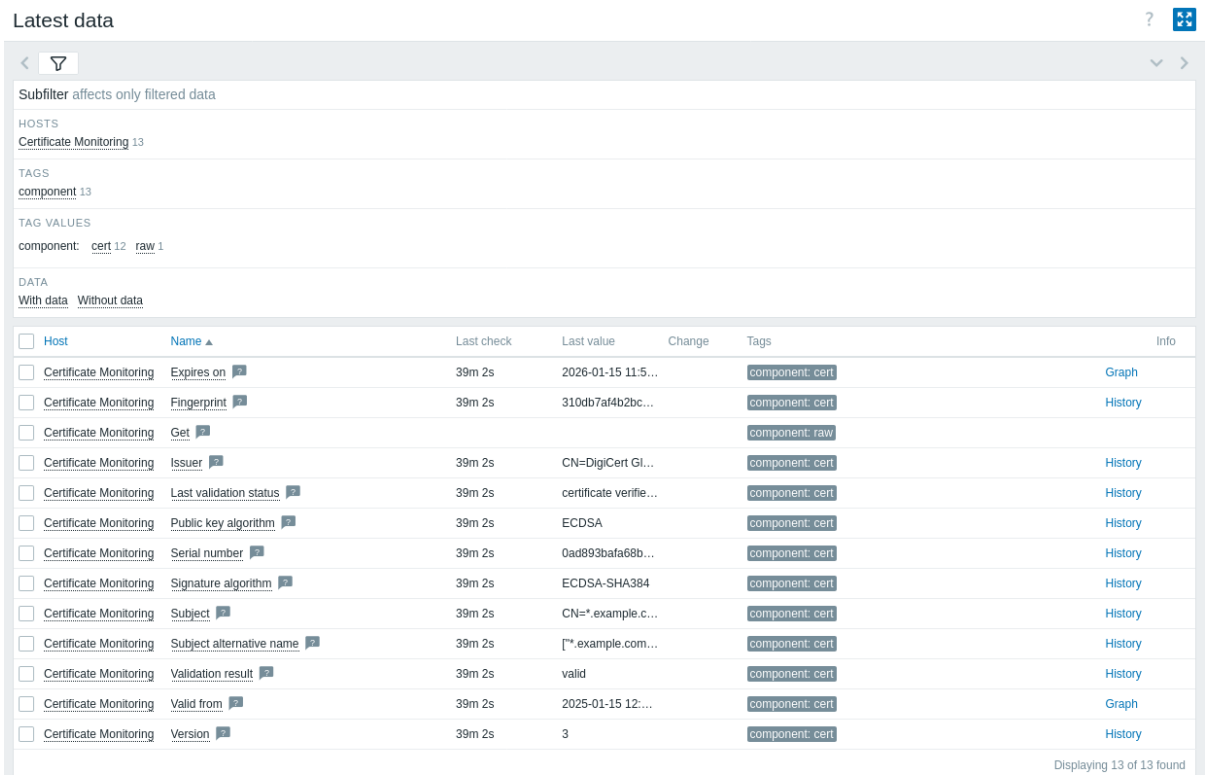
Pour chaque site web spécifié dans la macro `{$CERT.WEBSITE.HOSTNAME}`, Zabbix créera un ensemble correspondant d'éléments et de déclencheurs. Cela permet une surveillance et des alertes individuelles pour le certificat SSL de chaque site web.

Host	Name ▲	Last check	Last value
Certificate Monitoring	Get data	11m 16s	{"error":{"message…
Certificate Monitoring	[example.com]: Expires on	11m 6s	2026-01-15 11:59:…
Certificate Monitoring	[example.com]: Fingerprint	11m 6s	310db7af4b2bc90…
Certificate Monitoring	[example.com]: Get		
Certificate Monitoring	[example.com]: Issuer	11m 6s	CN=DigiCert Glob…
Certificate Monitoring	[example.com]: Last validation status	11m 6s	certificate verified …
Certificate Monitoring	[example.com]: Public key algorithm	11m 6s	ECDSA
Certificate Monitoring	[example.com]: Serial number	11m 6s	0ad893bafa68b0b…
Certificate Monitoring	[example.com]: Signature algorithm	11m 6s	ECDSA-SHA384
Certificate Monitoring	[example.com]: Subject	11m 6s	CN=*.example.co…
Certificate Monitoring	[example.com]: Subject alternative name	11m 6s	["*.example.com","…
Certificate Monitoring	[example.com]: Validation result	11m 6s	valid
Certificate Monitoring	[example.com]: Valid from	11m 6s	2025-01-15 12:00:…
Certificate Monitoring	[example.com]: Version	11m 6s	3
Certificate Monitoring	[example.org]: Expires on	10m 52s	2026-01-15 11:59:…
Certificate Monitoring	[example.org]: Fingerprint	10m 52s	3b451cfce915637…
Certificate Monitoring	[example.org]: Get		
Certificate Monitoring	[example.org]: Issuer	10m 52s	CN=DigiCert Glob…
Certificate Monitoring	[example.org]: Last validation status	10m 52s	certificate verified …
Certificate Monitoring	[example.org]: Public key algorithm	10m 52s	ECDSA
Certificate Monitoring	[example.org]: Serial number	10m 52s	0722a749b558476…
Certificate Monitoring	[example.org]: Signature algorithm	10m 52s	ECDSA-SHA384
Certificate Monitoring	[example.org]: Subject	10m 52s	CN=*.example.org…
Certificate Monitoring	[example.org]: Subject alternative name	10m 52s	["*.example.org","e…
Certificate Monitoring	[example.org]: Validation result	10m 52s	valid
Certificate Monitoring	[example.org]: Valid from	10m 52s	2025-01-15 12:00:…

Afficher les métriques collectées

Félicitations ! À ce stade, Zabbix surveille déjà le certificat web souhaité.

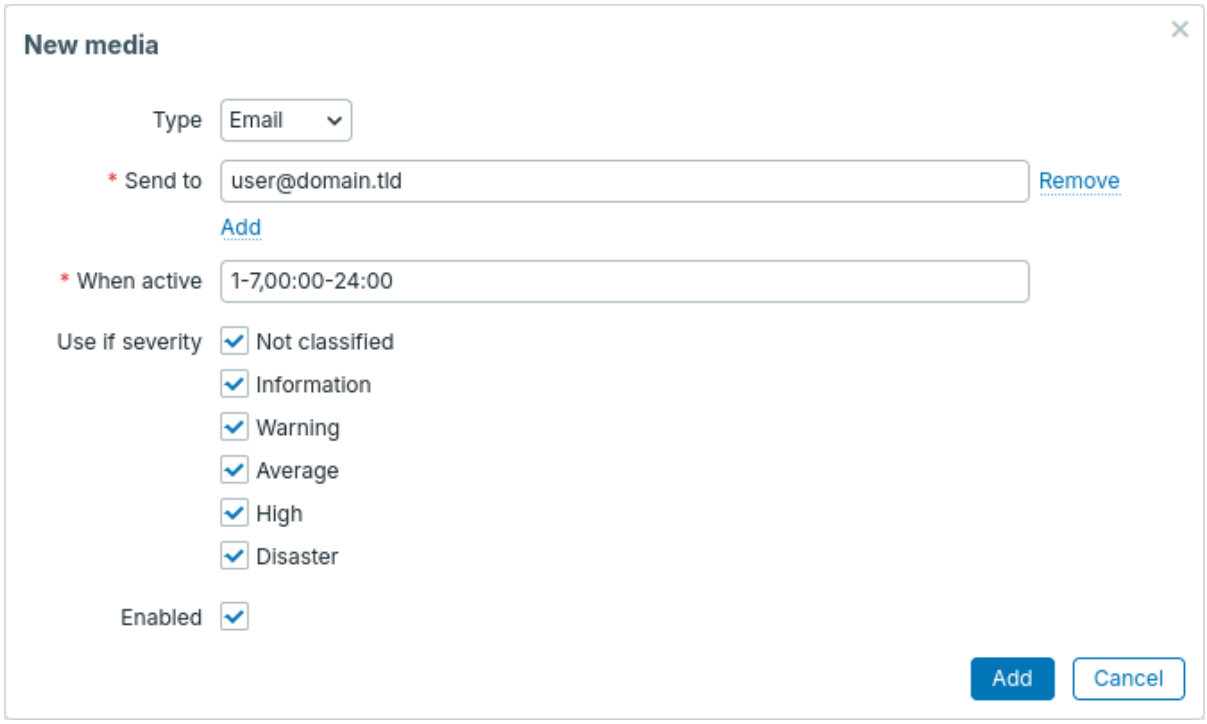
Pour afficher les métriques collectées, accédez à la section de menu *Monitoring > Hosts* et cliquez sur *Latest data* à côté de l'hôte pour afficher toutes les dernières métriques collectées dans une liste, comme la date d'expiration, l'émetteur et le sujet.



Configurer les alertes de problème

Zabbix peut vous avertir des problèmes d'infrastructure. Ce guide fournit les étapes de configuration de base pour l'envoi d'alertes par e-mail.

1. Accédez à *Paramètres utilisateur > Profil*, ouvrez l'onglet *Media* et ajoutez votre adresse e-mail.



2. Suivez le guide pour Recevoir une notification de problème.

La prochaine fois que Zabbix détectera un problème, vous devriez recevoir une alerte par e-mail.

Testez votre configuration

Pour tester votre configuration, nous pouvons simuler un problème réel en mettant à jour la configuration de l'hôte dans l'interface Zabbix.

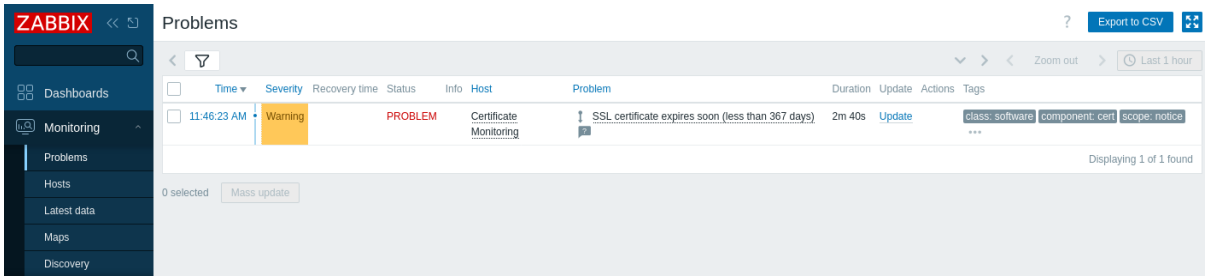
1. Ouvrez la configuration de votre hôte "Certificate Monitoring" dans Zabbix.

2. Accédez à l'onglet *Macros* et sélectionnez *Inherited and host macros*.

3. Cliquez sur *Change* à côté de la valeur de macro `{$CERT.EXPIRY.WARN}` précédemment configurée et définissez un nombre de jours très élevé (plus de 365 jours devrait suffire) afin de recevoir un avertissement avant l'expiration du certificat.

4. Cliquez sur *Update* pour mettre à jour la configuration de l'hôte.

5. Dans quelques instants, Zabbix détectera le problème "SSL certificate expires soon", avec le nombre de jours restants avant l'expiration. Le problème apparaîtra dans *Monitoring > Problems*.



Si les alertes sont configurées, vous recevrez également la notification du problème.

6. Rétablissez la valeur de la macro à sa valeur précédente pour résoudre le problème et continuer à surveiller les valeurs du certificat.

Voir aussi

- Zabbix agent 2 - répertorie les clés d'élément.
- Modèle *Website certificate by Zabbix agent 2* - informations supplémentaires sur le modèle *Website certificate by Zabbix agent 2*.
- Modèle *Website certificate by Zabbix agent 2 active* - informations supplémentaires sur le modèle *Website certificate by Zabbix agent 2 active*.
- Surveiller des sites web avec des éléments Browser - comment démarrer la surveillance de base des sites web avec des éléments Browser.

Surveiller un commutateur réseau ou un routeur avec Zabbix

Introduction

Ce guide vous accompagne à travers les étapes nécessaires pour commencer une surveillance de base de votre commutateur réseau ou routeur à l'aide de Zabbix. Un routeur Cisco est utilisé comme exemple, mais la procédure s'applique à tout périphérique réseau compatible SNMP.

À qui s'adresse ce guide

Ce guide est conçu pour les nouveaux utilisateurs de Zabbix et les administrateurs réseau qui souhaitent activer rapidement une surveillance de base des périphériques réseau. Si vous avez besoin d'une personnalisation poussée ou d'options de configuration avancées, veuillez consulter la page agent SNMP ou la section Configuration du manuel Zabbix.

Prérequis

Avant de poursuivre ce guide, assurez-vous d'avoir:

- serveur Zabbix et interface Zabbix installés: installez-les conformément aux instructions de votre système d'exploitation (voir Installation à partir des paquets et Installation de l'interface Web).
- agent Zabbix installé si vous surveillez des métriques réseau locales.
- périphérique compatible SNMP: un commutateur réseau ou un routeur (par exemple, un routeur Cisco) avec SNMP activé.
- Fichiers MIB installés: l'installation des fichiers MIB permet à Zabbix de traduire les OID numériques en noms et descriptions lisibles par l'humain. Sans prise en charge correcte des MIB, vous ne verrez peut-être que des valeurs numériques, ce qui rendra plus difficile la configuration des éléments et le dépannage des problèmes.

Pour installer les fichiers MIB sur Ubuntu:

1. Installez le paquet de téléchargement des MIB:

```
sudo apt-get update
sudo apt-get install snmp-mibs-downloader
```

Si vous devez ajouter des MIB spécifiques à un fournisseur (par exemple, Cisco, Juniper), placez-les dans le répertoire MIB approprié:

- Pour les systèmes basés sur Linux, les emplacements courants incluent `/usr/share/snmp/mibs/` ou `/usr/local/share/snmp/mib`
- Pour les installations Zabbix, les fichiers MIB peuvent être stockés dans `/var/lib/zabbix/mibs/`.

Assurez-vous que la variable d'environnement MIBDIRS ou le fichier snmp.conf inclut le bon chemin.

Pour vérifier que votre système reconnaît les nouveaux MIB, utilisez:

```
snmptranslate -IR -On <MIB-NAME>::<object>
```

Pour des instructions détaillées, consultez la documentation de votre bibliothèque SNMP:

- MIB Cisco
- MIB Juniper

2. Modifiez /etc/snmp/snmp.conf et commentez la ligne qui commence par mibs : afin de permettre au système de charger tous les MIB disponibles.

3. Vérifiez en exécutant un snmpwalk (par exemple, snmpwalk -v 2c -c <your_community_string> <device_IP>) et assurez-vous que les OID s'affichent avec des noms descriptifs.

Selon votre environnement, certaines étapes de ce guide peuvent légèrement différer. Ce guide est basé sur un environnement exécutant Ubuntu et sur un périphérique réseau Cisco Catalyst 3750V2-24FS surveillé.

Il est supposé que votre périphérique réseau est déjà physiquement installé et connecté.

Configurer le périphérique réseau (exemple de routeur Cisco)

Pour la supervision via SNMP, vous devez configurer votre périphérique réseau afin d'autoriser les requêtes SNMP. L'exemple ci-dessous concerne SNMPv2 et ne tient pas compte des paramètres existants. Attention: l'application de ces commandes peut remplacer les configurations SNMP actuelles.

Pour un routeur Cisco, la configuration implique généralement les étapes décrites ci-dessous.

Exemple SNMPv2

1. Activez SNMP et définissez la chaîne de communauté.

Connectez-vous à la console de votre routeur Cisco et passez en mode de configuration :

```
configure terminal
```

Ensuite, activez SNMP en spécifiant une chaîne de communauté en lecture seule. Par exemple :

```
snmp-server community <your_community_string> RO
```

Remplacez <your_community_string> par votre chaîne de communauté sécurisée. Remarque : l'option RO (Read-Only) permet à SNMP de récupérer des données depuis l'appareil, mais empêche toute modification de la configuration.

Il est recommandé de restreindre l'accès SNMP aux seuls appareils nécessaires pour des raisons de sécurité. Pour plus d'informations sur la configuration des listes de contrôle d'accès (ACL), consultez la documentation officielle de Cisco.

2. Enregistrez la configuration.

Enregistrez vos modifications pour vous assurer que les paramètres SNMP persistent après un redémarrage :

```
write memory
```

Exemple SNMPv3

SNMPv3 offre une sécurité renforcée grâce à l'authentification et au chiffrement. Sa configuration est plus sécurisée que celle de SNMPv2 et doit être vérifiée dans la documentation spécifique à votre appareil.

1. Créez un groupe SNMP.

Configurez un groupe SNMPv3 avec la confidentialité (chiffrement) activée :

```
configure terminal
snmp-server group <your_group> v3 priv
```

2. Créez un utilisateur SNMP.

Ajoutez un utilisateur SNMPv3 avec authentification et confidentialité. Remplacez les espaces réservés par les valeurs souhaitées :

```
snmp-server user <your_user> <your_group> v3 auth md5 <auth_password> priv aes 128 <priv_password>
```

3. Enregistrez la configuration :

```
write memory
```

Pour plus de détails ou des instructions spécifiques au modèle, vous pouvez consulter des tutoriels externes sur la configuration SNMP Cisco. Ce guide fournit toutefois les étapes de base pour activer la supervision SNMP.

Configurer l'interface Zabbix

Créer un hôte dans l'interface Zabbix

1. Connectez-vous à l'interface Zabbix.

2. Ajoutez un nouvel hôte.

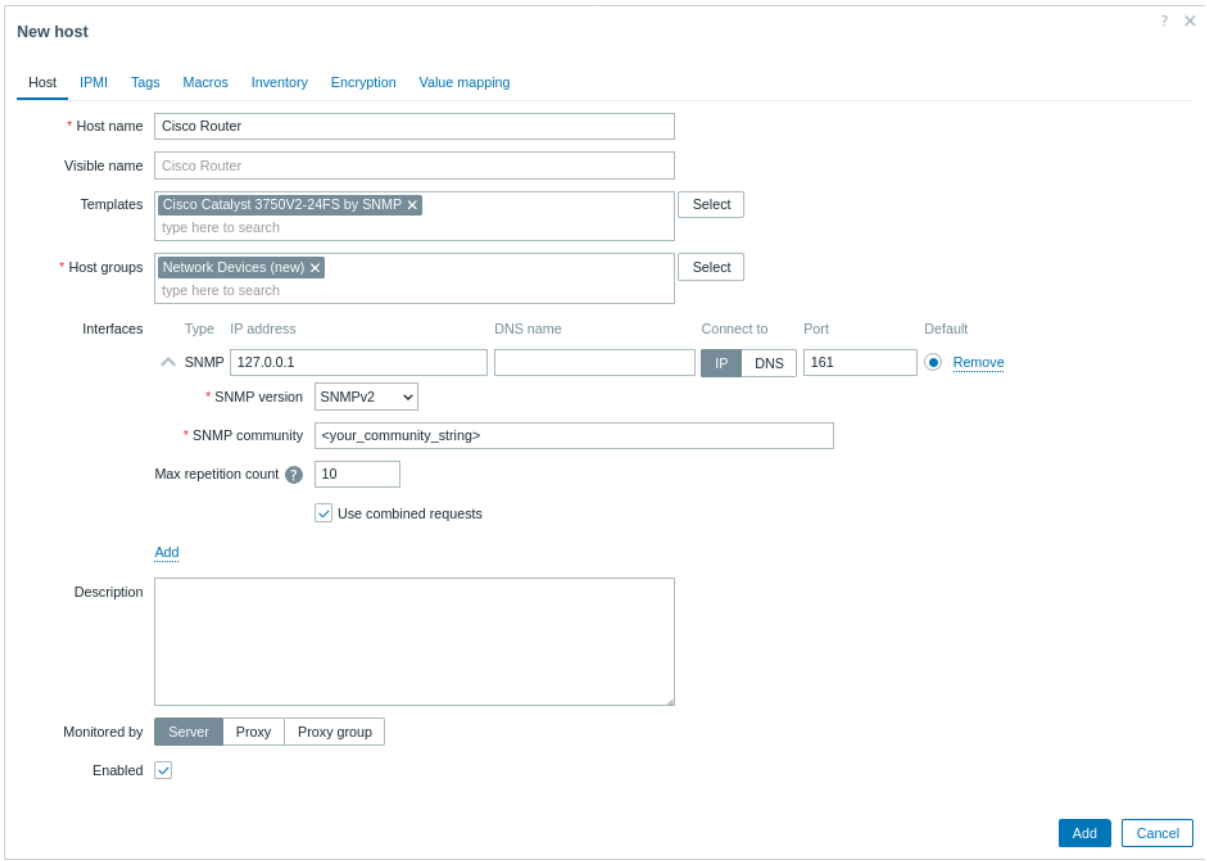
Accédez à *Collecte de données* > *Hôtes* et cliquez sur *Créer un hôte*.

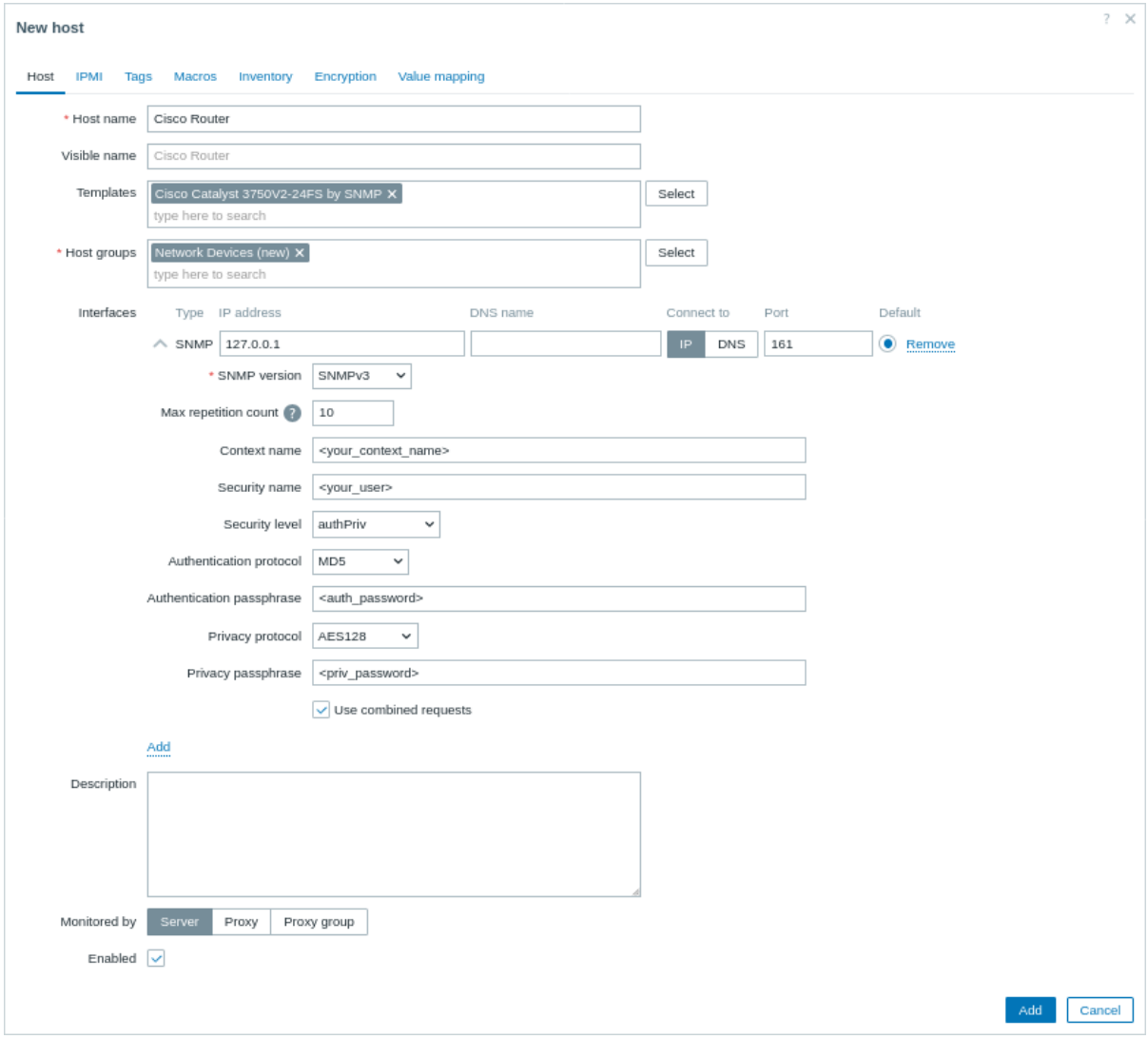
- *Nom de l'hôte* : saisissez un nom pour votre appareil (par exemple, "Cisco Router").
- Groupes d'hôtes : sélectionnez un groupe existant ou créez un nouveau groupe, par exemple "Network Devices".
- Interfaces :
  - Cliquez sur *Ajouter* sous Interfaces.
  - Choisissez *SNMP* comme type d'interface.
  - Saisissez l'adresse IP ou le nom DNS de votre routeur Cisco.
  - Définissez le port SNMP par défaut (généralement 161).
  - Utilisez le menu déroulant pour sélectionner la version SNMP appropriée (par exemple, SNMPv2).
  - Pour SNMPv1/v2, saisissez la chaîne de communauté dans le champ *Communauté SNMP*. Pour SNMPv3, des identifiants supplémentaires (*Nom de contexte*, *Nom de sécurité* et *Niveau de sécurité*, etc.) vous seront demandés.

3. Lier des modèles

Dans le champ *Modèles*, sélectionnez le modèle SNMP qui correspond le mieux à votre appareil. Zabbix fournit une gamme de modèles SNMP prédéfinis pour de nombreuses familles d'appareils. Par exemple, si vous surveillez un appareil Cisco, choisissez le modèle correspondant au système d'exploitation ou au modèle de votre appareil (comme Cisco IOS SNMP ou Cisco Catalyst 3750<device model> SNMP).

4. Cliquez sur *Ajouter* pour enregistrer l'hôte.



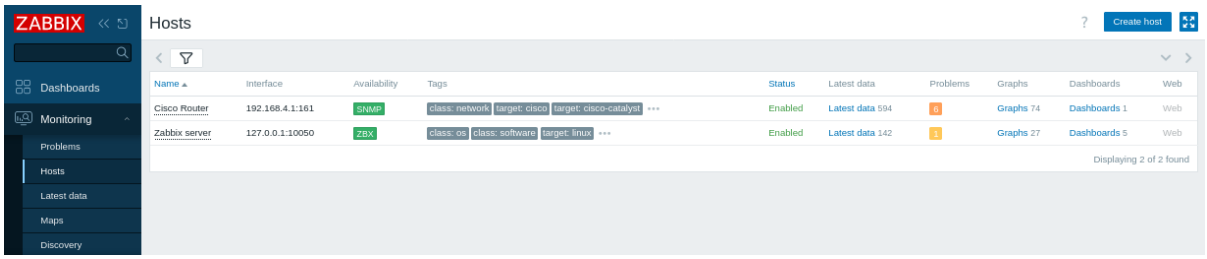


Afficher les métriques collectées

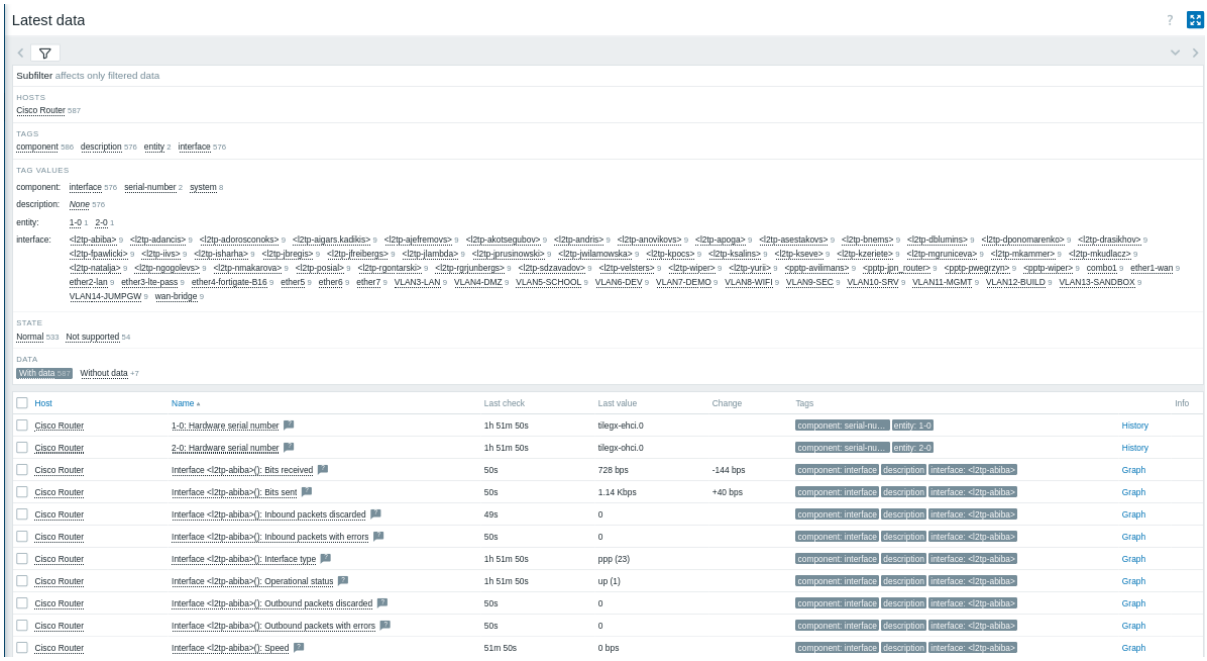
Félicitations ! Zabbix est maintenant configuré pour surveiller votre périphérique réseau.

Dernières données :

- Accédez à Surveillance > Dernières données dans l'interface Zabbix.



- Sélectionnez votre hôte "Cisco Router" (ou les hôtes découverts) pour afficher des métriques telles que la disponibilité matérielle et réseau, la perte ICMP, le ping et le temps de réponse, etc.



- Graphiques et écrans :

Pour visualiser les données de performance, cliquez sur *Graphiques* à côté des éléments SNMP afin d'afficher des métriques détaillées.

Comme prochaine étape, vous pouvez :

- Ajouter des éléments SNMP personnalisés pour surveiller des métriques supplémentaires.
- Configurer des alertes de problème pour recevoir des notifications concernant d'éventuels problèmes.

Créer des éléments SNMP

Une fois l'hôte configuré, vous pouvez créer des éléments pour surveiller des métriques spécifiques. Remarque: cette étape est facultative si vous utilisez un modèle, car les modèles contiennent déjà des ensembles d'éléments par défaut.

1. Identifiez l'OID SNMP:

Utilisez la commande `snmpwalk` pour lister les OID disponibles sur votre périphérique. Par exemple:

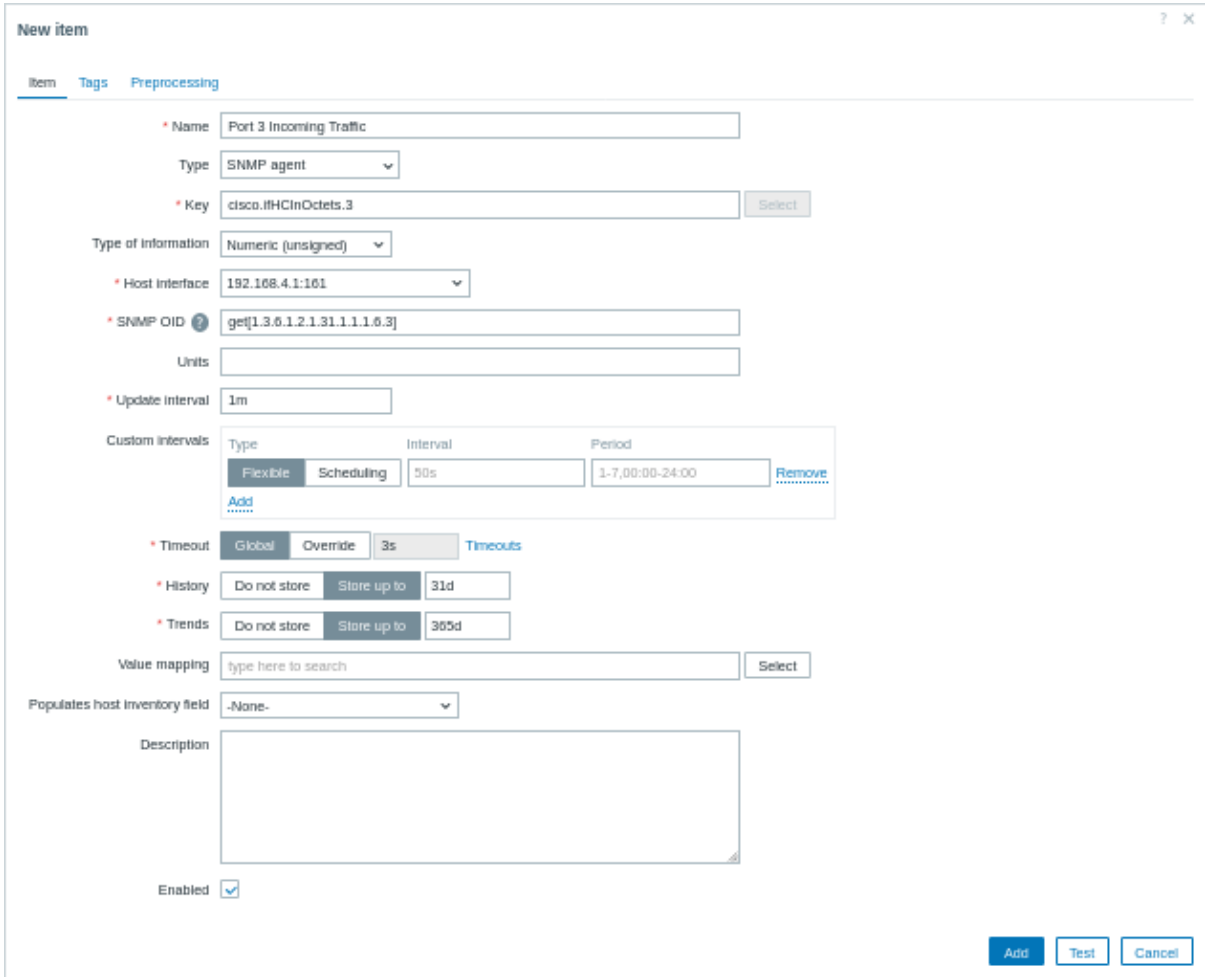
```
snmpwalk -v 2c -c <your_community_string> <device_IP> .
```

Trouvez l'OID correspondant à la métrique que vous souhaitez surveiller (par exemple, IF-MIB::ifHCInOctets.3 pour le trafic entrant sur le port 3). Pour obtenir l'OID numérique, vous pouvez utiliser:

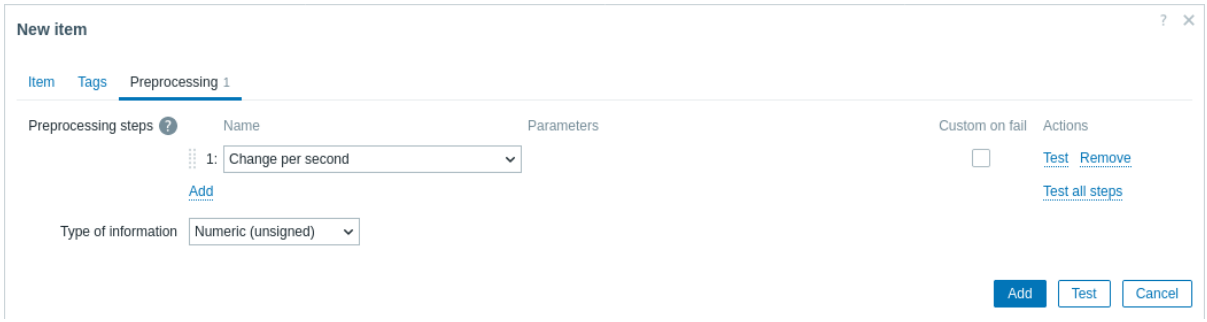
```
snmpget -v 2c -c <your_community_string> -On <device_IP> IF-MIB::ifHCInOctets.3
```

2. Créez un élément SNMP:

- Accédez à *Collecte de données* > *Hôtes* et cliquez sur l'onglet *Éléments* pour votre hôte SNMP, puis cliquez sur *Créer un élément*.
- *Nom*: saisissez un nom descriptif (par exemple, "Trafic entrant du port 3").
- *Type*: sélectionnez *agent SNMP*.
- *Clé*: fournissez une clé explicite (par exemple, `cisco.ifHCInOctets.3`).
- *Interface de l'hôte*: assurez-vous que l'interface SNMP est sélectionnée.
- *OID SNMP*: saisissez l'OID en utilisant l'un des formats pris en charge, par exemple:
  - `get[1.3.6.1.2.1.31.1.1.1.6.3]` pour une valeur unique;
  - `walk[1.3.6.1.2.1.31.1.1.1.6.3]` pour récupérer de manière asynchrone un sous-arbre de valeurs.



- *Prétraitement* (si nécessaire): si l'élément renvoie un compteur cumulatif (comme le trafic d'interface), accédez à l'onglet *Prétraitement* et ajoutez une étape de prétraitement telle que "Variation par seconde" pour calculer le débit.



Pour récupérer plusieurs valeurs dans une seule transaction SNMP, vous pouvez spécifier plusieurs OID à l'aide de la syntaxe `walk[OID1,OID2,...]`.

Traduction des OID entre noms numériques et noms MIB

Lorsque vous travaillez avec SNMP, vous pouvez avoir besoin de convertir des OID numériques en leurs noms MIB correspondants, et inversement. Cette traduction facilite l'identification et le dépannage des métriques.

- Traduire un nom MIB en OID numérique: utilisez la commande `snmptranslate` avec l'option `-On`. Par exemple, pour traduire le nom MIB `IF-MIB::ifHCInOctets.3` en son OID numérique, exécutez:

```
snmptranslate -On IF-MIB::ifHCInOctets.3
```

Cette commande peut renvoyer:

```
.1.3.6.1.2.1.31.1.1.1.6.3
```

- Traduire un OID numérique en son nom MIB: utilisez la commande `snmptranslate` avec l'option `-IR` (ou `-m ALL`) pour inverser la traduction. Par exemple, pour traduire l'OID numérique `.1.3.6.1.2.1.31.1.1.1.6.3` en son nom MIB, exécutez:

```
snmptranslate -IR -On .1.3.6.1.2.1.31.1.1.1.6.3
```

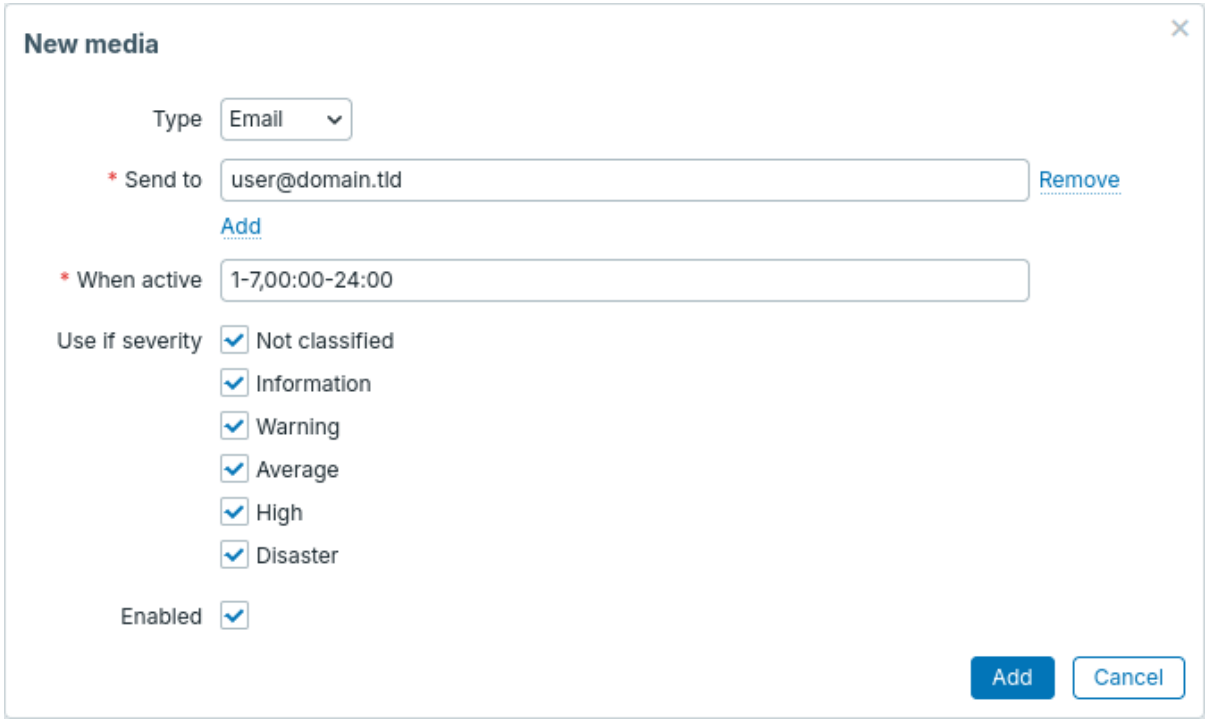
Cette commande peut renvoyer:

```
IF-MIB::ifHCInOctets.3
```

Configurer les alertes de problème

Ce guide fournit les étapes de configuration de base pour l'envoi d'alertes par e-mail.

1. Accédez à *Paramètres utilisateur* > *Profil*, ouvrez l'onglet *Média* et ajoutez votre adresse e-mail.



2. Suivez le guide pour Recevoir une notification de problème.

La prochaine fois que Zabbix détectera un problème, vous devriez recevoir une alerte par e-mail.

Testez votre configuration

Pour vous assurer que Zabbix détecte correctement les problèmes de performance réseau, simulez un problème réel en augmentant le seuil du temps de réponse du ping ICMP.

1. Ouvrez la configuration de votre hôte "Cisco Router" dans Zabbix.

2. Accédez à l'onglet *Macros* et sélectionnez *Inherited and host macros*.

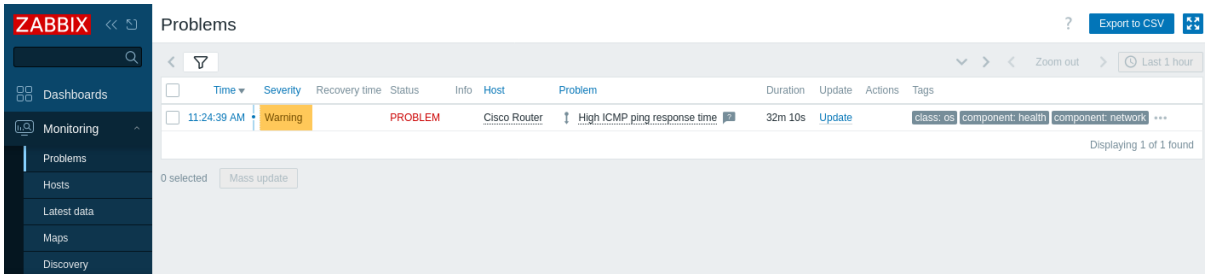
3. Repérez la macro `{$ICMP_RESPONSE_TIME_WARN}` (ou une macro de seuil de temps de réponse similaire).

4. Définissez une valeur très faible (par exemple, 0.001) afin de déclencher une alerte lorsque le temps de réponse du ping dépasse cette valeur.

5. Cliquez sur *Update* pour appliquer les modifications.

6. Attendez quelques instants que Zabbix détecte le problème simulé.

7. Accédez à *Monitoring* > *Problems* pour vérifier qu'une alerte apparaît (par exemple, "High ICMP ping response time").



Si les alertes sont configurées, vous devriez également recevoir une alerte de problème.

8. Rétablissez la valeur de la macro à son paramètre d'origine, puis cliquez sur *Update* pour enregistrer les modifications.

9. Vérifiez que le problème est résolu et qu'il disparaît de la section *Problems*.

Dépannage de la supervision SNMP

Si vous remarquez que l'icône SNMP dans l'interface Zabbix apparaît en ROUGE ou qu'aucune donnée n'est collectée, essayez les étapes suivantes :

1. Vérifiez la connectivité SNMP.

Pour SNMPv2, exécutez la commande suivante depuis votre serveur Zabbix :

```
snmpwalk -v 2c -c <community_string> <device_IP> .
```

Cette commande vérifie que l'appareil répond aux requêtes SNMP.

Pour SNMPv3, incluez les identifiants SNMPv3 appropriés :

```
snmpwalk -v3 -u <your_user> -l authPriv -a MD5 -A <auth_password> -x AES -X <priv_password> <device_IP> .
```

Cela vérifie que les identifiants SNMPv3 sont corrects et que l'appareil répond de manière sécurisée.

2. Assurez-vous que les fichiers MIB sont installés et activés comme décrit dans les prérequis. Pour cela, la commande suivante ne doit pas renvoyer d'erreur lorsque vous interrogez un périphérique réseau :

```
snmpwalk -v 2c -c <your_community_string> <device_IP> ifInOctets
```

Cela doit renvoyer des OID traduits sans erreur.

3. Confirmez que la version SNMP et les identifiants configurés dans Zabbix correspondent à ceux définis sur votre appareil. Par exemple, vérifiez les paramètres SNMP dans la configuration de l'hôte Zabbix et comparez-les à la configuration de votre appareil. Sur un appareil Cisco, vous pouvez vérifier les paramètres SNMP en exécutant :

```
show running-config | include snmp
```

Cela garantit que la chaîne de communauté (pour SNMPv2) ou les informations de l'utilisateur SNMPv3 sont correctes.

4. Vérifiez que SNMP est correctement activé sur votre périphérique réseau. Sur un routeur Cisco, connectez-vous à la console et exécutez :

```
show running-config | include snmp
```

Cette commande affiche la configuration SNMP active et aide à confirmer que SNMP est correctement configuré.

5. Assurez-vous qu'aucun pare-feu ni problème réseau ne bloque le trafic SNMP (généralement sur le port 161) entre le serveur Zabbix et l'appareil. Vous pouvez tester la connectivité à l'aide de :

nc -zv <device_IP> 161

`nc -zv` vérifie si le port 161 est ouvert et à l'écoute sur l'appareil.

De plus, si vous utilisez UFW sur Ubuntu, vérifiez l'état du pare-feu :

```
sudo ufw status
```

Ou, pour iptables :

```
sudo iptables -L -n
```

6. Consultez les fichiers journaux du serveur Zabbix pour détecter d'éventuelles erreurs liées à SNMP afin d'identifier le problème :

```
tail -f /tmp/zabbix_server.log
```

`tail -f` vous permet de surveiller les mises à jour du journal en temps réel.

Voir aussi :

- Créer un élément - apprendre à ajouter des métriques supplémentaires.
- agent SNMP - informations supplémentaires sur la supervision SNMP avec Zabbix.
- Modèles standardisés pour les périphériques réseau - informations sur les modèles SNMP disponibles.
- Découverte des OID SNMP - informations supplémentaires sur la découverte SNMP sur un commutateur.
- Configuration d'une règle de découverte réseau - informations supplémentaires sur la configuration d'une règle de découverte réseau utilisée par Zabbix pour découvrir des hôtes et des services.

Surveiller le journal des événements Windows à l'aide de vérifications actives

Introduction

Ce guide explique comment surveiller les journaux d'événements Windows avec Zabbix à l'aide de vérifications actives. Grâce aux clés d'élément spécifiques à Windows de Zabbix, vous pouvez collecter et analyser en temps réel des événements critiques (tels que des tentatives de connexion échouées, des erreurs système, etc.).

À qui ce guide s'adresse-t-il

Ce guide est conçu pour les nouveaux utilisateurs de Zabbix et les administrateurs réseau qui souhaitent surveiller les journaux d'événements Windows. Pour les options de configuration avancées, consultez la documentation sur les clés d'élément spécifiques à Windows.

Prérequis

Avant de poursuivre ce guide, vous devez télécharger et installer le serveur Zabbix et l'interface Zabbix conformément aux instructions de votre système d'exploitation. Vous devez également télécharger et installer l'agent Zabbix sur la machine Windows que vous souhaitez surveiller.

Configurer l'agent Zabbix pour surveiller les journaux d'événements Windows

1. Ouvrez le fichier de configuration de l'agent sur votre hôte Windows et vérifiez que le paramètre `ServerActive` est défini sur l'adresse IP de votre serveur Zabbix et que le paramètre `Hostname` correspond au nom d'hôte qui sera défini dans l'interface Zabbix. Cela permet à l'agent de demander des vérifications actives pour son hôte auprès du serveur Zabbix spécifié. Par exemple :

```
ServerActive=192.0.2.1
Hostname=MyWindowsHost
```

2. Redémarrez le service de l'agent Zabbix pour appliquer les modifications :

```
net stop "Zabbix Agent" && net start "Zabbix Agent"
```

3. Vérifiez que l'hôte Windows est en cours d'exécution :

- Vérifiez que le service de l'agent Zabbix est en cours d'exécution sur l'hôte Windows.
- Vérifiez que l'hôte Windows peut se connecter au serveur Zabbix sur le port 10051. Pour tester la connectivité depuis l'hôte Windows, ouvrez PowerShell et exécutez la commande suivante :

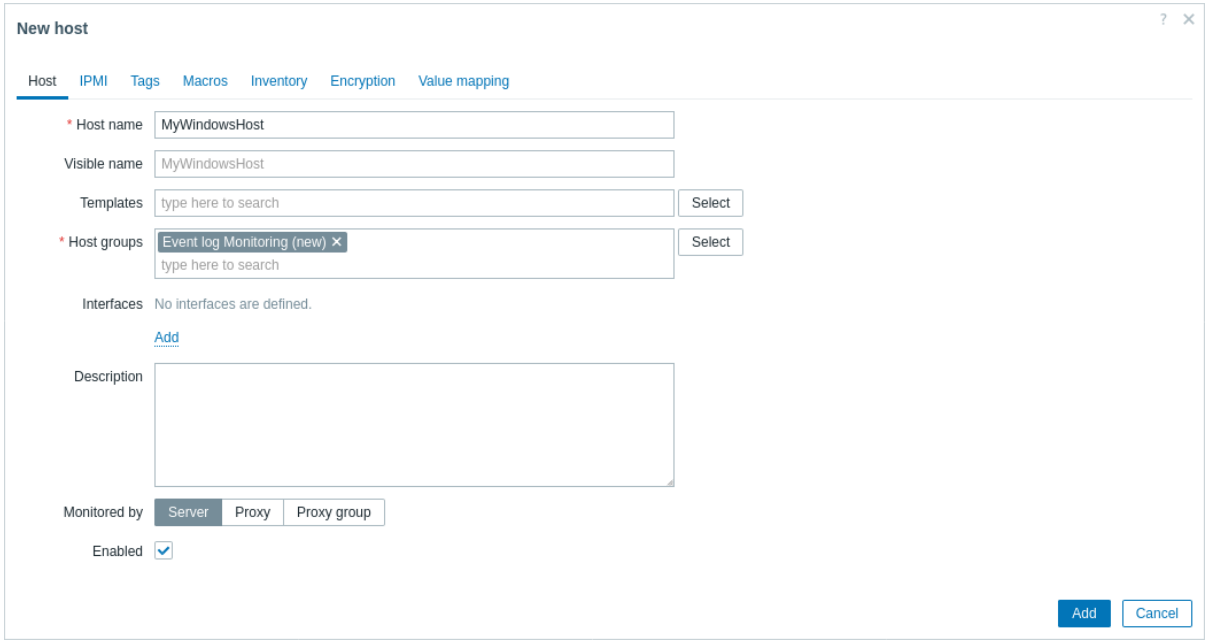
```
Test-NetConnection -ComputerName <Zabbix-server-IP> -Port 10051
```

Configurer l'interface Zabbix

1. Accédez à *Data collection* > *Hosts* et créez un hôte :

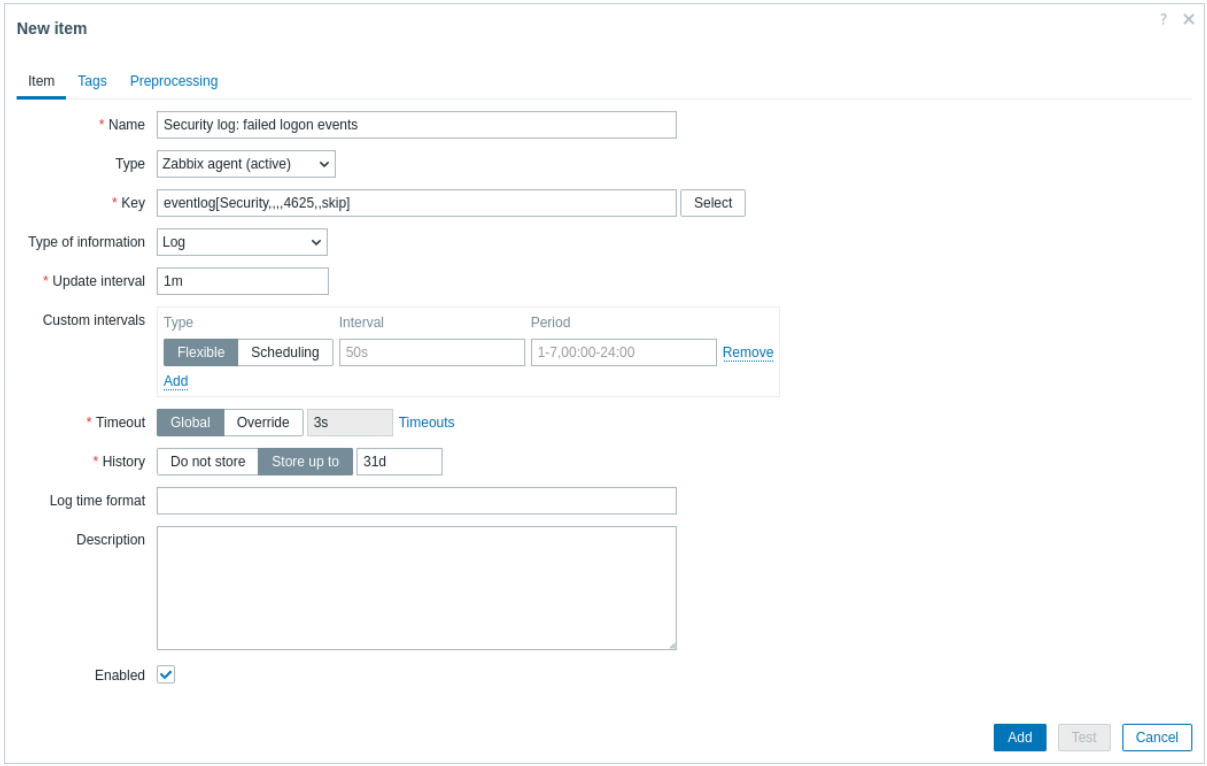
- Dans le champ *Host name*, saisissez un nom d'hôte (par exemple, "MyWindowsHost").
- Dans le champ *Host groups*, saisissez ou sélectionnez un groupe d'hôtes (par exemple, "Event log Monitoring").
- Cliquez sur *Add* pour enregistrer l'hôte configuré.

> **Note:**
> Dans le champ *Templates*, vous pouvez ajouter le modèle "Windows by Zabbix agent active" pour vous aider à diagnostiquer le problème en observant si d'autres éléments actifs sur le même hôte sont mis à jour.



2. Créez un nouvel élément avec les paramètres suivants :

- Dans le champ *Name*, saisissez un nom d'élément descriptif (par exemple, "Security log: failed logon events").
- Dans la liste déroulante *Type*, sélectionnez "Zabbix agent (active)" (requis pour la surveillance du journal d'événements).
- Dans le champ *Key*, utilisez la clé d'élément eventlog. Par exemple, pour surveiller les tentatives de connexion échouées (Event ID: 4625) dans le journal Security et ignorer les entrées plus anciennes que la dernière vérification de l'élément (en utilisant le paramètre `skip`), saisissez la clé d'élément suivante : `eventlog[Security,,,,4625,,skip]`
- Dans la liste déroulante *Type of information*, sélectionnez "Log".



3. Cliquez sur *Add* pour enregistrer l'élément.

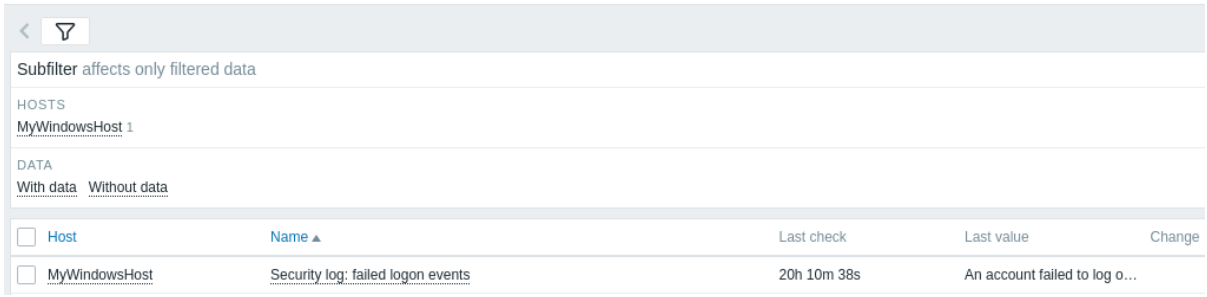
Tester et afficher les métriques collectées

Félicitations ! Zabbix est maintenant configuré pour collecter vos journaux d'événements Windows. Pour vérifier que les journaux d'événements sont bien collectés, vous pouvez tester l'élément "Security log: failed logon events" en vous déconnectant de votre compte Windows, puis en tentant de vous connecter avec des identifiants incorrects.

Ensuite, affichez les journaux collectés dans l'interface Zabbix :

1. Accédez à *Monitoring* > *Latest data* dans l'interface Zabbix.

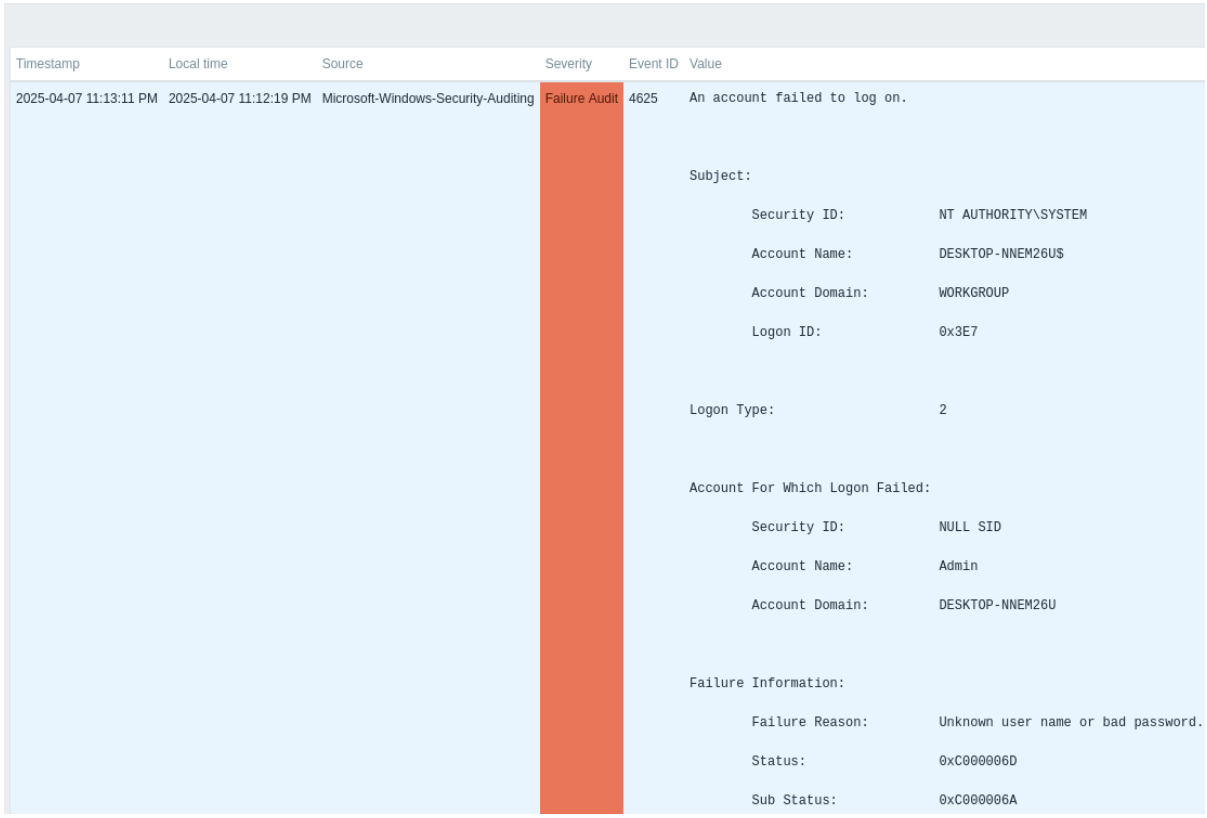
Latest data



2. Filtrez par votre hôte ”MyWindowsHost” dans le champ *Name*.

3. Cliquez sur *History* pour afficher les valeurs de journal enregistrées.

MyWindowsHost: Security log: failed logon events



4. Si les valeurs du journal sont absentes, passez à la section Dépannage du guide.

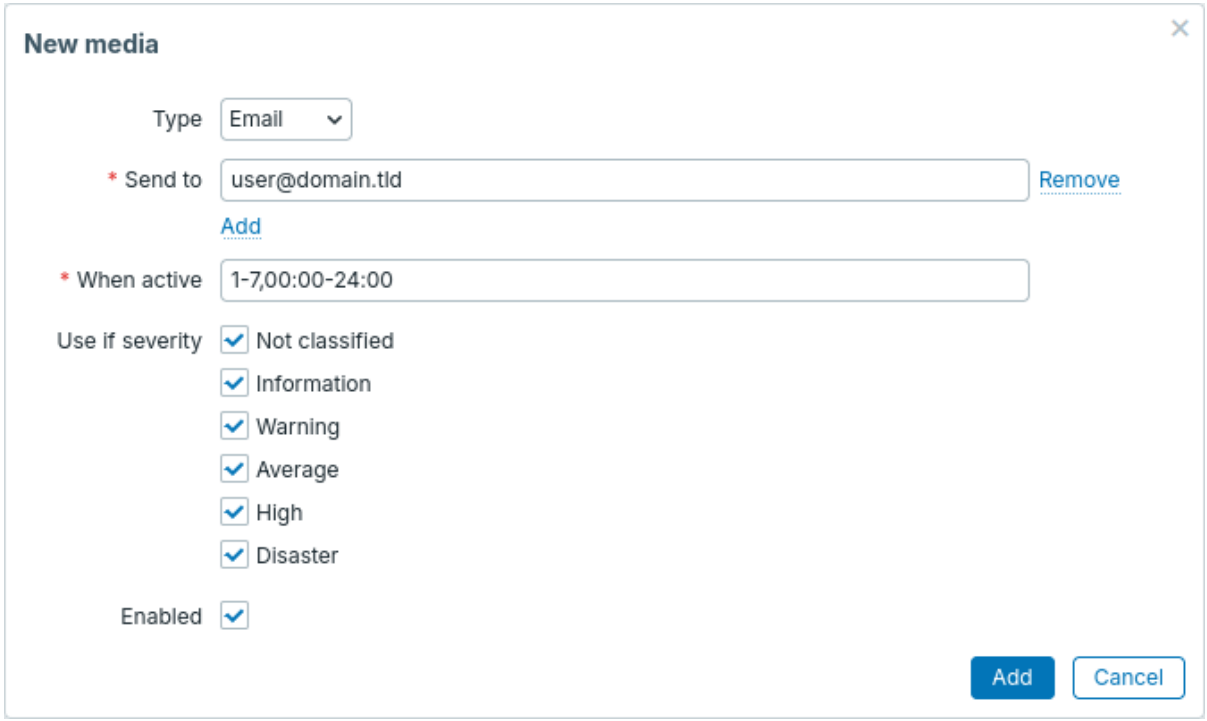
Configuration des alertes de problème

Ce guide fournit les étapes de configuration de base pour l’envoi d’alertes par e-mail.

1. Accédez à Collecte de données > Hôtes pour définir un déclencheur qui se déclenche lorsque votre élément journal d’événements enregistre le motif qui vous intéresse. Par exemple, pour détecter les tentatives de connexion échouées dans le journal Security, utilisez la fonction find() :

find(/MyWindowsHost/eventlog[Security,,,4625,skip],10m,”like”,”Logon failed”)

2. Accédez à *Paramètres utilisateur > Profil*, ouvrez l’onglet *Média* et ajoutez votre e-mail.



3. Suivez le guide pour Recevoir une notification de problème.

La prochaine fois que Zabbix détectera un problème, vous devriez recevoir une alerte par e-mail.

Dépannage

Si vous rencontrez des problèmes lors de la collecte ou de l'affichage des journaux d'événements Windows, utilisez les conseils ci-dessous pour identifier et résoudre les problèmes courants :

1. Sur le serveur Zabbix (Linux), listez vos règles iptables avec la commande suivante :

```
sudo iptables -L -n
```

et vérifiez qu'il existe une règle ACCEPT pour le port TCP 10051.

2. Assurez-vous que votre clé `eventlog[...]` utilise exactement le nom du journal (sensible à la casse), l'ID d'événement, le mode (par exemple, skip) et les autres paramètres exactement comme indiqué dans les clés d'éléments spécifiques à Windows.

Voir aussi :

- Créer un élément - découvrez comment ajouter des métriques supplémentaires.
- Agent Zabbix sur Microsoft Windows - instructions d'installation détaillées.
- Surveiller Windows avec l'agent Zabbix - un guide complet pour configurer la supervision de base des machines Windows à l'aide de l'agent Zabbix.
- Clés d'élément spécifiques à Windows - informations détaillées sur les clés d'élément spécifiques à Windows prises en charge par les agents Zabbix, y compris celles destinées à la supervision des journaux d'événements.
- Supervision des fichiers journaux - instructions pour configurer Zabbix afin d'assurer la supervision centralisée et l'analyse des fichiers journaux, applicable aux journaux d'événements Windows.

Installation : guides supplémentaires

Si vous recherchez la méthode la plus simple pour installer Zabbix, consultez Installation et premières étapes.

Les guides ci-dessous présentent des méthodes **supplémentaires** ou spécifiques à certains systèmes pour installer Zabbix ou ses composants :

- Installation à partir de conteneurs
- Installation à partir des sources
- Remarques sur l'installation à partir de paquets
- Installation de l'agent Zabbix à partir d'un MSI Windows
- Compilation de l'agent Zabbix 2 sous Windows
- Compilation de l'agent Zabbix sous Windows
- Installation de l'agent macOS à partir d'un PKG

- Agent Zabbix sur IBM AIX
- Compilation de l'agent Zabbix sous macOS
- Appliance Zabbix
- Guide de déploiement du serveur Zabbix sur AWS
- Guide de déploiement du proxy Zabbix sur AWS
- Guide de déploiement du serveur Zabbix sur Azure
- Guide de déploiement du proxy Zabbix sur Azure
- Guide de déploiement sur Google Cloud

Installation à partir de conteneurs

Vue d'ensemble

Cette page décrit comment déployer Zabbix à l'aide de Docker ou de Docker Compose.

Docker Compose est le moyen le plus rapide de déployer Zabbix. Il lit un fichier de configuration et démarre automatiquement tous les conteneurs qui composent une installation Zabbix complète, dans le bon ordre.

Docker (déploiement manuel) permet d'obtenir le même résultat en déployant chaque composant étape par étape.

Prérequis

Avant de commencer, assurez-vous que Docker (1.12.0 ou version ultérieure) est installé sur votre système. Si ce n'est pas le cas, suivez le guide d'installation de Docker. Pour un déploiement avec Docker Compose, Docker Compose (2.24.0 ou version ultérieure) est également requis.

Certains composants Zabbix nécessitent que des ports spécifiques soient ouverts sur l'hôte exécutant Docker (par exemple, 10051/TCP pour le serveur Zabbix et 162/UDP pour les traps SNMP). Consultez la page Prérequis pour obtenir la liste complète des ports utilisés par les composants Zabbix. Pour le serveur Zabbix et l'agent, le port par défaut peut être modifié en définissant la variable d'environnement `ZBX_LISTENPORT` sur le conteneur correspondant.

Images Docker disponibles

Zabbix fournit une image Docker pour chaque composant Zabbix, toutes publiées sur Docker Hub. Chaque image est utilisée pour créer un conteneur exécutant le composant correspondant.

Composant Zabbix	Image Docker
Agent	zabbix/zabbix-agent
Agent 2	zabbix/zabbix-agent2
Serveur (MySQL)	zabbix/zabbix-server-mysql
Serveur (PostgreSQL)	zabbix/zabbix-server-pgsql
Interface Web (Apache + MySQL)	zabbix/zabbix-web-apache-mysql
Interface Web (Apache + PostgreSQL)	zabbix/zabbix-web-apache-pgsql
Interface Web (Nginx + MySQL)	zabbix/zabbix-web-nginx-mysql
Interface Web (Nginx + PostgreSQL)	zabbix/zabbix-web-nginx-pgsql
Proxy (SQLite3)	zabbix/zabbix-proxy-sqlite3
Proxy (MySQL)	zabbix/zabbix-proxy-mysql
Java gateway	zabbix/zabbix-java-gateway
Service Web	zabbix/zabbix-web-service
Traps SNMP	zabbix/zabbix-snmptraps
Serveur MCP	zabbix/zabbix-mcp-server

Note:
Pour utiliser les traps SNMP, le conteneur de traps SNMP doit partager un volume avec le conteneur du serveur ou du proxy Zabbix (voir comment utiliser cette image sur Docker Hub et l'exemple ci-dessous).

Toutes les images des composants Zabbix sur Docker Hub sont basées sur les dernières versions majeures des systèmes d'exploitation pris en charge. Ces images sont automatiquement reconstruites lorsque les images du système d'exploitation sous-jacent sont mises à jour.

Balises d'image

Chaque image prend en charge des balises pour sélectionner le système d'exploitation de base et la version de Zabbix :

```
zabbix/<image>:<os>-<version>
```

Valeurs <os> prises en charge :

- `alpine` - Alpine Linux
- `ubuntu` - Ubuntu
- `centos` - CentOS Stream
- `ol` - Oracle Linux
- `ltsc2022` - Windows 11 LTSC 2022 (agent Zabbix uniquement)

Valeurs <version> prises en charge :

- `latest` - Dernière version stable de Zabbix sur Alpine Linux
- `<os>-latest` - Dernière version stable de Zabbix sur le système d'exploitation sélectionné
- `<os>-trunk` - Dernière compilation de développement (nightly) sur le système d'exploitation sélectionné
- `<os>-X.X-latest` - Dernière version mineure de Zabbix d'une version majeure spécifique de Zabbix sur le système d'exploitation sélectionné
- `<os>-X.X.*` - Version mineure spécifique de Zabbix sur le système d'exploitation sélectionné

Exemples :

```
#### Dernière version stable du proxy Zabbix (SQLite3) sur Alpine Linux :
docker pull zabbix/zabbix-proxy-sqlite3:latest

#### Dernière version stable du proxy Zabbix (SQLite3) sur Ubuntu :
docker pull zabbix/zabbix-proxy-sqlite3:ubuntu-latest

#### Dernière compilation de développement (nightly) du serveur Zabbix (MySQL) sur Ubuntu :
docker pull zabbix/zabbix-server-mysql:ubuntu-trunk

#### Dernière version mineure 8.0 du serveur Zabbix (MySQL) sur Alpine Linux :
docker pull zabbix/zabbix-server-mysql:alpine-8.0-latest

#### Version 8.0.1 du serveur Zabbix (MySQL) sur Alpine Linux :
docker pull zabbix/zabbix-server-mysql:alpine-8.0.1
```

Docker Compose

Docker Compose est le moyen le plus rapide de déployer Zabbix. Il lit un fichier de configuration (fichier Compose) et gère l'ensemble de la configuration : téléchargement des images Docker, création d'un réseau interne entre les conteneurs, configuration du stockage, initialisation de la base de données et démarrage de tous les composants dans le bon ordre.

Le dépôt des Dockerfiles officiels de Zabbix fournit des fichiers Docker Compose prêts à l'emploi ainsi qu'un système de configuration basé sur `.env`, prenant en charge différents systèmes d'exploitation, moteurs de base de données et configurations des composants Zabbix.

1. Clonez le dépôt, accédez à son répertoire et passez à la version 8.0 :

```
git clone https://github.com/zabbix/zabbix-docker.git
cd zabbix-docker
git checkout 8.0
```

2. (Facultatif) Personnalisez le déploiement à l'aide de variables d'environnement. Cette étape peut être ignorée pour une configuration par défaut.

3. Déployez la configuration par défaut : serveur Zabbix, interface web (fonctionnant sur NGINX) et moteur de base de données (MySQL ou PostgreSQL), chacun s'exécutant dans son propre conteneur sur Alpine Linux.

```
#### Avec MySQL comme base de données :
docker compose -f ./compose.yaml up -d

#### Avec PostgreSQL comme base de données :
docker compose -f ./compose_pgsql.yaml up -d
```

Une fois les conteneurs démarrés et opérationnels (généralement sous 1 à 3 minutes), le serveur Zabbix démarre et l'interface web devient disponible pour la connexion.

Note:
Utilisez `docker compose ps` pour vérifier l'état des conteneurs. Tous les conteneurs (à l'exception du conteneur `zabbix-docker-server-db-init-1`) doivent avoir l'état `Up`. Si l'un d'eux affiche l'état `Exited`, consultez ses journaux pour rechercher d'éventuelles erreurs à l'aide de `docker logs -f <container-name>`.

Variables d'environnement

Le comportement de Docker Compose et la configuration des composants Zabbix peuvent être personnalisés à l'aide de variables d'environnement.

Les **variables au niveau de Compose** (définies dans le fichier `.env`) contrôlent les images Docker, les ports et les plages d'adresses IP réseau utilisés. Vous pouvez utiliser ces variables directement avant la commande `docker compose` ou modifier le fichier `.env`.

Par exemple, la commande suivante déploie une configuration Zabbix complète composée de plusieurs conteneurs à l'aide d'une image basée sur Ubuntu (au lieu de l'image Alpine Linux utilisée par défaut) et expose l'interface web via NGINX sur des ports HTTP (8282) et HTTPS (8443) personnalisés :

```
OS=ubuntu \
ZABBIX_WEB_NGINX_HTTP_PORT=8282 \
ZABBIX_WEB_NGINX_HTTPS_PORT=8443 \
docker compose -f ./compose.yaml up -d
```

Les **variables au niveau des composants** (définies dans les fichiers `env_vars/.env_<component>`) contrôlent la configuration des composants Zabbix. Modifiez le fichier `.env_<component>` correspondant avant d'exécuter la commande `docker compose`.

Par exemple, vous pouvez ajuster le nombre de processus d'interrogation passifs du serveur Zabbix et les paramètres du cache, ainsi que configurer le fuseau horaire de l'interface web, en modifiant les variables suivantes :

```
#### env_vars/.env_srv
ZBX_STARTPOLLERS=20
ZBX_CACHESIZE=64M
ZBX_HISTORYCACHESIZE=32M

#### env_vars/.env_web
PHP_TZ=Europe/Paris
```

Note:
Les variables d'environnement au niveau des composants correspondent aux paramètres de configuration des composants Zabbix, mais utilisent une convention de nommage différente (par exemple, `ZBX_STARTPOLLERS` correspond à `StartPollers`). Certaines variables sont spécifiques à Docker et certains paramètres de configuration ne peuvent pas être modifiés (par exemple, `PIDFile` et `LogType`). Lorsque vous utilisez des variables d'environnement, consultez la section *Environment variables* de la description de l'image Docker de chaque composant sur Docker Hub.

Volumes

Docker Compose stocke les données persistantes dans le répertoire `zbx_env/` créé à côté du fichier Compose. Ce répertoire conserve les données des composants lors des redémarrages et des mises à jour des conteneurs.

Le contenu de `zbx_env/` est prédéfini pour l'image de chaque composant. Par exemple :

- `zbx_env/etc/` - utilisé par le conteneur de l'interface web Zabbix
- `zbx_env/usr/` et `zbx_env/var/` - utilisés par le conteneur du serveur Zabbix

Pour plus d'informations sur chaque volume, consultez la section *Allowed volumes* dans la description de chaque image Docker de composant sur Docker Hub.

Makefile

Le dépôt des fichiers Docker de Zabbix fournit également un Makefile qui permet d'exécuter rapidement les tâches courantes de Docker Compose. Au lieu d'utiliser la commande complète `docker compose`, vous pouvez utiliser des commandes `make` plus courtes (exécutez `make help` pour afficher les options disponibles) :

```
#### Déploiement par défaut (serveur Zabbix, interface, MySQL, le tout sur Alpine Linux)
make up

#### Déploiement personnalisé (serveur Zabbix, interface avec des ports NGINX personnalisés, PostgreSQL,
```

```
make up \
  OS=ubuntu \
  DB=pgsql \
  ZABBIX_WEB_NGINX_HTTP_PORT=8282 \
  ZABBIX_WEB_NGINX_HTTPS_PORT=8443
```

Attention:
Lors de l'arrêt ou de la suppression des conteneurs, spécifiez toujours le même type de base de données que celui utilisé pour le déploiement (par exemple, `make down DB=pgsql`).

Par défaut, `make up` démarre uniquement un ensemble minimal de services (serveur Zabbix, interface et base de données). Cela permet de conserver une configuration légère et d'éviter de démarrer des composants qui pourraient ne pas être nécessaires. Pour inclure des composants supplémentaires, vous pouvez utiliser les profils Compose suivants :

```
#### Déploiement par défaut + agent Zabbix, Java gateway, service web et traps SNMP :
make up COMPOSE_PROFILES=full

#### COMPOSE_PROFILES=full + proxy Zabbix (MySQL et SQLite3) :
make up COMPOSE_PROFILES=all
```

Docker (déploiement manuel)

Utilisez le déploiement manuel lorsque vous souhaitez déployer Zabbix étape par étape, exécuter des composants individuels, les intégrer à un environnement existant ou utiliser un runtime de conteneurs alternatif tel que Podman.

Par exemple, pour déployer un proxy Zabbix, exécutez la commande suivante :

```
docker run --name zabbix-proxy-sqlite3 \
  -e ZBX_SERVER_HOST=192.0.2.1 \
  -e ZBX_PROXYMODE=0 \
  -e ZBX_HOSTNAME=zabbix-proxy-sqlite3 \
  -v zabbix-proxy-data:/var/lib/zabbix/db_data \
  --init \
  -d \
  zabbix/zabbix-proxy-sqlite3:alpine-8.0-latest
```

Cette commande :

- Télécharge l'image Docker `zabbix/zabbix-proxy-sqlite3:alpine-8.0-latest`.
- Crée et démarre un conteneur `zabbix-proxy-sqlite3` basé sur l'image Docker (avec les options `--init` et `-d`).
- Configure le paramètre `Server` du proxy Zabbix via la variable d'environnement `ZBX_SERVER_HOST`. Ce paramètre spécifie l'adresse IP du serveur Zabbix depuis lequel le proxy récupérera les données de configuration et auquel il enverra les données collectées. Notez que les autres variables requises pour le fonctionnement d'un proxy actif utilisent les valeurs par défaut et peuvent être omises.

Note:
Les variables d'environnement correspondent aux paramètres de configuration des composants Zabbix selon une convention de nommage différente (par exemple, `ZBX_SERVER_HOST` correspond à `Server`). Certaines variables sont spécifiques à Docker, et certains paramètres de configuration ne peuvent pas être modifiés (par exemple, `PIDFile` et `LogType`). Lorsque vous utilisez des variables d'environnement, veuillez vous référer à la section *Environment variables* dans la description de l'image Docker de chaque composant sur Docker Hub.

- Connecte un emplacement de stockage géré par Docker (par exemple, `/var/lib/docker/volumes/zabbix-proxy-data/`) au répertoire `/var/lib/zabbix` du conteneur afin que les données du proxy Zabbix soient conservées de manière permanente, même si le conteneur est supprimé.

Note:
Pour plus d'informations sur chaque volume, consultez la section *Allowed volumes* dans la description de l'image Docker de chaque composant sur Docker Hub.

Après avoir déployé le conteneur du proxy Zabbix, vous pouvez procéder à l'ajout du proxy dans votre interface web Zabbix et configurer les hôtes à surveiller par ce proxy.

Les exemples ci-dessous couvrent trois scénarios de déploiement supplémentaires :

- serveur Zabbix (MySQL) avec Java gateway
- serveur Zabbix (PostgreSQL) avec des traps SNMP
- serveur Zabbix (MySQL) avec Java gateway sur RHEL 8-10

Pour plus d'exemples, veuillez consulter la description de l'image Docker de chaque composant sur Docker Hub.

Déployer le serveur Zabbix (MySQL) avec Java gateway

Cet exemple montre comment déployer un serveur Zabbix avec un backend MySQL, une Java gateway pour la supervision JMX et une interface basée sur NGINX.

1. Créez un réseau Docker dédié afin que tous les conteneurs des composants Zabbix puissent communiquer entre eux à l'aide du nom du conteneur :

```
docker network create --subnet 172.20.0.0/16 --ip-range 172.20.240.0/20 zabbix-net
```

2. Exécutez un conteneur de serveur MySQL vide (remplacez `zabbix_pwd` et `root_pwd` par des mots de passe robustes et utilisez ces valeurs lors des étapes suivantes) :

```
docker run --name mysql-server -t \
  -e MYSQL_DATABASE="zabbix" \
  -e MYSQL_USER="zabbix" \
  -e MYSQL_PASSWORD="zabbix_pwd" \
  -e MYSQL_ROOT_PASSWORD="root_pwd" \
  --network=zabbix-net \
  --restart unless-stopped \
  -d mysql:8.4-oracle \
  --character-set-server=utf8mb4 --collation-server=utf8mb4_bin
```

3. Suivez les journaux MySQL et attendez que MySQL soit prêt à accepter les connexions (utilisez `Ctrl+C` pour quitter les journaux) :

```
docker logs -f mysql-server

#### ... [Server] /usr/sbin/mysqld: ready for connections. ...
```

4. Exécutez un conteneur Java gateway Zabbix :

```
docker run --name zabbix-java-gateway -t \
  --network=zabbix-net \
  --restart unless-stopped \
  -d zabbix/zabbix-java-gateway:alpine-8.0-latest
```

5. Activez `log_bin_trust_function_creators` dans le conteneur du serveur MySQL. Cette option assouplit temporairement les restrictions de sécurité appliquées aux fonctions stockées ; elle sera désactivée lors des étapes suivantes. Elle permet à l'utilisateur de la base de données Zabbix de créer des fonctions stockées sans le privilège `SUPER`, requis par MySQL lorsque la journalisation binaire est activée (par défaut depuis MySQL 8.0). Exécutez la commande suivante :

```
docker exec -it mysql-server mysql -u root -p'root_pwd' \
  -e "SET GLOBAL log_bin_trust_function_creators = 1;"
```

6. Exécutez un conteneur de serveur Zabbix et liez-le aux conteneurs Java Gateway et du serveur MySQL :

```
docker run --name zabbix-server-mysql -t \
  -e DB_SERVER_HOST="mysql-server" \
  -e MYSQL_DATABASE="zabbix" \
  -e MYSQL_USER="zabbix" \
  -e MYSQL_PASSWORD="zabbix_pwd" \
  -e ZBX_JAVAGATEWAY="zabbix-java-gateway" \
  --network=zabbix-net \
  -p 10051:10051 \
  --restart unless-stopped \
  --init \
  -d zabbix/zabbix-server-mysql:alpine-8.0-latest
```

7. Suivez les journaux du serveur Zabbix et attendez que le serveur Zabbix ait terminé l'initialisation du schéma de la base de données (utilisez `Ctrl+C` pour quitter les journaux) :

```
docker logs -f zabbix-server-mysql
```

```
#### ... [info]: ** Creating 'zabbix' schema in MySQL
#### ... [info]: ** Database schema successfully created!
```

8. Désactivez `log_bin_trust_function_creators` dans le conteneur du serveur MySQL. Cette opération rétablit la restriction de sécurité qui empêche les utilisateurs ne disposant pas du privilège `SUPER` de créer des fonctions stockées. Exécutez la commande suivante :

```
docker exec -it mysql-server mysql -u root -p'root_pwd' \
  -e "SET GLOBAL log_bin_trust_function_creators = 0;"
```

9. Exécutez un conteneur d'interface web Zabbix et liez-le aux conteneurs du serveur Zabbix et du serveur MySQL :

```
docker run --name zabbix-web-nginx-mysql -t \
  -e ZBX_SERVER_HOST="zabbix-server-mysql" \
  -e DB_SERVER_HOST="mysql-server" \
  -e MYSQL_DATABASE="zabbix" \
  -e MYSQL_USER="zabbix" \
  -e MYSQL_PASSWORD="zabbix_pwd" \
  -e PHP_TZ="Europe/Riga" \
  --network=zabbix-net \
  -p 80:8080 \
  --restart unless-stopped \
  -d zabbix/zabbix-web-nginx-mysql:alpine-8.0-latest
```

Une fois les conteneurs démarrés et opérationnels, le serveur Zabbix démarre et l'interface web devient disponible pour la connexion.

Note:
Utilisez `docker ps` pour vérifier l'état des conteneurs. Tous les conteneurs doivent avoir l'état `Up`. Si l'un d'eux affiche l'état `Exited`, consultez ses journaux pour rechercher les erreurs avec `docker logs -f <container-name>`.

Déployer le serveur Zabbix (PostgreSQL) avec des traps SNMP

Cet exemple montre comment déployer un serveur Zabbix avec un backend PostgreSQL, des traps SNMP et une interface basée sur NGINX.

1. Créez un réseau Docker dédié afin que tous les conteneurs des composants Zabbix puissent communiquer entre eux à l'aide du nom du conteneur :

```
docker network create --subnet 172.20.0.0/16 --ip-range 172.20.240.0/20 zabbix-net
```

2. Exécutez un conteneur de serveur PostgreSQL vide (remplacez `zabbix_pwd` par un mot de passe robuste et utilisez cette valeur lors des étapes suivantes) :

```
docker run --name postgres-server -t \
  -e POSTGRES_USER="zabbix" \
  -e POSTGRES_PASSWORD="zabbix_pwd" \
  -e POSTGRES_DB="zabbix" \
  --network=zabbix-net \
  --restart unless-stopped \
  -d postgres:latest
```

3. Exécutez un conteneur de traps SNMP Zabbix :

```
docker run --name zabbix-snmptraps -t \
  -v /zbx_instance/snmptraps:/var/lib/zabbix/snmptraps:rw \
  -v /var/lib/zabbix/mibs:/usr/share/snmp/mibs:ro \
  --network=zabbix-net \
  -p 162:1162/udp \
  --restart unless-stopped \
  -d zabbix/zabbix-snmptraps:alpine-8.0-latest
```

4. Exécutez un conteneur de serveur Zabbix et liez-le aux conteneurs du serveur PostgreSQL et des traps SNMP :

```
docker run --name zabbix-server-pgsql -t \
  -e DB_SERVER_HOST="postgres-server" \
  -e POSTGRES_USER="zabbix" \
  -e POSTGRES_PASSWORD="zabbix_pwd" \
```

```
  -e POSTGRES_DB="zabbix" \
  -e ZBX_ENABLE_SNMP_TRAPS="true" \
  --network=zabbix-net \
  -p 10051:10051 \
  --volumes-from zabbix-snmptraps \
  --restart unless-stopped \
  --init \
  -d zabbix/zabbix-server-pgsql:alpine-8.0-latest
```

5. Suivez les journaux du serveur Zabbix et attendez que le serveur Zabbix ait terminé l'initialisation du schéma de la base de données (utilisez Ctrl+C pour quitter les journaux) :

```
docker logs -f zabbix-server-pgsql

#### ... [info]: ** Creating 'zabbix' schema in PostgreSQL
#### ... [info]: ** Database schema successfully created!
```

6. Exécutez un conteneur d'interface web Zabbix et liez-le aux conteneurs du serveur Zabbix et du serveur PostgreSQL :

```
docker run --name zabbix-web-nginx-pgsql -t \
  -e ZBX_SERVER_HOST="zabbix-server-pgsql" \
  -e DB_SERVER_HOST="postgres-server" \
  -e POSTGRES_DB="zabbix" \
  -e POSTGRES_USER="zabbix" \
  -e POSTGRES_PASSWORD="zabbix_pwd" \
  -e PHP_TZ="Europe/Riga" \
  --network=zabbix-net \
  -p 443:8443 \
  -p 80:8080 \
  -v /etc/ssl/nginx:/etc/ssl/nginx:ro \
  --restart unless-stopped \
  -d zabbix/zabbix-web-nginx-pgsql:alpine-8.0-latest
```

Une fois les conteneurs démarrés et opérationnels, le serveur Zabbix démarre et l'interface web devient disponible pour la connexion.

> **Note:**
> Utilisez `docker ps` pour vérifier l'état des conteneurs. Tous les conteneurs doivent avoir l'état `Up`. Si l'un d'eux affiche l'état `Exited`, consultez ses journaux pour rechercher les erreurs avec `docker logs -f <container-name>`.

Déployer le serveur Zabbix (MySQL) avec Java gateway sur RHEL 8-10

Cet exemple montre comment déployer un serveur Zabbix avec un backend MySQL, une Java gateway pour la supervision JMX et une interface basée sur NGINX, le tout exécuté sur Red Hat Enterprise Linux 8, 9 ou 10.

Sur Red Hat Enterprise Linux, l'environnement d'exécution de conteneurs recommandé est Podman plutôt que Docker. Podman fonctionne de manière similaire à Docker, mais ne nécessite pas l'exécution en arrière-plan d'un service avec les privilèges root, ce qui en fait une solution mieux adaptée aux environnements Red Hat.

1. Créez un nouveau pod nommé `zabbix` et exposez les ports de l'interface web Zabbix et du trapper du serveur Zabbix :

```
podman pod create --name zabbix -p 80:8080 -p 10051:10051
```

2. (Facultatif) Exécutez un conteneur agent Zabbix dans le pod `zabbix` :

```
podman run --name zabbix-agent \
  -e ZBX_SERVER_HOST="127.0.0.1,localhost" \
  --restart=always \
  --pod=zabbix \
  -d registry.connect.redhat.com/zabbix/zabbix-agent-80:latest
```

3. Créez un répertoire `./mysql/` sur l'hôte RHEL :

```
mkdir -p ./mysql
```

4. Exécutez un conteneur de serveur MySQL vide (remplacez `zabbix_pwd` et `root_pwd` par des mots de passe robustes et utilisez ces valeurs lors des étapes suivantes) :

```
podman run --name mysql-server -t \
  -e MYSQL_DATABASE="zabbix" \
  -e MYSQL_USER="zabbix" \
  -e MYSQL_PASSWORD="zabbix_pwd" \
  -e MYSQL_ROOT_PASSWORD="root_pwd" \
  -v ./mysql/:/var/lib/mysql/:Z \
  --restart=always \
  --pod=zabbix \
  -d mysql:8.4 \
  --character-set-server=utf8mb4 --collation-server=utf8mb4_bin
```

5. Suivez les journaux MySQL et attendez que MySQL soit prêt à accepter les connexions (utilisez `Ctrl+C` pour quitter les journaux) :

```
podman logs -f mysql-server

#### ... [Server] /usr/sbin/mysqld: ready for connections. ...
```

6. Exécutez un conteneur Java gateway Zabbix :

```
podman run --name zabbix-java-gateway -t \
  --restart=always \
  --pod=zabbix \
  -d registry.connect.redhat.com/zabbix/zabbix-java-gateway-80
```

7. Activez `log_bin_trust_function_creators` dans le conteneur du serveur MySQL. Cette option assouplit temporairement les restrictions de sécurité appliquées aux fonctions stockées ; elle sera désactivée lors des étapes suivantes. Elle permet à l'utilisateur de la base de données Zabbix de créer des fonctions stockées sans le privilège `SUPER`, requis par MySQL lorsque la journalisation binaire est activée (par défaut depuis MySQL 8.0). Exécutez la commande suivante :

```
podman exec -it mysql-server mysql -u root -p'root_pwd' \
  -e "SET GLOBAL log_bin_trust_function_creators = 1;"
```

8. Exécutez un conteneur de serveur Zabbix et liez-le aux conteneurs Java Gateway et du serveur MySQL :

```
podman run --name zabbix-server-mysql -t \
  -e DB_SERVER_HOST="127.0.0.1" \
  -e MYSQL_DATABASE="zabbix" \
  -e MYSQL_USER="zabbix" \
  -e MYSQL_PASSWORD="zabbix_pwd" \
  -e ZBX_JAVAGATEWAY="127.0.0.1" \
  --restart=always \
  --init \
  --pod=zabbix \
  -d registry.connect.redhat.com/zabbix/zabbix-server-mysql-80
```

9. Suivez les journaux du serveur Zabbix et attendez que le serveur Zabbix ait terminé l'initialisation du schéma de la base de données (utilisez `Ctrl+C` pour quitter les journaux) :

```
podman logs -f zabbix-server-mysql

#### ... [info]: ** Creating 'zabbix' schema in MySQL
#### ... [info]: ** Database schema successfully created!
```

10. Désactivez `log_bin_trust_function_creators` dans le conteneur du serveur MySQL. Cette opération rétablit la restriction de sécurité qui empêche les utilisateurs ne disposant pas du privilège `SUPER` de créer des fonctions stockées. Exécutez la commande suivante :

```
podman exec -it mysql-server mysql -u root -p'root_pwd' \
  -e "SET GLOBAL log_bin_trust_function_creators = 0;"
```

11. Exécutez un conteneur d'interface web Zabbix et liez-le aux conteneurs du serveur Zabbix et du serveur MySQL :

```
podman run --name zabbix-web-mysql -t \
  -e ZBX_SERVER_HOST="127.0.0.1" \
  -e DB_SERVER_HOST="127.0.0.1" \
  -e MYSQL_DATABASE="zabbix" \
  -e MYSQL_USER="zabbix" \
```

```
  -e MYSQL_PASSWORD="zabbix_pwd" \
  -e PHP_TZ="Europe/Riga" \
  --restart=always \
  --pod=zabbix \
  -d registry.connect.redhat.com/zabbix/zabbix-web-mysql-80
```

Note:
Le pod `zabbix` expose le port 80/TCP (HTTP) à la machine hôte à partir du port 8080/TCP du conteneur `zabbix-web-mysql`.

Une fois les conteneurs démarrés et opérationnels, le serveur Zabbix démarre et l'interface web devient disponible pour la connexion.

Note:
Utilisez `docker ps` pour vérifier l'état des conteneurs. Tous les conteneurs doivent avoir l'état `Up`. Si l'un d'eux affiche l'état `Exited`, consultez ses journaux pour rechercher les erreurs avec `docker logs -f <container-name>`.

Installation à partir des sources

Vous pouvez obtenir la toute dernière version de Zabbix en la compilant à partir des sources. Voir aussi Obtention du code source de Zabbix.

Un tutoriel étape par étape pour installer Zabbix à partir des sources est fourni ici.

Installation des démons Zabbix

1 Téléchargez l'archive des sources

Accédez à la page de téléchargement de Zabbix et téléchargez l'archive des sources. Une fois le téléchargement terminé, extrayez les sources en exécutant :

```
tar -zxvf zabbix-8.0.0.tar.gz
```

Note:
Saisissez la version correcte de Zabbix dans la commande. Elle doit correspondre au nom de l'archive téléchargée.

2 Créer un compte utilisateur

Tous les processus daemon Zabbix s'exécutent sous des utilisateurs système non privilégiés.
Si un daemon Zabbix est démarré depuis un compte utilisateur non privilégié, il continuera à s'exécuter sous cet utilisateur.

Dans la configuration par défaut, si un daemon est démarré en tant que `root`, il basculera vers le compte utilisateur `zabbix`, qui doit donc exister. Pour créer un utilisateur système et un groupe `zabbix`, exécutez les commandes indiquées ci-dessous.

Système basé sur RedHat :

```
groupadd --system zabbix
useradd --system -g zabbix -d /usr/lib/zabbix -s /sbin/nologin -c "Zabbix Monitoring System" zabbix
```

Système basé sur Debian :

```
addgroup --system --quiet zabbix
adduser --quiet --system --disabled-login --ingroup zabbix --home /var/lib/zabbix --no-create-home zabbix
```

Il n'est pas nécessaire de créer un compte utilisateur distinct pour le frontend Zabbix.

Recommandation de sécurité

Si le server et l'agent Zabbix s'exécutent sur la même machine, il est recommandé de les exécuter sous **des comptes utilisateur distincts**. Le fait d'exécuter les deux sous le même utilisateur permet à l'agent d'accéder au fichier de configuration du serveur, ce qui pourrait exposer des informations sensibles, comme le mot de passe de la base de données, à tout utilisateur de niveau Admin dans Zabbix.

Attention:
Exécuter Zabbix en tant que `root`, `bin` ou tout autre compte disposant de privilèges spéciaux constitue un risque de sécurité.

Répertoire personnel (facultatif)

Les processus Zabbix n'ont pas besoin d'un répertoire personnel ; sa création n'est donc généralement pas recommandée. Cependant, si vous avez besoin d'une fonctionnalité nécessitant un répertoire personnel (par exemple, pour stocker les identifiants MySQL dans $HOME/.my.cnf), vous pouvez le créer à l'aide des commandes indiquées ci-dessous.

Sur les systèmes basés sur RedHat, exécutez :

```
mkdir -m u=rwx,g=rwx,o= -p /usr/lib/zabbix
chown zabbix:zabbix /usr/lib/zabbix
```

Sur les systèmes basés sur Debian, exécutez :

```
mkdir -m u=rwx,g=rwx,o= -p /var/lib/zabbix
chown zabbix:zabbix /var/lib/zabbix
```

3 Créer la base de données Zabbix

Pour les démons Zabbix server et proxy, ainsi que pour le frontend Zabbix, une base de données est requise. Elle n'est pas nécessaire pour exécuter l' agent Zabbix.

Des scripts SQL sont fournis pour créer le schéma de la base de données et insérer le jeu de données. La base de données du proxy Zabbix nécessite uniquement le schéma, tandis que la base de données du serveur Zabbix requiert également le jeu de données en plus du schéma.

Après avoir créé une base de données Zabbix, passez aux étapes suivantes de la compilation de Zabbix.

4 Configurer les sources

C99 avec les extensions GNU est requis pour compiler le serveur Zabbix, le proxy Zabbix ou l'agent Zabbix. Cette version peut être spécifiée explicitement en définissant CFLAGS="-std=gnu99" :

```
export CFLAGS="-std=gnu99"
```

> **Note:**
> Si l'installation est effectuée depuis le dépôt Git de Zabbix, il est nécessaire d'exécuter d'abord :
> `./bootstrap.sh`

Lors de la configuration des sources pour un serveur Zabbix ou un proxy, vous devez spécifier le type de base de données à utiliser. Un seul type de base de données peut être compilé avec un processus serveur ou proxy à la fois.

Pour afficher toutes les options de configuration prises en charge, exécutez la commande suivante dans le répertoire des sources Zabbix extrait :

```
./configure --help
```

Pour configurer les sources d'un serveur et d'un agent Zabbix, vous pouvez exécuter une commande similaire à :

```
./configure --enable-server --enable-agent --with-mysql --enable-ipv6 --with-net-snmp --with-libcurl --wit
```

Pour configurer les sources d'un serveur Zabbix (avec PostgreSQL, etc.), vous pouvez exécuter :

```
./configure --enable-server --with-postgresql --with-net-snmp
```

Pour configurer les sources d'un proxy Zabbix (avec SQLite, etc.), vous pouvez exécuter :

```
./configure --prefix=/usr --enable-proxy --with-net-snmp --with-sqlite3 --with-ssh2
```

Pour configurer les sources d'un agent Zabbix, vous pouvez exécuter :

```
./configure --enable-agent
```

ou, pour l'agent Zabbix 2 :

```
./configure --enable-agent2
```

> **Note:**
> Une version de Go prise en charge doit être installée pour compiler l'agent Zabbix 2.

Remarques sur les options de compilation :

- `--enable-agent` - compile l'agent Zabbix, ainsi que les utilitaires en ligne de commande Zabbix get et Zabbix sender.
- `--with-libcurl` - requis pour la surveillance des machines virtuelles, l'authentification SMTP, les éléments de l'agent Zabbix `web.page.*` et les éléments de requête de télémétrie. Voir également : Configuration requise (libcurl).
- `--with-libxml2` - requis pour la surveillance des machines virtuelles.

- `--with-libpcre2[=DIR]` - Zabbix est toujours compilé avec la bibliothèque PCRE2 ; cette option permet uniquement de spécifier un chemin d'installation personnalisé de PCRE2.
- `--with-mysql=/path/to/mysql_config` - spécifie le chemin vers la configuration d'une bibliothèque cliente MySQL particulière. Utile lorsque plusieurs versions de MySQL ou MariaDB sont installées.
- `--enable-static` - lie statiquement les bibliothèques (non pris en charge sur Solaris). Utilisez cette option si vous prévoyez de distribuer des binaires compilés sur des systèmes dépourvus des bibliothèques requises. Cette option n'est pas recommandée lors de la compilation du serveur Zabbix. Pour compiler le serveur de manière statique, une version statique de chaque bibliothèque externe est requise. Le script configure ne vérifie pas automatiquement ce point.
- `--with-stacksize=<value>` - définit la taille de pile par thread en kilo-octets (par exemple, `--with-stacksize=512`). Vous pouvez augmenter cette valeur si Zabbix se bloque ou se fige en raison de débordements de pile (par exemple, lors du prétraitement sur des systèmes dont la limite de pile par thread par défaut est faible).
- `--with-apm` (à utiliser uniquement avec `--enable-proxy`) - requis pour la surveillance des performances des applications (APM) sur un proxy Zabbix. Voir également : Configuration requise (g++, gRPC, protobuf, Abseil).

Attention:
Si `./configure` échoue en raison de bibliothèques manquantes ou d'autres problèmes, consultez le fichier `config.log` pour obtenir des informations détaillées sur l'erreur.
Par exemple, si `libssl` est manquante, le message d'erreur immédiat peut être trompeur :
```
checking for main in -lmysqlclient... no
configure: error: Not found mysqlclient library
```
Dans ce cas, `config.log` révèle la cause réelle :
```
/usr/bin/ld: cannot find -lssl
/usr/bin/ld: cannot find -lcrypto
```

Voir également :

- Compiler Zabbix avec la prise en charge du chiffrement
- Problèmes de compilation connus

5 Construire et installer l'ensemble

Note:
Si l'installation se fait depuis le dépôt Git Zabbix, il est nécessaire d'exécuter d'abord :
```
$ make dbschema
```

```
make install
```

Cette étape doit être exécutée par un utilisateur disposant de permissions suffisantes (généralement `root`, ou en utilisant `sudo`).

L'exécution de `make install` installera par défaut les binaires du démon (zabbix_server, zabbix_agentd, zabbix_proxy) dans /usr/local/sbin et les binaires client (zabbix_get, zabbix_sender) dans /usr/local/bin.

Note:
Pour spécifier un emplacement différent de `/usr/local`, utilisez une clé `--prefix` à l'étape précédente de configuration des sources, par exemple `--prefix=/home/zabbix`. Dans ce cas, les binaires du démon seront installés dans `<prefix>/sbin`, tandis que les utilitaires seront installés dans `<prefix>/bin`. Les pages de manuel seront installées dans `<prefix>/share`.

6 Examiner et modifier les fichiers de configuration

- modifier le fichier de configuration de l'agent Zabbix (`zabbix_agentd.conf`).

Vous devez configurer ce fichier pour chaque hôte sur lequel `zabbix_agentd` est installé.

Vous devez spécifier l'**adresse IP** du serveur Zabbix dans le fichier. Les connexions provenant d'autres hôtes seront refusées.

- modifier le fichier de configuration du serveur Zabbix (`zabbix_server.conf`).

Vous devez spécifier le nom de la base de données, l'utilisateur et le mot de passe (le cas échéant).

Les autres paramètres par défaut devraient vous convenir si vous disposez d'une petite installation (jusqu'à dix hôtes surveillés). Vous devriez toutefois modifier les paramètres par défaut si vous souhaitez optimiser les performances du serveur Zabbix (ou du proxy).

- si vous avez installé un proxy Zabbix, modifier le fichier de configuration du proxy (`zabbix_proxy.conf`).

Vous devez spécifier l'adresse IP du serveur et le nom d'hôte du proxy (qui doit être connu du serveur), ainsi que le nom de la base de données, l'utilisateur et le mot de passe (le cas échéant).

Note:
Avec SQLite, le chemin complet vers le fichier de base de données doit être spécifié ; l'utilisateur et le mot de passe de la base de données ne sont pas requis.

7 Démarrer les daemons

Exécutez zabbix_server du côté serveur.

```
zabbix_server
```

Note:
Assurez-vous que votre système autorise l'allocation de 36 Mo (ou un peu plus) de mémoire partagée, sinon le serveur risque de ne pas démarrer et vous verrez "Cannot allocate shared memory for <type of cache>." dans le fichier journal du serveur. Cela peut se produire sur FreeBSD, Solaris 8.

Exécutez zabbix_agentd sur toutes les machines surveillées.

```
zabbix_agentd
```

Note:
Assurez-vous que votre système autorise l'allocation de 2 Mo de mémoire partagée, sinon l'agent risque de ne pas démarrer et vous verrez "Cannot allocate shared memory for collector." dans le fichier journal de l'agent. Cela peut se produire sur Solaris 8.

Si vous avez installé Zabbix proxy, exécutez zabbix_proxy.

```
zabbix_proxy
```

Installation de l'interface web Zabbix

Copie des fichiers PHP

L'interface Zabbix est écrite en PHP, donc pour l'exécuter, un serveur web prenant en charge PHP est nécessaire. L'installation se fait simplement en copiant les fichiers PHP du répertoire ui vers le répertoire des documents HTML du serveur web.

Les emplacements courants des répertoires de documents HTML pour les serveurs web Apache incluent :

- /usr/local/apache2/htdocs (répertoire par défaut lors de l'installation d'Apache à partir des sources)
- /srv/www/htdocs (OpenSUSE, SLES)
- /var/www/html (Debian, Ubuntu, Fedora, RHEL)

Il est recommandé d'utiliser un sous-répertoire plutôt que la racine HTML. Pour créer un sous-répertoire et y copier les fichiers de l'interface Zabbix, exécutez les commandes suivantes, en remplaçant <htdocs> par le répertoire réel :

```
mkdir <htdocs>/zabbix
cd ui
cp -a . <htdocs>/zabbix
```

Si vous prévoyez d'utiliser une autre langue que l'anglais, consultez Installation de langues supplémentaires de l'interface pour obtenir des instructions.

Installation de l'interface web

Veuillez consulter la page Installation de l'interface web pour obtenir des informations sur l'assistant d'installation de l'interface web de Zabbix.

Installation des plugins chargeables de Zabbix agent 2

L'installation des plugins chargeables de Zabbix agent 2 n'est nécessaire que si vous souhaitez surveiller des cibles non couvertes par les plugins intégrés (par exemple, des serveurs ou des clusters MongoDB, PostgreSQL et ses dérivés, etc.). Consultez la liste complète des plugins chargeables et des plugins intégrés.

Attention:
Avant d'installer un plugin, veuillez consulter son fichier README. Il peut contenir des exigences spécifiques et des instructions d'installation.

Pour installer à partir des sources, commencez par télécharger puis extraire l'archive source du plugin chargeable.

Pour compiler le plugin, accédez au répertoire du plugin extrait et exécutez `make` :

```
make
```

> **Note:**
> Une version prise en charge de Go doit être installée pour compiler les plugins chargeables de Zabbix agent 2.

L'exécutable du plugin peut être placé n'importe où, à condition qu'il puisse être chargé par Zabbix agent 2. Spécifiez le chemin vers le binaire du plugin dans le fichier de configuration du plugin, par exemple dans `postgresql.conf` pour le plugin PostgreSQL :

```
Plugins.PostgreSQL.System.Path=/path/to/executable/zabbix-agent2-plugin-postgresql
```

Le chemin vers le fichier de configuration du plugin doit être spécifié dans le paramètre `Include` du fichier de configuration de Zabbix agent 2 :

```
Include=/path/to/plugin/configuration/file/postgresql.conf
```

Passez à la section configuration pour plus de détails sur la configuration des plugins.

Les plugins chargeables fournis par Zabbix utilisent des makefiles simples avec les cibles de compilation suivantes :

- `make` - compiler le plugin
- `make clean` - supprimer tous les fichiers créés lors de la compilation du plugin
- `make check` - exécuter les auto-tests (nécessite une vraie cible de surveillance, par exemple une base de données PostgreSQL)
- `make style` - vérifier le style du code Go avec `golangci-lint`
- `make format` - formater le code Go avec `go fmt`
- `make dist` - créer une archive source incluant toutes les dépendances

Installation de Java gateway

Il est nécessaire d'installer Java gateway uniquement si vous souhaitez surveiller des applications JMX. Java gateway est léger et ne nécessite pas de base de données.

Pour installer à partir des sources, commencez par télécharger et extraire l'archive des sources.

Pour compiler Java gateway, exécutez le script `./configure` avec l'option `--enable-java`. Il est recommandé de spécifier l'option `--prefix` pour demander un chemin d'installation autre que le chemin par défaut /usr/local, car l'installation de Java gateway créera une arborescence complète de répertoires, et non pas un seul exécutable.

```
./configure --enable-java --prefix=$PREFIX
```

Pour compiler et empaqueter Java gateway dans un fichier JAR, exécutez `make`. Notez que pour cette étape, vous aurez besoin des exécutables `javac` et `jar` dans votre chemin d'accès.

```
make
```

Vous disposez maintenant d'un fichier zabbix-java-gateway-$VERSION.jar dans `src/zabbix_java/bin`. Si vous êtes à l'aise avec l'exécution de Java gateway depuis `src/zabbix_java` dans le répertoire de distribution, vous pouvez alors passer aux instructions de configuration et d'exécution de Java gateway. Sinon, assurez-vous d'avoir les privilèges suffisants et exécutez `make install`.

```
make install
```

Consultez la configuration pour plus de détails sur la configuration et l'exécution de Java gateway.

Installation du service web Zabbix

L'installation du service web Zabbix n'est nécessaire que si vous souhaitez utiliser les rapports planifiés.

Pour installer à partir des sources, commencez par télécharger et extraire l'archive source.

Pour compiler le service web Zabbix, exécutez le script `./configure` avec l'option `--enable-webservice`.

> **Note:**
> Une version Go prise en charge doit être installée pour compiler le service web Zabbix.

Exécutez zabbix_web_service sur la machine où le service web est installé :

```
zabbix_web_service
```

Consultez la section configuration pour plus de détails sur la configuration de la génération des rapports planifiés.

Installation du serveur MCP Zabbix

Pour installer à partir des sources, commencez par télécharger et extraire l'archive source.

Pour compiler le serveur MCP Zabbix, exécutez la commande suivante dans le répertoire des sources du serveur MCP :

```
make build
```

> **Note:**
> Une version de Go prise en charge doit être installée pour compiler le serveur MCP Zabbix.

Pour indiquer une version spécifique dans la sortie de `zabbix_mcp_server -V` et dans l'en-tête `User-Agent` des requêtes de l'API Zabbix, définissez la variable `VERSION`, par exemple : `make build VERSION=8.0.0`.

Pour installer le binaire dans `/usr/local/sbin` et le fichier de configuration d'exemple dans `/etc/zabbix`, exécutez :

```
sudo make install
```

Modifiez le fichier de configuration `/etc/zabbix/zabbix_mcp_server.conf` et définissez au moins les paramètres suivants :

- `FrontendURL` - l'URL de l'API Zabbix, se terminant par `api_jsonrpc.php` ;
- `AllowedIP` - les adresses des clients MCP autorisés à se connecter.

Pour chiffrer les connexions des clients MCP, définissez `TLSAccept=cert` et indiquez `TLSCertFile` et `TLSKeyFile`, ou exécutez le serveur MCP derrière un proxy inverse de confiance qui gère TLS.

Si l'utilisateur `zabbix` n'existe pas (par exemple, si seul le serveur MCP est installé sur la machine), créez-le comme décrit dans Créer un compte utilisateur.

Validez le fichier de configuration et démarrez zabbix_mcp_server :

```
zabbix_mcp_server -c /etc/zabbix/zabbix_mcp_server.conf -T
zabbix_mcp_server -c /etc/zabbix/zabbix_mcp_server.conf
```

Consultez Serveur MCP pour plus d'informations sur la configuration du serveur et la connexion des clients MCP.

Obtention du code source de Zabbix

Il existe plusieurs façons d'obtenir le code source de Zabbix :

- Vous pouvez télécharger les versions stables publiées depuis le site officiel de Zabbix
- Vous pouvez télécharger les compilations nightly depuis la page développeur du site officiel de Zabbix
- Vous pouvez obtenir la dernière version de développement depuis le système de dépôt de code source Git :
  - L'emplacement principal du dépôt complet est https://git.zabbix.com/scm/zbx/zabbix.git
  - La branche master et les versions prises en charge sont également mises en miroir sur Github à l'adresse https://github.com/zabbix/zabbix

Un client Git doit être installé pour cloner le dépôt. Le paquet officiel du client Git en ligne de commande s'appelle généralement **git** dans les distributions. Pour l'installer, par exemple, sur Debian/Ubuntu, exécutez :

```
sudo apt-get update
sudo apt-get install git
```

Pour récupérer l'ensemble du code source de Zabbix, placez-vous dans le répertoire où vous souhaitez enregistrer le code, puis exécutez :

```
git clone https://git.zabbix.com/scm/zbx/zabbix.git
```

Problèmes de compilation

Voici les problèmes connus concernant la compilation de Zabbix à partir des sources. Pour tous les autres cas, consultez la page Problèmes connus.

Bibliothèque dans un emplacement non standard

Zabbix vous permet de spécifier une bibliothèque située dans un emplacement non standard. Dans l'exemple ci-dessous, Zabbix exécutera `curl-config` depuis l'emplacement non standard spécifié et utilisera sa sortie pour déterminer la bonne version de libcurl à utiliser.

```
$ ./configure --enable-server --with-mysql --with-libcurl=/usr/local/bin/curl-config
```

Cela fonctionnera si c'est la seule version de libcurl installée sur le système, mais ce ne sera pas forcément le cas si une autre version de libcurl est installée dans un emplacement standard (par le gestionnaire de paquets, par exemple). C'est notamment le cas lorsque vous avez besoin d'une version plus récente de la bibliothèque pour Zabbix et de l'ancienne pour d'autres applications.

Par conséquent, spécifier un composant dans un emplacement non standard ne fonctionnera pas toujours si le même composant existe également dans un emplacement standard.

Par exemple, si vous utilisez une version plus récente de libcurl installée dans `/usr/local` alors que le paquet libcurl est toujours installé, Zabbix peut sélectionner la mauvaise version et la compilation échouera :

```
usr/bin/ld: ../../src/libs/zbxhttp/libzbxhttp.a(http.o): in function 'zbx_http_convert_to_utf8':
/tmp/zabbix-master/src/libs/zbxhttp/http.c:957: undefined reference to 'curl_easy_header'
collect2: error: ld returned 1 exit status
```

Ici, la fonction `curl_easy_header()` n'est pas disponible dans l'ancienne bibliothèque `/usr/lib/x86_64-linux-gnu/libcurl.so`, mais elle est disponible dans la plus récente `/usr/local/lib/libcurl.so`.

Le problème vient de l'ordre des options de l'éditeur de liens, et une solution consiste à spécifier le chemin complet vers la bibliothèque dans une variable LDFLAGS :

```
$ LDFLAGS="-Wl,--no-as-needed /usr/local/lib/libcurl.so" ./configure --enable-server --with-mysql --with-l
```

Notez l'option `-Wl,--no-as-needed`, qui peut être nécessaire sur certains systèmes (voir aussi : options d'édition de liens par défaut sur les systèmes basés sur Debian).

Taille de pile trop faible sur certains systèmes

Si Zabbix plante ou se fige en raison de débordements de pile, vous pouvez augmenter la taille de la pile par thread à l'aide de l'option `--with-stacksize` lors de la configuration des sources. Ce problème peut se produire sur des systèmes dont les limites par défaut de la pile des threads sont faibles, en particulier pendant le prétraitement, où plusieurs threads sont créés.

L'exemple suivant définit la taille de la pile à 512 Ko par thread :

```
./configure --enable-server --with-mysql --with-stacksize=512
```

Vous pouvez vérifier les limites de pile des threads du système à l'exécution à l'aide de la commande `ulimit -s` sur les systèmes basés sur Linux.

Remarques sur l'installation à partir de paquets

Présentation

Les paquets officiels **préliminaires** de Zabbix 8.0 sont disponibles sur le site web de Zabbix, où vous pouvez sélectionner votre système d'exploitation et le composant Zabbix afin de générer les instructions et les commandes d'installation.

En plus des notes sur l'installation des paquets ci-dessous, veuillez également consulter la page Installation et configuration.

Les paquets sont disponibles pour les distributions Linux suivantes :

- Red Hat Enterprise Linux et ses dérivées : AlmaLinux, Amazon Linux 2023, CentOS Stream, CentOS 7, Oracle Linux, Rocky Linux
- Debian, Ubuntu, Raspberry Pi OS, Raspbian
- SUSE Linux Enterprise Server, openSUSE Leap

> **Attention:**
> Certaines distributions de systèmes d'exploitation, notamment les distributions basées sur Debian, fournissent leurs propres paquets Zabbix. Ces paquets ne sont **pas** pris en charge par Zabbix et peuvent être obsolètes ou ne pas inclure les dernières fonctionnalités et corrections de bogues. Il est recommandé d'utiliser uniquement les paquets officiels du dépôt officiel Zabbix. Si vous avez précédemment installé Zabbix depuis le dépôt de votre système d'exploitation, consultez les étapes pour mettre à niveau les paquets Zabbix provenant des dépôts du système d'exploitation.

Les paquets prennent en charge les bases de données MySQL/PostgreSQL et les serveurs web Apache/Nginx. Notez que le serveur Zabbix et le proxy ne peuvent pas partager la même base de données ; utilisez des noms de base de données différents s'ils sont tous deux installés sur le même hôte.

Si nécessaire, des paquets distincts pour l'agent Zabbix/agent 2, Zabbix get et Zabbix sender sont disponibles dans le dépôt officiel Zabbix.

Sur les systèmes basés sur Debian ou Ubuntu, l'installation d'un paquet Zabbix supplémentaire avec `apt` peut également sélectionner les mises à niveau des paquets Zabbix déjà installés depuis le même dépôt. Pour installer un paquet sans mettre à niveau un paquet de serveur Zabbix ou de proxy déjà installé, bloquez temporairement ce paquet avant l'installation, puis débloquez-le ensuite :

```
apt-mark hold zabbix-server-mysql
apt install zabbix-sender
apt-mark unhold zabbix-server-mysql
```

Zabbix fournit également des binaires précompilés de l'agent Zabbix pour les systèmes d'exploitation autres que Linux ; consultez :

- Installation de l'agent Windows depuis un fichier MSI
- Installation de l'agent macOS depuis un fichier PKG
- Agent Zabbix sur IBM AIX
- Binaires hérités (pour les systèmes plus anciens ou moins courants, tels que HP-UX, NetBSD, Tru64 et les anciennes versions de SLES)

Remarques sur l'installation des paquets

Les remarques suivantes s'appliquent à tous les systèmes :

- Si vous utilisez PostgreSQL, `DBHost=localhost` (ou une adresse IP) dans la configuration du serveur/proxy Zabbix oblige PostgreSQL à utiliser un socket réseau au lieu d'un socket UNIX local ; consultez Configuration de SELinux pour obtenir les instructions de configuration associées.
- Si vous utilisez TimescaleDB, consultez les instructions supplémentaires de configuration de TimescaleDB.
- Si vous installez le Java gateway Zabbix (pour surveiller les applications JMX), consultez les instructions de configuration supplémentaires pour les systèmes basés sur RHEL et les systèmes basés sur Debian.
- Si vous installez le serveur MCP Zabbix, installez le paquet `zabbix-mcp-server` et mettez à jour le fichier de configuration (au moins `FrontendURL` et `AllowedIP`) avant de démarrer le service.
- Pour exécuter l'agent Zabbix en tant que root, consultez Exécution de l'agent en tant que root.
- La surveillance des performances des applications (APM) sur un proxy Zabbix est prise en charge par les paquets suivants :
  - RHEL 10 et ses dérivés - installez le paquet APM du proxy Zabbix (par exemple, `zabbix-proxy-mysql-apm` au lieu de `zabbix-proxy-mysql`) ; il nécessite également l'installation du dépôt EPEL et l'activation des dépôts de paquets de dépendances CRB.
  - Debian 12 ou version ultérieure, Raspberry Pi OS (Raspbian) 12 ou version ultérieure, Ubuntu 24.04 ou version ultérieure - APM est inclus dans les paquets du proxy Zabbix.
  - Sur les autres systèmes, vous pouvez installer le proxy Zabbix à partir des sources avec l'option `--with-apm`.

Les remarques suivantes s'appliquent à RHEL et à ses dérivés :

- Si vous avez activé le dépôt EPEL (par exemple, pour le paquet APM du proxy Zabbix), qui fournit également des paquets Zabbix, celui-ci doit être exclu de la résolution des paquets avant l'installation des paquets Zabbix officiels ; consultez Installation accidentelle de paquets Zabbix depuis EPEL.
- Pour installer les paquets Zabbix dans des environnements Red Hat UBI, consultez Paquets Zabbix pour RHEL dans les environnements Red Hat UBI.
- Pour utiliser les éléments de vérification ping ICMP, des paquets pour `fping` sont également disponibles dans le dépôt officiel Zabbix.

Configuration SELinux

Zabbix utilise une communication inter-processus basée sur des sockets. Sur les systèmes où Security-Enhanced Linux (SELinux) est activé, vous devrez peut-être ajouter des règles SELinux pour autoriser Zabbix à créer/utiliser des sockets de domaine UNIX dans le répertoire SocketDir. Les fichiers socket sont utilisés par le serveur Zabbix (alerter, prétraitement, IPMI) et le proxy Zabbix (IPMI), et ils sont présents tant que le processus est en cours d'exécution.

Avec SELinux activé en mode enforcing, exécutez les commandes suivantes pour autoriser la communication entre l'interface Zabbix et le serveur :

Pour RHEL 7 (et versions ultérieures), AlmaLinux, CentOS Stream, Oracle Linux, Rocky Linux 8 (et versions ultérieures) :

```
setsebool -P httpd_can_connect_zabbix on
```

Si la base de données est accessible via le réseau (y compris `localhost` pour PostgreSQL), autorisez également l'interface Zabbix à se connecter à la base de données :

```
setsebool -P httpd_can_network_connect_db on
```

Pour RHEL antérieur à la version 7 :

```
setsebool -P httpd_can_network_connect on
setsebool -P zabbix_can_network on
```

Après avoir appliqué les paramètres SELinux, redémarrez Apache :

```
systemctl restart httpd
```

En option, vous pouvez installer un package `zabbix-selinux-policy` prédéfini depuis le Zabbix Official Repository. Ce package est fourni pour toutes les versions d'OS prises en charge afin de simplifier le déploiement de Zabbix et d'éviter que les utilisateurs ne désactivent SELinux en raison de la complexité de la configuration.
 Pour la version 10 d'AlmaLinux, CentOS Stream,

RHEL, Rocky Linux et Oracle Linux, la politique SELinux est fournie par le package `zabbix-selinux-policy-main`; le package `zabbix-selinux-policy` dépend soit de `zabbix-selinux-policy-main`, soit de `selinux-policy-targeted-extra`. Le package `zabbix-selinux-policy-main` entre en conflit avec `selinux-policy-targeted-extra`; par conséquent, les deux packages ne peuvent pas être installés en même temps.

> **Attention:**
> Pour une sécurité maximale, il est recommandé de définir des paramètres SELinux personnalisés.

Le package `zabbix-selinux-policy` (`zabbix-selinux-policy-main` pour la version 10 d'AlmaLinux, CentOS Stream, RHEL, Rocky Linux et Oracle Linux) contient une politique SELinux de base qui autorise Zabbix à créer et utiliser des sockets et permet à HTTPd d'établir une connexion à PostgreSQL (utilisé par l'interface).

Le fichier source `zabbix_policy.te` contient les règles suivantes :

```
module zabbix_policy 1.2;

require {
    type zabbix_t;
    type zabbix_port_t;
    type zabbix_var_run_t;
    type postgresql_port_t;
    type httpd_t;
    class tcp_socket name_connect;
    class sock_file { create unlink };
    class unix_stream_socket connectto;
}

####============= zabbix_t ==============
allow zabbix_t self:unix_stream_socket connectto;
allow zabbix_t zabbix_port_t:tcp_socket name_connect;
allow zabbix_t zabbix_var_run_t:sock_file create;
allow zabbix_t zabbix_var_run_t:sock_file unlink;
allow httpd_t zabbix_port_t:tcp_socket name_connect;

####============= httpd_t ==============
allow httpd_t postgresql_port_t:tcp_socket name_connect;
```

Paquets Debuginfo

Les paquets `Debuginfo` contiennent les symboles de débogage pour les binaires Zabbix. Ils ne sont pas requis pour une installation ou un fonctionnement normal, mais sont utiles pour un dépannage avancé.

Veuillez noter que les paquets `Debuginfo` sont disponibles pour les versions suivantes: Red Hat Enterprise Linux 7 et antérieures (y compris les dérivés), et SUSE Linux Enterprise Server 15 et antérieures (y compris les dérivés).

Pour activer le dépôt `zabbix-debuginfo`:

- Sur RHEL 7, modifiez `/etc/yum.repos.d/zabbix.repo` et définissez `enabled=1` pour la section `zabbix-debuginfo`:

```
[zabbix-debuginfo]
name=Zabbix Official Repository debuginfo - $basearch
baseurl=http://repo.zabbix.com/zabbix/8.0/stable/rhel/7/$basearch/debuginfo/
enabled=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-ZABBIX-A14FE591
gpgcheck=1
```

- Sur SUSE, modifiez `/etc/zypp/repos.d/zabbix.repo` et définissez `enabled=1` pour la section `zabbix-debuginfo`:

```
[zabbix-debuginfo]
name=Zabbix Official Repository debuginfo
type=rpm-md
baseurl=https://repo.zabbix.com/zabbix/8.0/unstable/sles/15/$basearch/debuginfo/
gpgcheck=1
gpgkey=https://repo.zabbix.com/zabbix/8.0/unstable/sles/15/$basearch/debuginfo/repodata/repomd.xml.key
enabled=0
update=1
```

Une fois activé, installez les paquets:

- Sur RHEL, installez un seul paquet contenant les informations de débogage pour tous les composants Zabbix:

```
dnf install zabbix-debuginfo
```

- Sur SUSE, installez les paquets debuginfo spécifiques à chaque composant:

```
zypper install zabbix-<component>-debuginfo
```

Activation des dépôts des versions instables

Les instructions ci-dessous servent à activer les dépôts de versions Zabbix instables (désactivés par défaut), utilisés pour les versions candidates des versions mineures de Zabbix.

Commencez par installer ou mettre à jour le paquet zabbix-release vers la dernière version. Pour activer les paquets rc sur votre système, procédez comme suit :

Red Hat Enterprise Linux

Ouvrez le fichier `/etc/yum.repos.d/zabbix.repo` et définissez enabled=1 pour le dépôt `zabbix-unstable`.

```
[zabbix-unstable]
name=Zabbix Official Repository (unstable) - $basearch
baseurl=https://repo.zabbix.com/zabbix/8.0/unstable/rhel/8/$basearch/
enabled=1
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-ZABBIX-A14FE591
```

Debian/Ubuntu

Ouvrez le fichier `/etc/apt/sources.list.d/zabbix.list` et décommentez « Zabbix unstable repository ».

```
#### Zabbix unstable repository
deb https://repo.zabbix.com/zabbix/8.0/unstable/debian bullseye main
deb-src https://repo.zabbix.com/zabbix/8.0/unstable/debian bullseye main
```

SUSE

Ouvrez le fichier `/etc/zypp/repos.d/zabbix.repo` et définissez enable=1 pour le dépôt `zabbix-unstable`.

```
[zabbix-unstable]
name=Zabbix Official Repository
type=rpm-md
baseurl=https://repo.zabbix.com/zabbix/8.0/unstable/sles/15/x86_64/
gpgcheck=1
gpgkey=https://repo.zabbix.com/zabbix/8.0/unstable/sles/15/x86_64/repodata/repomd.xml.key
enabled=1
update=1
```

Installation de l'agent Windows à partir d'un MSI

Aperçu

L'agent Zabbix peut être installé sur Windows à l'aide de packages d'installation MSI 32 bits ou 64 bits, disponibles au téléchargement.

La configuration minimale requise du système d'exploitation pour l'installation MSI est la suivante :

- **Pour l'agent Zabbix :** Windows XP (64 bits) ou Windows Server 2003
- **Pour l'agent Zabbix 2 :** Windows 10 (32 bits) ou Windows Server 2016

Les packages 32 bits ne peuvent pas être installés sur des systèmes 64 bits.

Les packages incluent :

- la prise en charge de TLS (la configuration TLS est facultative)
- les utilitaires Zabbix get et Zabbix sender (peuvent être installés avec l'agent Zabbix/l'agent 2 ou séparément)

Attention:
Les packages de l'agent Zabbix 2 n'incluent pas les plugins chargeables (MongoDB, PostgreSQL, MSSQL), qui doivent être téléchargés et installés séparément.

L'installation peut être effectuée à l'aide de l'assistant d'installation ou de la ligne de commande.

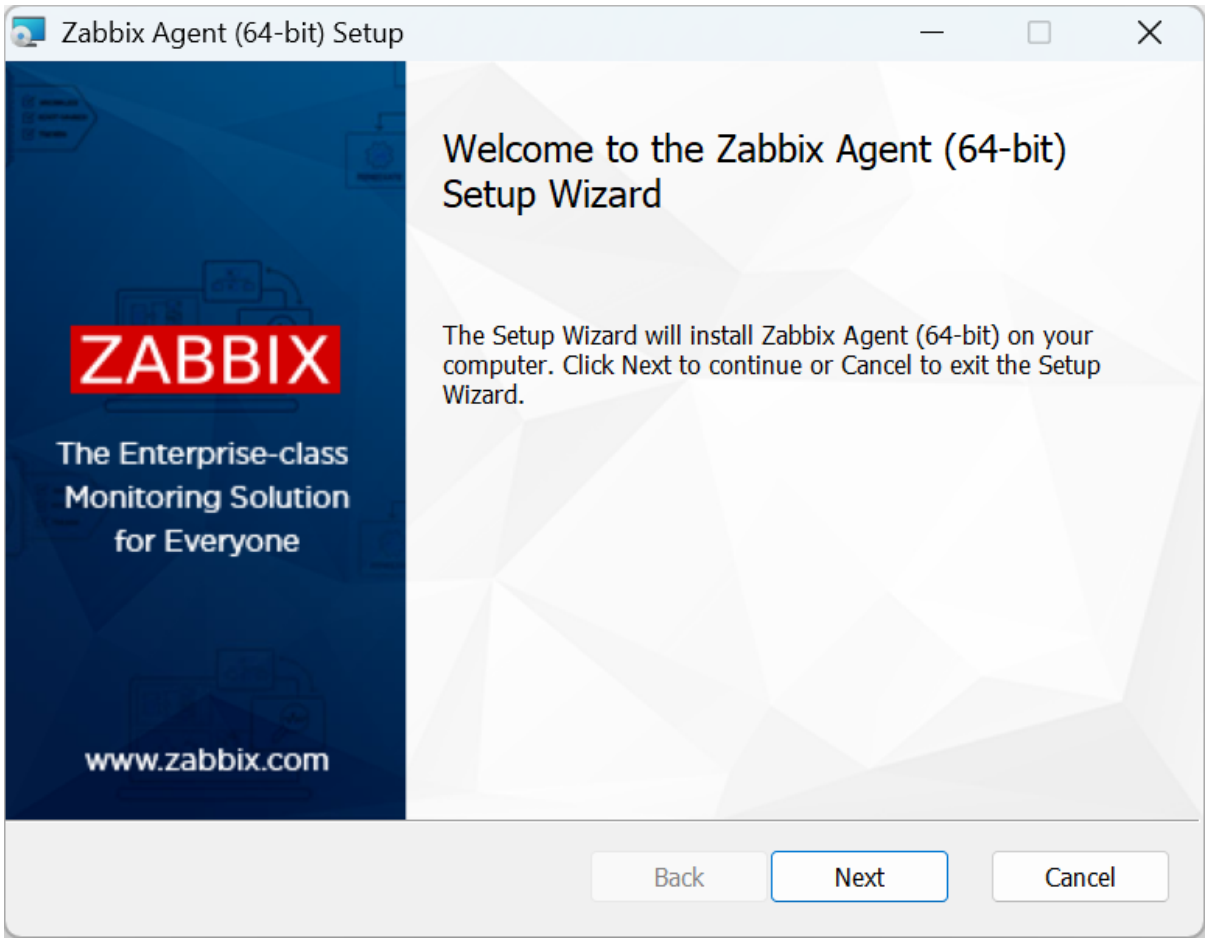
Bien que l'installation à l'aide de packages MSI soit entièrement prise en charge, il est recommandé d'installer au minimum Microsoft .NET Framework 2 pour une gestion correcte des erreurs.

Attention:
Il est recommandé d'utiliser l'emplacement d'installation par défaut fourni par le programme d'installation. L'utilisation d'un emplacement personnalisé sans les autorisations nécessaires peut compromettre la sécurité de l'installation.

Installation avec l'assistant d'installation

Les étapes d'installation suivantes s'appliquent à Zabbix agent et à Zabbix agent 2.

1. Double-cliquez sur le fichier MSI téléchargé pour démarrer l'installation :

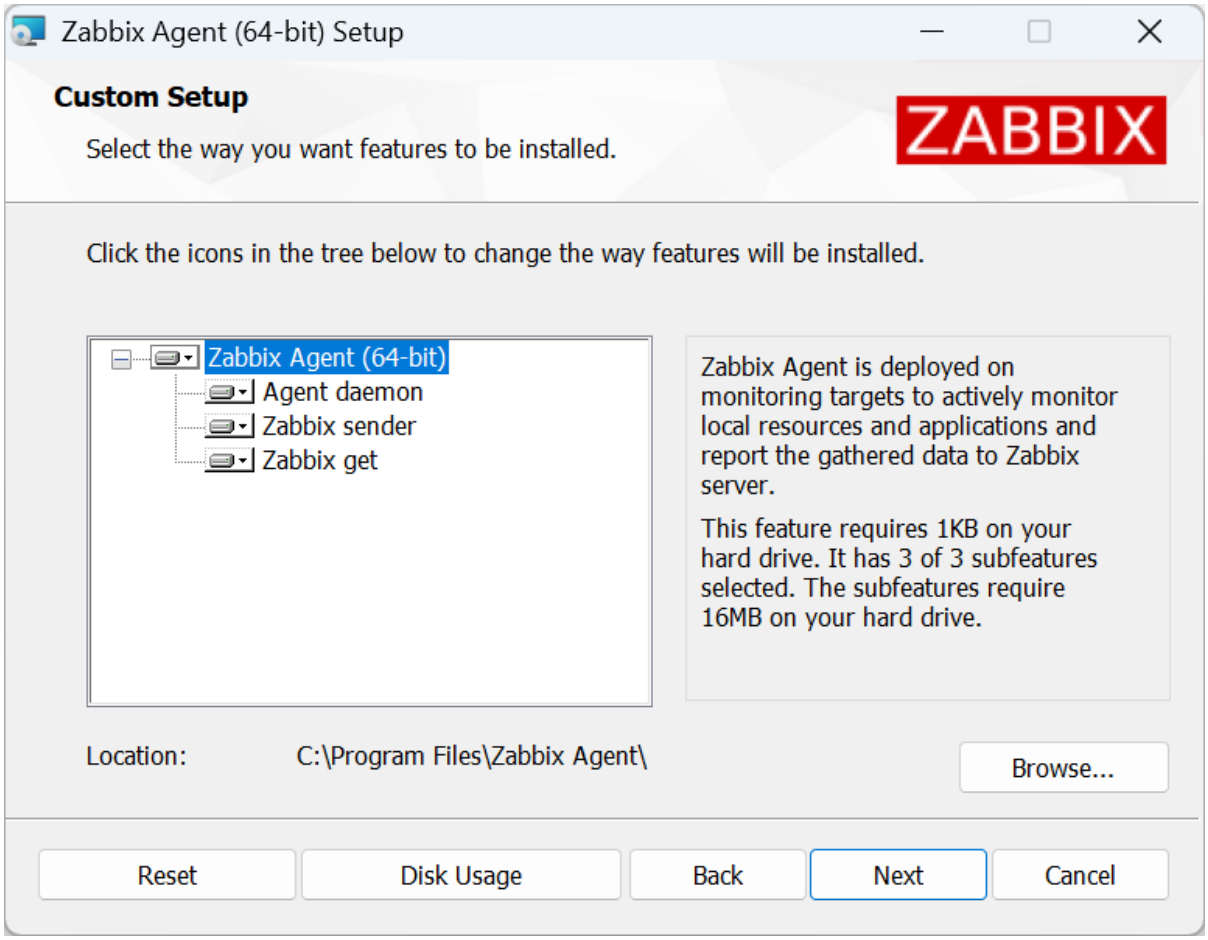


2. Acceptez le contrat de licence utilisateur final :



3. Sélectionnez les composants Zabbix (daemon agent, Zabbix sender, Zabbix get) à installer :

Attention:
Il est recommandé d'utiliser l'emplacement d'installation par défaut fourni par le programme d'installation. L'utilisation d'un emplacement personnalisé sans les autorisations nécessaires peut compromettre la sécurité de l'installation.

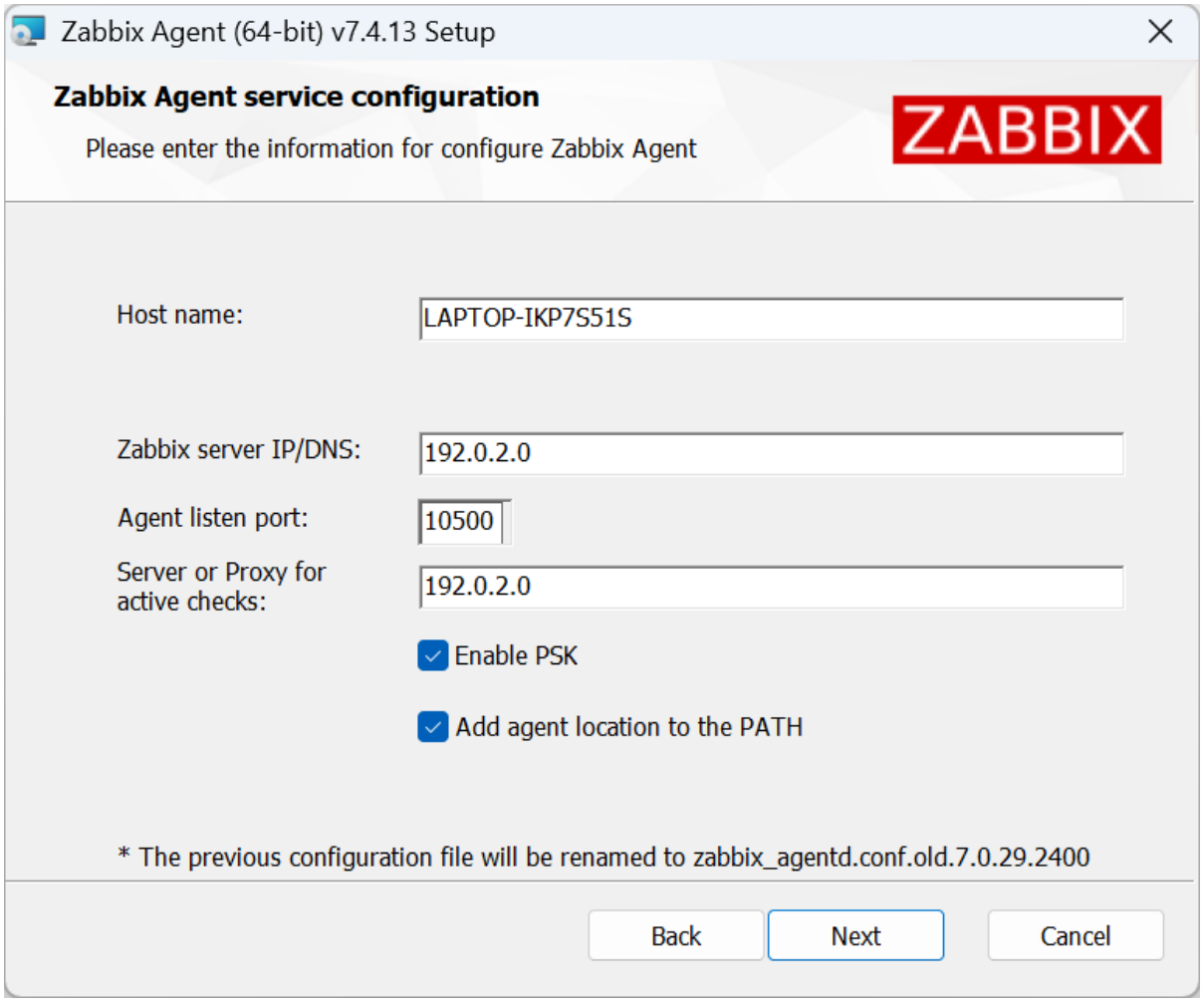


Note:
Si un emplacement personnalisé situé en dehors du répertoire `Program Files` est utilisé et que le dossier de destination existe déjà, le programme d'installation ne modifiera pas ses autorisations. Lors d'une nouvelle installation, vous serez invité à définir manuellement les autorisations appropriées ou à continuer sans les modifier.

4. Configurez les paramètres suivants. Leurs valeurs seront définies dans le fichier de configuration de l'agent Zabbix :

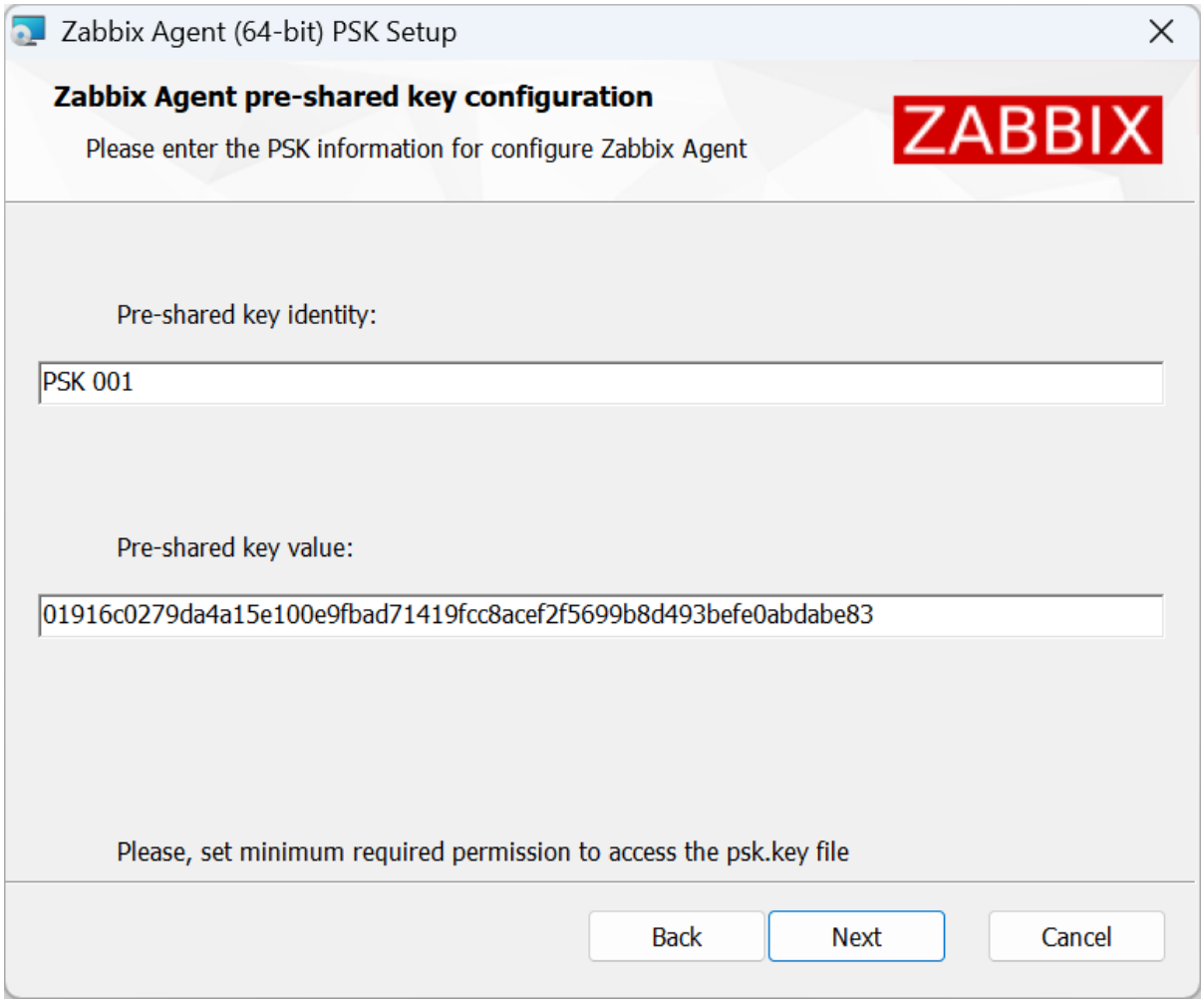
Paramètre	Description
Nom d'hôte	Nom d'hôte de la machine sur laquelle Zabbix agent est installé. Définit le paramètre Hostname.
IP/DNS du serveur Zabbix	Liste d'adresses IP séparées par des virgules, éventuellement au format CIDR, ou de noms DNS des serveurs Zabbix ou des proxy Zabbix. Ce paramètre est **obligatoire**. Définit le paramètre Server.
Port d'écoute de l'agent	L'agent écoutera sur ce port les connexions provenant du serveur. Définit le paramètre ListenPort.
Serveur ou proxy pour les vérifications actives	Adresse du serveur/proxy Zabbix ou configuration de cluster à partir de laquelle obtenir les vérifications actives. L'adresse du serveur/proxy est une adresse IP ou un nom DNS, avec un port facultatif séparé par deux-points. Définit le paramètre ServerActive.
Activer PSK	Cochez la case pour activer la prise en charge de TLS à l'aide de clés prépartagées. Définit les paramètres TLSConnect et TLSAccept sur `psk`.
Ajouter l'emplacement de l'agent au PATH	Cochez la case pour ajouter l'emplacement de Zabbix agent à la variable système PATH.

Note:
Si un Zabbix agent existant est détecté, les valeurs des paramètres de son fichier de configuration seront affichées. De plus, le fichier de configuration existant sera renommé pendant l'installation et un nouveau fichier de configuration sera créé.



5. Configurez les paramètres PSK si vous avez coché la case *Activer PSK* à l'étape précédente. Ces paramètres seront également définis dans le fichier de configuration de Zabbix agent :

Paramètre	Description
Identité de la clé prépartagée	Chaîne d'identité de la clé prépartagée. Définit le paramètre TLSPSKIdentity.
Valeur de la clé prépartagée	Valeur de la chaîne de la clé prépartagée. Crée le fichier `psk.key` contenant la clé et définit le paramètre TLSPSKFile sur l'emplacement de la clé (par défaut : `C:\Program Files\Zabbix Agent\psk.key`). Il est **recommandé** de restreindre l'accès au fichier de clé prépartagée en ajustant ses paramètres de sécurité afin que seul Zabbix agent (ou l'utilisateur exécutant l'agent) puisse le lire.

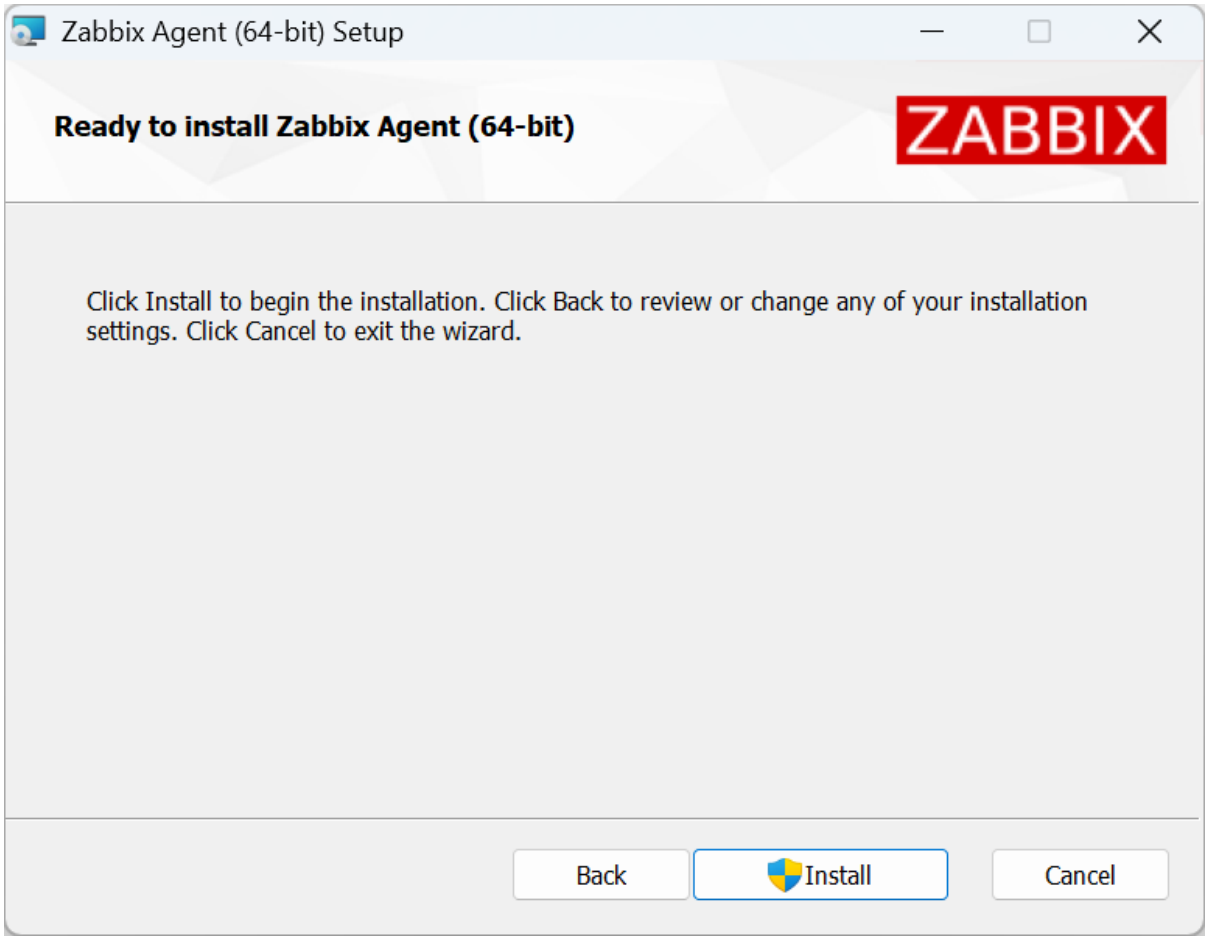


6. Cliquez sur *Installer* pour commencer l'installation.

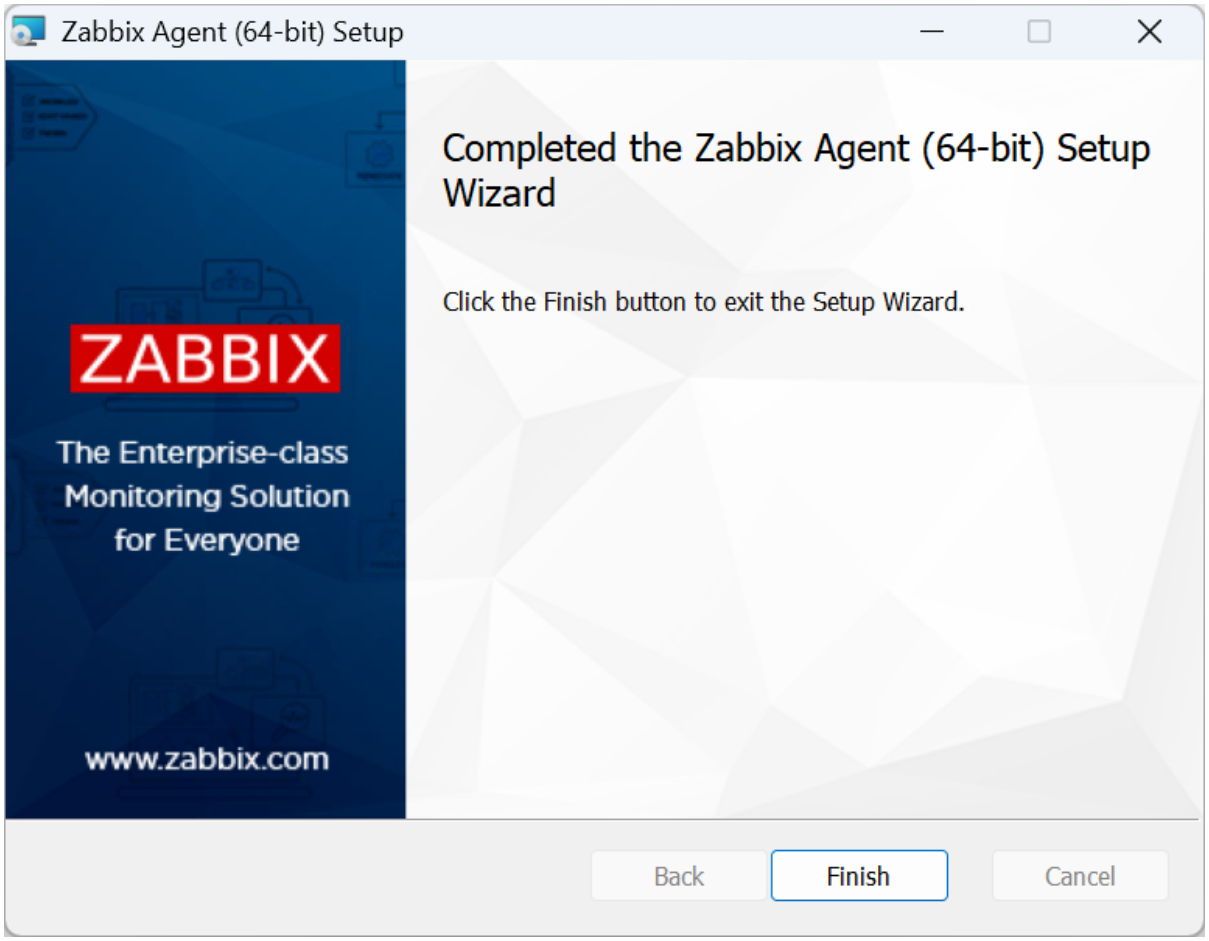
Tous les composants Zabbix sélectionnés et le fichier de configuration de Zabbix agent seront installés à l'emplacement spécifié (par défaut : `C:\Program Files\Zabbix Agent`). Il en va de même pour Zabbix agent 2, à ceci près que des fichiers de configuration supplémentaires pour ses plugins intégrés seront installés dans le sous-dossier `zabbix_agent2.d\plugins.d`.

De plus, `zabbix_agentd.exe` (ou `zabbix_agent2.exe`) sera configuré comme un service Windows avec un démarrage automatique différé (ou un démarrage automatique sur les versions de Windows antérieures à Windows Vista/Server 2008).

Si une autre version de Zabbix agent est en cours d'exécution pendant l'installation, vous serez invité à choisir entre fermer l'application et tenter de la redémarrer, ou la laisser ouverte, auquel cas un redémarrage sera nécessaire.



7. Cliquez sur le bouton *Terminer* pour quitter l'assistant d'installation.



Installation depuis la ligne de commande

L'agent Zabbix peut être installé depuis la ligne de commande en exécutant le programme d'installation MSI avec `msiexec`. Par exemple :

```
msiexec.exe /l*v "C:\package.log" /i "C:\zabbix_agent-8.0.0-windows-amd64-openssl.msi" /qn+ SERVER=192.0.2
```

Cette méthode permet des installations sans intervention et des configurations personnalisées à l'aide de paramètres.

Paramètres pris en charge

Les packages d'installation MSI de Zabbix agent prennent en charge les paramètres suivants pour Zabbix agent et Zabbix agent 2.

Note:
Les paramètres de Zabbix agent/agent2 sont définis dans le fichier de configuration pendant l'installation. Cliquez sur le nom d'un paramètre pour afficher sa description détaillée et des exemples de configuration sur la page Zabbix agent (Windows). Pour Zabbix agent 2, consultez la page Zabbix agent 2 (Windows).

Parameter	Description
ADDDEFAULT	Une liste de composants séparés par des virgules à installer dans leur configuration par défaut. Pour plus d'informations, consultez la propriété ADDDEFAULT. Valeurs possibles: `AgentProgram`, `GetProgram`, `SenderProgram`, `ALL` Exemple: `ADDDEFAULT=AgentProgram,GetProgram`
ADDLOCAL	Une liste de composants séparés par des virgules à installer localement. Pour plus d'informations, consultez la propriété ADDLOCAL. Valeurs possibles: `AgentProgram`, `GetProgram`, `SenderProgram`, `ALL` Exemple: `ADDLOCAL=AgentProgram,SenderProgram`
ALLOWDENYKEY	Une liste de paramètres AllowKey ou DenyKey séparés par des points-virgules pour restreindre les vérifications de Zabbix agent. Si nécessaire, utilisez une barre oblique inverse pour échapper le délimiteur (`\;`). Définit les paramètres AllowKey et DenyKey dans le fichier de configuration de l'agent. Exemple: `ALLOWDENYKEY="AllowKey=system.run[more C:\Windows\System32\drivers\etc\hosts\\; echo 'File read complete'];DenyKey=system.run[*]"`
CONF	Le chemin complet vers un fichier de configuration de modèle pour Zabbix agent. Pendant l'installation, ce fichier deviendra le fichier de configuration de l'agent. Le fichier doit contenir au moins les paramètres Server et LogFile. Exemple: `CONF="C:\full\path\to\example.conf"`
DONOTSTART	Utilisez `DONOTSTART=1` pour empêcher le programme d'installation MSI de démarrer le service Zabbix agent.
ENABLEPATH	Utilisez `ENABLEPATH=1` pour ajouter l'emplacement de Zabbix agent à la variable système PATH.
ENABLEPERSISTENTBUFFER	Zabbix agent 2 uniquement. Active l'utilisation du stockage persistant local pour les éléments actifs.
HOSTINTERFACE	Paramètre facultatif qui définit l'interface de l'hôte.
HOSTMETADATA	Paramètre facultatif qui définit les métadonnées de l'hôte.
HOSTMETADATAITEM	Paramètre facultatif qui définit un élément utilisé pour obtenir les métadonnées de l'hôte.
HOSTNAME	Paramètre facultatif qui définit le nom d'hôte.
HOSTNAMEITEM	Paramètre facultatif qui définit un élément utilisé pour obtenir le nom d'hôte.
INCLUDE	Une liste de fichiers individuels séparés par des points-virgules ou de tous les fichiers d'un répertoire à inclure dans le fichier de configuration de Zabbix agent.
IGNOREALREADYEXISTSFOLDER	Pour les nouvelles installations, utilisez `IGNOREALREADYEXISTSFOLDER=1` pour autoriser l'installation dans un dossier de destination existant en dehors du répertoire `Program Files`. Si ce paramètre n'est pas spécifié, l'installation échoue lorsque ce dossier de destination existe déjà.
INSTALLFOLDER	Le chemin complet vers un dossier dans lequel les composants Zabbix et le fichier de configuration de Zabbix agent seront installés. Pour Zabbix agent 2, des fichiers de configuration supplémentaires pour les plugins intégrés seront installés dans le sous-dossier `zabbix_agent2.d\plugins.d`. Exemple: `INSTALLFOLDER="C:\Program Files\Zabbix Agent"`
LISTENIP	Une liste d'adresses IP séparées par des virgules sur lesquelles l'agent doit écouter.
LISTENPORT	L'agent écoutera sur ce port pour les connexions provenant du serveur.
LOGFILE	Le nom du fichier journal de Zabbix agent.
LOGTYPE	Le type de sortie du journal.

Parameter	Description
NONMSICONFNAME	Le chemin complet vers un fichier de configuration personnalisé pour Zabbix agent. Pendant l'installation, tous les paramètres de configuration valides de l'agent présents dans ce fichier (limités à ceux répertoriés dans ce tableau) seront écrits dans le nouveau fichier de configuration de l'agent créé. Le fichier doit contenir au moins le paramètre Server. Exemple: `NONMSICONFNAME="C:\full\path\to\example.conf"`
PERSISTENTBUFFERFILE	Zabbix agent 2 uniquement. Le fichier dans lequel Zabbix agent 2 doit conserver la base de données SQLite.
PERSISTENTBUFFERPERIOD	Zabbix agent 2 uniquement. La période pendant laquelle les données doivent être stockées lorsqu'il n'y a pas de connexion au serveur ou au proxy.
SERVER	Une liste d'adresses IP séparées par des virgules, éventuellement en notation CIDR, ou de noms DNS de serveurs Zabbix ou de proxies Zabbix. Ce paramètre est **obligatoire**, sauf lorsque STARTAGENTS est défini sur 0.
SERVERACTIVE	L'adresse du serveur/proxy Zabbix ou la configuration de cluster à partir de laquelle obtenir les vérifications actives.
SKIP	Utilisez `SKIP=fw` pour empêcher le programme d'installation MSI d'ajouter une règle d'exception au Pare-feu Windows pour Zabbix agent.
SOURCEIP	L'adresse IP source pour les connexions sortantes vers le serveur Zabbix ou le proxy Zabbix, ou pour établir des connexions lors de l'exécution de certains éléments (web.page.get, net.tcp.port, etc.).
STARTAGENTS	Le nombre d'instances pré-forkées de zabbix_agentd qui traitent les vérifications passives. Si la valeur est définie sur 0, les vérifications passives sont désactivées et l'agent n'écoutera sur aucun port TCP.
STARTUPTYPE	Type de démarrage du service Zabbix agent. Valeurs possibles: **automatic** - démarre le service automatiquement au démarrage de Windows; **delayed** - *(par défaut)* retarde le démarrage du service après la fin du démarrage des services lancés automatiquement (disponible sur Windows Vista/Server 2008 et versions ultérieures); **manual** - démarre le service manuellement (par un utilisateur ou une application); **disabled** - désactive le service afin qu'il ne puisse pas être démarré par un utilisateur ou une application. Exemple: `STARTUPTYPE=disabled`
STATUSPORT	Zabbix agent 2 uniquement. Si défini, l'agent écoutera sur ce port les requêtes d'état HTTP (http://localhost:<port>/status).
TIMEOUT	Spécifie le délai d'attente (en secondes) pour établir la connexion et échanger des données avec le proxy ou le serveur Zabbix.
TLSACCEPT	Les connexions entrantes à accepter (utilisé pour les vérifications passives). Si défini sur `psk`, alors TLSCONNECT sera également défini sur `psk` (sauf indication contraire).
TLSCAFILE	Le chemin complet vers un fichier contenant les certificats des AC de niveau supérieur pour la vérification du certificat du pair.
TLSCERTFILE	Le chemin complet vers un fichier contenant le certificat de l'agent ou la chaîne de certificats.
TLSCONNECT	La manière dont l'agent doit se connecter au serveur ou au proxy Zabbix (utilisé pour les vérifications actives). Si défini sur `psk`, alors TLSACCEPT sera également défini sur `psk` (sauf indication contraire).
TLSCRLFILE	Le chemin complet vers un fichier contenant les certificats révoqués.
TLSKEYFILE	Le chemin complet vers un fichier contenant la clé privée de Zabbix agent.
TLSPSKFILE	Le chemin complet vers un fichier contenant la clé prépartagée de Zabbix agent. Si TLSPSKFILE et TLSPSKVALUE sont tous deux définis, la valeur de TLSPSKVALUE sera écrite dans le fichier spécifié dans TLSPSKFILE. Il est **recommandé** de restreindre l'accès au fichier de clé prépartagée en ajustant les paramètres de sécurité du fichier afin que seul Zabbix agent (ou l'utilisateur exécutant l'agent) puisse le lire.
TLSPSKIDENTITY	La chaîne d'identité de la clé prépartagée.
TLSPSKVALUE	La valeur de la clé prépartagée. Si TLSPSKFILE et TLSPSKVALUE sont tous deux définis, la valeur de TLSPSKVALUE sera écrite dans le fichier spécifié dans TLSPSKFILE. Exemple: `TLSPSKVALUE=1f87b595725ac58dd977beef14b97461a7c1045b9a1c963065002c5473194952`
TLSSERVERCERTISSUER	L'émetteur du certificat du serveur (proxy) autorisé.
TLSSERVERCERTSUBJECT	Le sujet du certificat du serveur (proxy) autorisé.
UNSAFEUSERPARAMETERS	Autorise le passage de tous les caractères dans les arguments des paramètres définis par l'utilisateur.

Exemples

L'exemple suivant installe Zabbix agent avec une configuration personnalisée. Il active également la prise en charge TLS à l'aide de clés prépartagées.

```
mkdir "C:\Program Files\Zabbix Agent" 2>nul
msiexec.exe /l*v "C:\package.log" /i "C:\zabbix_agent-8.0.0-windows-amd64-openssl.msi" /qn+^
 SERVER=192.0.2.1^
 INSTALLFOLDER="C:\Program Files\Zabbix Agent"^
 HOSTNAME=LAPTOP-IKP7S51S^
 TLSACCEPT=psk^
 TLSCONNECT=psk^
 TLSPSKIDENTITY="PSK 001"^
 TLSPSKFILE="C:\Program Files\Zabbix Agent\psk.key"^
 TLSPSKVALUE=1f87b595725ac58dd977beef14b97461a7c1045b9a1c963065002c5473194952^
 ENABLEPATH=1^
 ALLOWDENYKEY="AllowKey=system.run[type C:\Windows\System32\drivers\etc\hosts];DenyKey=system.run[*]"
```

L'exemple suivant installe une version plus récente de Zabbix agent et utilise un **fichier de configuration de modèle** (`CONF="C:\agent-template.conf"`). Pendant l'installation, ce fichier deviendra le fichier de configuration de l'agent. Pour hériter des paramètres de l'ancien fichier de configuration, utilisez des espaces réservés de paramètres (par exemple, `[AllowDenyKey]`).

```
msiexec.exe /l*v "C:\package.log" /i "C:\zabbix_agent-8.0.1-windows-amd64-openssl.msi" /qn+ NONMSICONFNAME

#### exemple de agent-template.conf :
LogFile=[LogFile]
[AllowDenyKey]
Server=192.0.2.8
Hostname=DESKTOP-X9F4A2B
[Include]
[TLSConnect]
[TLSAccept]
[TLSPSKIdentity]
[TLSPSKFile]
```

Vous pouvez également utiliser un **fichier de configuration personnalisé** (`NONMSICONFNAME="C:\agent-custom.conf"`). Pendant l'installation, tous les paramètres de configuration valides de l'agent présents dans ce fichier (limités à ceux répertoriés dans le tableau ci-dessus) seront écrits dans le nouveau fichier de configuration de l'agent. Pour conserver la configuration existante de l'agent, définissez les paramètres à préserver.

```
msiexec.exe /l*v "C:\package.log" /i "C:\zabbix_agent-8.0.1-windows-amd64-openssl.msi" /qn+ NONMSICONFNAME

#### exemple de agent-custom.conf :
Server=192.0.2.8
Hostname=DESKTOP-X9F4A2B
```

Plugins chargeables de Zabbix agent 2

Les plugins chargeables de Zabbix agent 2 peuvent être installés sous Windows à l'aide de packages d'installation MSI 64 bits, disponibles au téléchargement.

La configuration minimale requise du système d'exploitation pour l'installation MSI est Windows 10 (64 bits) ou Windows Server 2016.

> **Attention:**
> Avant d'installer un plugin, veuillez consulter son fichier README. Il peut contenir des exigences spécifiques et des instructions d'installation.

Comme pour Zabbix agent/agent2, les plugins chargeables peuvent être installés à l'aide de l'assistant d'installation ou de la ligne de commande.

Installation depuis l'assistant d'installation

1. Double-cliquez sur le fichier MSI téléchargé pour démarrer l'installation.
2. Acceptez le contrat de licence utilisateur final.
3. Sélectionnez les plugins chargeables de Zabbix agent 2 à installer.

Attention:
Il est recommandé d'utiliser l'emplacement d'installation par défaut fourni par le programme d'installation. L'utilisation d'un emplacement personnalisé sans les autorisations nécessaires peut compromettre la sécurité de l'installation.

4. Cliquez sur *Install* pour lancer l'installation.

Tous les plugins chargeables sélectionnés de Zabbix agent 2 seront installés à l'emplacement spécifié (par défaut : `C:\Program Files\Zabbix Agent 2`), avec leurs fichiers de configuration installés dans le sous-dossier `zabbix_agent2.d`.

5. Cliquez sur le bouton *Finish* pour quitter l'assistant d'installation.

Installation depuis la ligne de commande

Les plugins chargeables de Zabbix agent 2 peuvent être installés depuis la ligne de commande en exécutant le programme d'installation MSI avec `msiexec`. Par exemple :

```
msiexec.exe /l*v "C:\package.log" /i "C:\zabbix_agent2_plugins-8.0.0-windows-amd64.msi" /qn+
```

Les packages d'installation MSI des plugins chargeables de Zabbix agent 2 prennent en charge les paramètres suivants.

Parameter	Description
ADDDEFAULT	Une liste de composants séparés par des virgules à installer dans leur configuration par défaut. Pour plus d'informations, consultez la propriété ADDDEFAULT. Valeurs possibles : `ALL`, `CephPlugin`, `EmberplusPlugin`, `MongodbPlugin`, `MssqlPlugin`, `NvidiagpuPlugin`, `PostgresqlPlugin` Exemple : `ADDDEFAULT=MongodbPlugin,PostgresqlPlugin`
ADDLOCAL	Une liste de composants séparés par des virgules à installer localement. Pour plus d'informations, consultez la propriété ADDLOCAL. Valeurs possibles : `ALL`, `CephPlugin`, `EmberplusPlugin`, `MongodbPlugin`, `MssqlPlugin`, `NvidiagpuPlugin`, `PostgresqlPlugin` Exemple : `ADDLOCAL=MongodbPlugin,MssqlPlugin`
INSTALLFOLDER	Le chemin complet vers un dossier où les composants Zabbix seront installés, avec leurs fichiers de configuration installés dans le sous-dossier `zabbix_agent2.d`. Exemple : `INSTALLFOLDER="C:\Program Files\Zabbix Agent 2"`

Compilation de l'agent Zabbix 2 sous Windows

Présentation

Cette page explique comment compiler l'agent Zabbix 2 à partir des sources sous Windows 10 (64 bits ou 32 bits).

Les versions 32 bits et 64 bits peuvent être compilées sur une plateforme 64 bits, mais seule la version 32 bits peut être compilée sur une plateforme 32 bits.

La compilation de l'agent Zabbix 2 nécessite :

- Les outils de compilation MinGW
- Le langage de programmation Go
- OpenSSL (pour les fonctionnalités de chiffrement dans Zabbix)
- PCRE2 (expressions régulières compatibles Perl ; pour les fonctionnalités de recherche par motifs d'expressions régulières dans Zabbix)

Vous pouvez compiler l'agent Zabbix 2 à l'aide de l'une des méthodes suivantes :

- Utilisation de vcpkg - une approche automatisée qui simplifie la gestion des dépendances à l'aide d'un gestionnaire de paquets C++.
- Compilation manuelle - une approche manuelle qui nécessite l'installation de toutes les dépendances avant la compilation de l'agent.

Attention:
Avant de commencer le processus de compilation, veuillez tenir compte des points suivants :

- Pour exécuter les commandes, utilisez l'invite de commandes, lancée par un utilisateur disposant des autorisations suffisantes pour écrire dans les dossiers protégés. Cependant, lors de l'installation d'OpenSSL et de PCRE2, utilisez le terminal MSYS2.
- Il est recommandé de créer un répertoire de travail à l'emplacement `C:\Zabbix` pour tous les fichiers sources et les dossiers de compilation. Cependant, les composants compilés doivent être installés dans `C:\Zabbix\x64` (ou `C:\Zabbix\x86` pour les compilations 32 bits).

Compilation de Zabbix agent 2 avec vcpkg

Cette section contient les instructions pour compiler Zabbix agent avec vcpkg, un gestionnaire de paquets qui simplifie la gestion des dépendances et l'intégration aux projets C++.

1. Téléchargez et installez Build Tools for Visual Studio 2022. Lors de l'installation, veillez à sélectionner la charge de travail *Desktop development with C++*, qui inclut le gestionnaire de paquets vcpkg.

2. Téléchargez et installez Go (disponible sous forme d'installateur MSI). Lors de l'installation, veillez à spécifier `C:\Zabbix\Go` comme dossier d'installation.

3. Téléchargez la distribution MinGW qui utilise la bibliothèque d'exécution Microsoft Visual C ; par exemple :

- Pour les compilations 64 bits : `x86_64-15.1.0-release-win32-seh-msvcrt-rt_v12-rev0.7z`
- Pour les compilations 32 bits : `i686-15.1.0-release-win32-dwarf-msvcrt-rt_v12-rev0.7z`

Attention:
La version de MinGW doit respecter les exigences minimales.

Extrayez-le ensuite dans `C:\Zabbix\mingw64` (ou `C:\Zabbix\mingw32` pour les compilations 32 bits).

4. Initialisez vcpkg et installez les dépendances requises pour compiler Zabbix agent 2 (notez que cette opération peut prendre un certain temps) :

```
cd C:\Zabbix

set PATH=%PATH%;"C:\Program Files (x86)\Microsoft Visual Studio\2022\BuildTools\VC\vcpkg"
vcpkg new --application
vcpkg add port pcre2
vcpkg add port libiconv
vcpkg add port openssl

#### Pour les compilations 64 bits :
set PATH=C:\Zabbix\mingw64\bin;%PATH%
vcpkg install --triplet x64-mingw-static --x-install-root=x64

#### Pour les compilations 32 bits :
set PATH=C:\Zabbix\mingw32\bin;%PATH%
vcpkg install --triplet x86-mingw-static --x-install-root=x86
```

5. Téléchargez l'archive des sources de Zabbix et extrayez-la dans `C:\Zabbix\zabbix-8.0.0`.

6. Accédez au répertoire de compilation de Zabbix (`C:\Zabbix\zabbix-8.0.0\build\mingw`) et créez le script `build.bat` suivant :

- Pour les compilations 64 bits :

```
:: Ajouter MinGW et Go à la variable système `PATH` pour la session actuelle :
set PATH=C:\Zabbix\mingw64\bin;%PATH%
set PATH=C:\Zabbix\Go\bin;%PATH%

:: Définir le chemin d'installation de vcpkg :
set vcpkg="C:\Zabbix\x64\x64-mingw-static"

:: Définir les indicateurs de l'éditeur de liens pour la bibliothèque Crypt32 :
SET CGO_LDFLAGS="-lCrypt32"

:: Exécuter le processus de compilation :
```

```
mingw32-make GOFLAGS="-buildvcs=false" ARCH=AMD64 ^
    PCRE2="%vcpkg%" ^
    OPENSSL="%vcpkg%" ^
    all
```

- Pour les compilations 32 bits :

```
:: Ajouter MinGW et Go à la variable système `PATH` pour la session actuelle :
set PATH=C:\Zabbix\mingw32\bin;%PATH%
set PATH=C:\Zabbix\Go\bin;%PATH%

:: Définir le chemin d'installation de vcpkg :
set vcpkg="C:\Zabbix\x86\x86-mingw-static"

:: Définir les indicateurs de l'éditeur de liens pour la bibliothèque Crypt32 :
SET CGO_LDFLAGS="-lCrypt32"

:: Exécuter le processus de compilation :
mingw32-make GOFLAGS="-buildvcs=false" ARCH=x86 ^
    PCRE2="%vcpkg%" ^
    OPENSSL="%vcpkg%" ^
    all
```

7. Compilez Zabbix agent 2 en exécutant le script :

```
build.bat
```

Après la compilation, le binaire de Zabbix agent 2 se trouvera dans `C:\Zabbix\zabbix-8.0.0\bin\win64` (pour les compilations 64 bits) ou dans `C:\Zabbix\zabbix-8.0.0\bin\win32` (pour les compilations 32 bits). Les fichiers de configuration de Zabbix agent 2 se trouvent dans `C:\Zabbix\zabbix-8.0.0\src\go\conf`.

Pour exécuter l'agent, copiez `zabbix_agent2.exe` et ses fichiers de configuration dans un dossier dédié (par exemple, `C:\Zabbix\agent2`), puis exécutez l'agent :

```
mkdir C:\Zabbix\agent2

#### Pour les compilations 64 bits :
copy C:\Zabbix\zabbix-8.0.0\bin\win64\zabbix_agent2.exe C:\Zabbix\agent2\

#### Pour les compilations 32 bits :
copy C:\Zabbix\zabbix-8.0.0\bin\win32\zabbix_agent2.exe C:\Zabbix\agent2\

copy C:\Zabbix\zabbix-8.0.0\src\go\conf\zabbix_agent2.conf C:\Zabbix\agent2\
xcopy /E /I C:\Zabbix\zabbix-8.0.0\src\go\conf\zabbix_agent2.d C:\Zabbix\agent2\zabbix_agent2.d\

C:\Zabbix\agent2\zabbix_agent2.exe -c C:\Zabbix\agent2\zabbix_agent2.conf
```

Si nécessaire, poursuivez avec la compilation des plugins chargeables de Zabbix agent 2.

Compilation des plugins chargeables de Zabbix agent 2

1. Téléchargez le code source du plugin Zabbix correspondant à votre version de Zabbix agent 2 (par exemple, `zabbix-agent2-plugin-pos`
et extrayez-le dans `C:\Zabbix`.

> **Attention:**
> Avant d'installer un plugin, veuillez consulter son fichier README. Il peut contenir des exigences spécifiques et des instructions d'installation.

2. Accédez au répertoire du plugin extrait et compilez le plugin :

```
cd C:\Zabbix\zabbix-agent2-plugin-ember-plus-8.0.0

#### Pour les compilations 64 bits :
mingw32-make ARCH=AMD64

#### Pour les compilations 32 bits :
mingw32-make ARCH=x86
```

Après la compilation, le binaire du plugin Zabbix agent 2 et son fichier de configuration se trouveront dans le même répertoire du plugin.

L'exécutable du plugin peut être placé n'importe où, à condition qu'il puisse être chargé par Zabbix agent 2. Spécifiez le chemin vers le binaire du plugin dans le fichier de configuration du plugin, par exemple dans postgresql.conf pour le plugin PostgreSQL :

```
Plugins.PostgreSQL.System.Path=/path/to/executable/zabbix-agent2-plugin-postgresql
```

Le chemin vers le fichier de configuration du plugin doit être spécifié dans le paramètre Include du fichier de configuration de Zabbix agent 2 :

```
Include=/path/to/plugin/configuration/file/postgresql.conf
```

Passez à la configuration pour plus de détails sur la configuration des plugins.

Compilation manuelle de Zabbix agent 2

Attention:
Cette méthode de compilation de Zabbix agent 2 convient aux utilisateurs qui ont besoin d'un contrôle total sur l'environnement de compilation ou qui se trouvent dans un environnement restreint où l'utilisation de vcpkg n'est pas possible.

Cette section contient des instructions pour compiler manuellement Zabbix agent 2, notamment l'installation des outils de compilation et des dépendances requis, puis la compilation de l'agent.

Configuration des outils de compilation

1. Téléchargez et installez MSYS2 (disponible sous forme d'installateur MSI). Pendant l'installation, veillez à spécifier `C:\Zabbix\msys64` comme dossier d'installation.

2. Téléchargez et installez Go (disponible sous forme d'installateur MSI ; voir les versions de Go actuellement prises en charge). Pendant l'installation, veillez à spécifier `C:\Zabbix\Go` comme dossier d'installation.

3. Téléchargez la distribution MinGW qui utilise la bibliothèque d'exécution Microsoft Visual C ; par exemple :

- Pour les compilations 64 bits : `x86_64-15.1.0-release-win32-seh-msvcrt-rt_v12-rev0.7z`
- Pour les compilations 32 bits : `i686-15.1.0-release-win32-dwarf-msvcrt-rt_v12-rev0.7z`

Attention:
La version de MinGW doit satisfaire aux exigences minimales.

Ensuite, extrayez-la dans `C:\Zabbix\mingw64` (ou `C:\Zabbix\mingw32` pour les compilations 32 bits).

Installation d'OpenSSL

Note:
Pour compiler l'agent Zabbix sans prise en charge de TLS, passez à la section Installation de PCRE2.

1. Ouvrez le terminal MSYS2 MSYS avec des privilèges d'administrateur et exécutez les commandes suivantes :

```
pacman -S perl-Locale-Maketext-Simple
pacman -S nasm
pacman -S make
pacman -S cmake
```

2. Téléchargez l'archive des sources d'OpenSSL et extrayez-la dans `C:\Zabbix\openssl-3.5.0`.

3. Accédez au répertoire OpenSSL extrait et créez le script `build.sh` suivant :

- Pour les compilations 64 bits :

```
####!/usr/bin/env bash

export PATH="/c/Zabbix/mingw64/bin:$PATH"
export d="/c/Zabbix/x64/OpenSSL-Win64-350-static"

perl Configure mingw64 no-shared no-capieng no-winstore no-srp no-gost no-dgram no-dtls1-method no-dtls1_2

make
make install
```

- Pour les compilations 32 bits :

```
####!/usr/bin/env bash

export PATH="/c/Zabbix/mingw32/bin:$PATH"
export d="/c/Zabbix/x86/OpenSSL-Win64-350-static"

perl Configure mingw no-shared no-capieng no-winstore no-srp no-gost no-dgram no-dtls1-method no-dtls1_2-m

make
make install
```

Attention:
Assurez-vous de révoquer l'accès en écriture des utilisateurs non administrateurs au répertoire `C:\Zabbix\x64\OpenSSL-Win64-350-static`. Sinon, l'agent chargera les paramètres SSL depuis un chemin pouvant être modifié par des utilisateurs non privilégiés, ce qui pourrait entraîner une vulnérabilité de sécurité.

- L'option `no-shared` rend les bibliothèques statiques OpenSSL libcrypto.lib et libssl.lib autonomes, de sorte que les binaires Zabbix incluent OpenSSL sans nécessiter de DLL externes. Cela signifie que les binaires Zabbix peuvent être copiés sur d'autres machines Windows sans les bibliothèques OpenSSL ; toutefois, lorsqu'une nouvelle version corrective d'OpenSSL est publiée, l'agent Zabbix devra être recompilé.
- Sans l'option `no-shared`, Zabbix s'appuie sur les DLL OpenSSL lors de l'exécution. Cela signifie que les mises à jour d'OpenSSL peuvent ne pas nécessiter la recompilation de l'agent Zabbix ; toutefois, lors de sa copie sur d'autres machines, les DLL OpenSSL doivent également être copiées.

Pour plus d'informations sur les autres options de configuration d'OpenSSL, consultez la documentation OpenSSL.

4. Configurez et installez OpenSSL en exécutant le script (notez que cette opération peut prendre un certain temps) :

```
cd /c/Zabbix/openssl-3.5.0
./build.sh
```

Installation de PCRE2

1. Téléchargez l'archive source de PCRE2 et extrayez-la dans `C:\Zabbix\pcre2-10.45`.

2. Ouvrez le terminal MSYS2 MSYS avec les privilèges d'administrateur. Ensuite, créez un répertoire `build` dans le répertoire PCRE2 extrait et accédez-y :

```
mkdir /c/Zabbix/pcre2-10.45/build
cd /c/Zabbix/pcre2-10.45/build
```

3. Configurez PCRE2 :

```
#### For 64-bit builds:
export PATH="/c/Zabbix/mingw64/bin:$PATH"
cmake -DCMAKE_C_COMPILER=gcc -DCMAKE_C_FLAGS="-O2 -g" -DCMAKE_INSTALL_PREFIX="/c/Zabbix/x64/PCRE2" ..

#### For 32-bit builds:
export PATH="/c/Zabbix/mingw32/bin:$PATH"
cmake -DCMAKE_C_COMPILER=gcc -DCMAKE_C_FLAGS="-m32 -O2 -g" -DCMAKE_EXE_LINKER_FLAGS="-Wl,-mi386pe" -DCMAKE
```

Note:
Si des erreurs se produisent, il est recommandé de supprimer le cache CMake avant de tenter de répéter le processus de compilation CMake. Le cache (`CMakeCachecache.txt`) se trouve dans le répertoire build du répertoire PCRE2 extrait.

4. Installez PCRE2 :

```
make install
```

Compilation de Zabbix agent 2

1. Téléchargez l'archive des sources de Zabbix et extrayez-la dans `C:\Zabbix\zabbix-8.0.0`.

Si vous devez générer une archive des sources à partir du dépôt de sources brutes (par exemple, pour appliquer des correctifs personnalisés ou compiler à partir du code source le plus récent), exécutez les commandes suivantes sur une machine **Linux** sur laquelle Go est installé (requis pour configurer Zabbix agent 2) :

```
git clone https://git.zabbix.com/scm/zbx/zabbix.git
cd zabbix
./bootstrap.sh
./configure
make dist
```

Une archive des sources sera ainsi créée. Elle pourra ensuite être copiée sur une machine Windows.

2. Ouvrez l'invite de commandes avec des privilèges d'administrateur. Accédez ensuite au répertoire de compilation de Zabbix et compilez Zabbix agent ; veillez à spécifier correctement les répertoires dans lesquels OpenSSL et PCRE2 sont installés :

- Pour les compilations 64 bits :

```
cd C:\Zabbix\zabbix-8.0.0\build\mingw
set PATH=C:\Zabbix\mingw64\bin;%PATH%
mklink /D C:\Zabbix\x64\OpenSSL-Win64-350-static\lib C:\Zabbix\x64\OpenSSL-Win64-350-static\lib64

#### Avec la prise en charge de TLS :
mingw32-make ARCH=AMD64 PCRE2="C:\Zabbix\x64\PCRE2" OPENSSL="C:\Zabbix\x64\OpenSSL-Win64-350-static"

#### Sans la prise en charge de TLS :
mingw32-make ARCH=AMD64 PCRE2="C:\Zabbix\x64\PCRE2"
```

- Pour les compilations 32 bits :

```
cd C:\Zabbix\zabbix-8.0.0\build\mingw
set PATH=C:\Zabbix\mingw32\bin;%PATH%

#### Avec la prise en charge de TLS :
mingw32-make ARCH=x86 PCRE2="C:\Zabbix\x86\PCRE2" OPENSSL="C:\Zabbix\x86\OpenSSL-Win32-350-static"

#### Sans la prise en charge de TLS :
mingw32-make ARCH=x86 PCRE2="C:\Zabbix\x86\PCRE2"
```

Après la compilation, le binaire de Zabbix agent 2 se trouve dans `C:\Zabbix\zabbix-8.0.0\bin\win64` (ou dans `C:\Zabbix\zabbix-8.0.0\bin\win32` pour les compilations 32 bits). Les fichiers de configuration de Zabbix agent 2 se trouvent dans `C:\Zabbix\zabbix-8.0.0\src\go\conf`.

Pour exécuter l'agent, copiez le binaire `zabbix_agent2.exe` et ses fichiers de configuration dans un dossier dédié (par exemple, `C:\Zabbix\agent2`), puis exécutez l'agent :

```
mkdir C:\Zabbix\agent2
copy C:\Zabbix\zabbix-8.0.0\bin\win64\zabbix_agent2.exe C:\Zabbix\agent2\
copy C:\Zabbix\zabbix-8.0.0\src\go\conf\zabbix_agent2.conf C:\Zabbix\agent2\
xcopy /E /I C:\Zabbix\zabbix-8.0.0\src\go\conf\zabbix_agent2.d C:\Zabbix\agent2\zabbix_agent2.d\

C:\Zabbix\agent2\zabbix_agent2.exe -c C:\Zabbix\agent2\zabbix_agent2.conf
```

Si nécessaire, poursuivez avec la compilation des plugins chargeables de Zabbix agent 2.

Compilation de l'agent Zabbix sous Windows

Présentation

Cette page explique comment compiler l'agent Zabbix à partir des sources sous Windows 10 (64 bits).

Ces instructions s'appliquent aux versions de Windows compatibles avec Visual Studio 2022.

La compilation de l'agent Zabbix nécessite :

- Un compilateur C (inclus dans Build Tools for Visual Studio 2022)
- OpenSSL (pour les fonctionnalités de chiffrement de Zabbix)
- PCRE2 (expressions régulières compatibles avec Perl ; pour les fonctionnalités de recherche par motifs d'expressions régulières dans Zabbix)

Vous pouvez compiler l'agent Zabbix à l'aide de l'une des méthodes suivantes :

- Utilisation de vcpkg - une approche automatisée qui simplifie la gestion des dépendances à l'aide d'un gestionnaire de paquets C++.

- Compilation manuelle - une approche manuelle qui nécessite l'installation de toutes les dépendances avant la compilation de l'agent.

Selon vos besoins en matière de supervision, des bibliothèques supplémentaires peuvent être nécessaires. Pour plus d'informations, consultez la page Configuration requise.

Attention:
Avant de commencer le processus de compilation, veuillez tenir compte des points suivants :

- Pour exécuter les commandes, utilisez l'invite de commandes « x64 Native Tools Command Prompt » (incluse dans Build Tools for Visual Studio 2022), lancée par un utilisateur disposant des autorisations suffisantes pour écrire dans les dossiers protégés.
- Il est recommandé de créer un répertoire de travail `C:\Zabbix` pour tous les fichiers sources et les dossiers de compilation. Cependant, les composants compilés doivent être installés dans `C:\Program Files\Zabbix\x64`.

Compilation de l'agent Zabbix avec vcpkg

Cette section contient des instructions pour compiler l'agent Zabbix avec vcpkg, un gestionnaire de paquets qui simplifie la gestion des dépendances et l'intégration aux projets C++.

1. Téléchargez et installez Build Tools for Visual Studio 2022. Lors de l'installation, veillez à sélectionner la charge de travail *Développement Desktop en C++*, qui inclut les outils requis pour compiler l'agent avec vcpkg :

- Compilateur C (Microsoft Visual C++)
- Outil de ligne de commande NMake
- Gestionnaire de paquets vcpkg
- « x64 Native Tools Command Prompt »

2. Ouvrez `cmd.exe` avec des privilèges d'administrateur, initialisez vcpkg et installez les dépendances requises pour compiler l'agent Zabbix (notez que cette opération peut prendre un certain temps) :

```
cd C:\Zabbix
vcpkg new --application
vcpkg add port pcre2
vcpkg add port openssl
vcpkg install --triplet x64-windows-static --x-install-root="C:\Program Files\Zabbix\x64"
```

3. Téléchargez l'archive des sources de Zabbix et extrayez-la dans `C:\Zabbix\zabbix-8.0.0`.

4. Accédez au répertoire de compilation de Zabbix (`C:\Zabbix\zabbix-8.0.0\build\win32\project`) et créez le script `build.bat` suivant ; veillez à spécifier correctement les répertoires dans lesquels OpenSSL et PCRE2 sont installés :

```
:: Définir le chemin d'installation de vcpkg :
set vcpkg=C:\Program Files\Zabbix\x64\x64-windows-static

:: Exécuter le processus de compilation :
nmake -f Makefile CPU=AMD64 ^
    PCRE2INCDIR="%vcpkg%\include" ^
    PCRE2LIBDIR="%vcpkg%\lib" ^
    TLS=openssl ^
    TLSINCDIR="%vcpkg%\include" ^
    TLSLIBDIR="%vcpkg%\lib" ^
    LIBS="$(LIBS) Crypt32.lib" ^
    all
```

5. Ouvrez « x64 Native Tools Command Prompt » et compilez l'agent Zabbix en exécutant le script :

```
cd C:\Zabbix\zabbix-8.0.0\build\win32\project
build.bat
```

Si vous avez ouvert une autre invite de commandes, initialisez d'abord l'environnement de compilation de Visual Studio :

```
"%ProgramFiles(x86)%\Microsoft Visual Studio\2022\BuildTools\VC\Auxiliary\Build\vcvarsall.bat" x64
cd C:\Zabbix\zabbix-8.0.0\build\win32\project
build.bat
```

Placez entre guillemets les chemins qui contiennent des espaces.

Après la compilation, les fichiers binaires des composants Zabbix se trouvent dans `C:\Zabbix\zabbix-8.0.0\bin\win64`. Le fichier de configuration de l'agent Zabbix se trouve dans `C:\Zabbix\zabbix-8.0.0\conf`.

Pour exécuter l'agent, copiez `zabbix_agentd.exe` et son fichier de configuration dans un dossier dédié (par exemple, `C:\Zabbix\agent`), puis exécutez l'agent :

```
mkdir C:\Zabbix\agent
copy C:\Zabbix\zabbix-8.0.0\bin\win64\zabbix_agentd.exe C:\Zabbix\agent\
copy C:\Zabbix\zabbix-8.0.0\conf\zabbix_agentd.conf C:\Zabbix\agent\

C:\Zabbix\agent\zabbix_agentd.exe -c C:\Zabbix\agent\zabbix_agentd.conf
```

Compilation manuelle de l'agent Zabbix

Attention:
Cette méthode de compilation de l'agent Zabbix convient aux utilisateurs qui ont besoin d'un contrôle total sur l'environnement de compilation ou qui se trouvent dans un environnement restreint où l'utilisation de vcpkg n'est pas possible.

Cette section contient des instructions pour compiler manuellement l'agent Zabbix, ce qui inclut l'installation des outils de compilation et des dépendances requis (Perl, OpenSSL, PCRE2), puis la compilation de l'agent.

Installation des outils de compilation

1. Téléchargez et installez Build Tools for Visual Studio 2022. Lors de l'installation, veillez à sélectionner la charge de travail *Desktop development with C++*, qui inclut les outils requis pour compiler manuellement l'agent :

- Compilateur C (Microsoft Visual C++)
- Outil de ligne de commande NMake
- « x64 Native Tools Command Prompt »

Installation d'OpenSSL

Note:
Pour compiler l'agent Zabbix sans prise en charge de TLS, passez à la section Installation de PCRE2.

1. Téléchargez et installez Strawberry Perl (disponible sous forme d'installateur MSI). Lors de l'installation, veillez à spécifier `C:\Zabbix\Strawberry` comme dossier d'installation.

2. Installez le module Perl `Text::Template` :

```
cpanm Text::Template
```

3. Vérifiez que le Netwide Assembler (NASM ; requis pour compiler OpenSSL) a été compilé lors de l'installation de Strawberry Perl :

```
nasm -v
#### NASM version 2.16.01 compiled on May  3 2024
```

Si NASM n'est pas compilé, installez-le manuellement. Pour plus d'informations, consultez la documentation de NASM.

4. Téléchargez l'archive des sources d'OpenSSL et extrayez-la dans `C:\Zabbix\openssl-3.5.0`.

5. Accédez au répertoire extrait et configurez OpenSSL, par exemple :

```
cd C:\Zabbix\openssl-3.5.0
perl Configure VC-WIN64A no-shared no-capieng no-winstore no-srp no-gost no-dgram no-dtls1-method no-dtls1
```

Attention:
Si vous choisissez un répertoire personnalisé pour OpenSSL lors de la compilation de l'agent Zabbix sous Windows (par exemple, `C:\zabbix` ou `C:\openssl-64bit`), veillez à révoquer l'accès en écriture à ce répertoire pour les utilisateurs qui ne disposent pas de privilèges d'administrateur. Sinon, l'agent chargera les paramètres SSL depuis un chemin pouvant être modifié par des utilisateurs non privilégiés, ce qui peut entraîner une vulnérabilité de sécurité.

- L'option `no-shared` rend les bibliothèques statiques libcrypto.lib et libssl.lib d'OpenSSL autonomes, de sorte que les binaires Zabbix incluent OpenSSL sans nécessiter de DLL externes. Cela signifie que les binaires Zabbix peuvent être copiés sur d'autres machines Windows sans les bibliothèques OpenSSL ; toutefois, lorsqu'une nouvelle version corrective d'OpenSSL est publiée, l'agent Zabbix doit être recompilé.
- Sans l'option `no-shared`, Zabbix dépend des DLL OpenSSL lors de l'exécution. Cela signifie que les mises à jour d'OpenSSL peuvent ne pas nécessiter la recompilation de l'agent Zabbix ; toutefois, lors de sa copie sur d'autres machines, les DLL OpenSSL doivent également être copiées.

Pour plus d'informations sur les autres options de configuration d'OpenSSL, consultez la documentation d'OpenSSL.

6. Compilez OpenSSL et exécutez les tests (notez que cette opération peut prendre un certain temps) :

> **Attention:**
> Exécutez les tests sans privilèges d'administrateur ; sinon, cela peut entraîner des résultats inattendus ou des risques de sécurité. Si certains tests échouent, consultez la documentation d'OpenSSL pour résoudre le problème.

```
nmake
nmake test
...
All tests successful.
Files=325, Tests=3101, 822 wallclock secs ( 4.81 usr +  0.81 sys =  5.62 CPU)
Result: PASS
```

7. Installez OpenSSL :

```
nmake install
```

Pour installer uniquement les composants logiciels (bibliothèques et fichiers d'en-tête, mais pas la documentation), vous pouvez utiliser `nmake install_sw`.

Installation de PCRE2

1. Téléchargez et installez CMake (disponible sous forme d'installateur MSI). Pendant l'installation, veillez à spécifier `C:\Zabbix\CMake` comme dossier d'installation et à sélectionner l'option *Add CMake to the PATH environment variable*.

2. Téléchargez l'archive source de PCRE2 et extrayez-la dans `C:\Zabbix\pcre2-10.45`.

3. Créez un répertoire `build` dans le répertoire PCRE2 extrait et accédez-y :

```
mkdir C:\Zabbix\pcre2-10.45\build
cd C:\Zabbix\pcre2-10.45\build
```

4. Configurez PCRE2 :

```
cmake -G "NMake Makefiles" -DPCRE_SUPPORT_UNICODE_PROPERTIES=ON -DCMAKE_BUILD_TYPE=Release -DCMAKE_INSTALL
```

> **Note:**
> Si des erreurs se produisent, il est recommandé de supprimer le cache CMake avant de tenter de répéter le processus de compilation CMake. Le cache (`CMakeCachecache.txt`) se trouve dans le répertoire build du répertoire PCRE2 extrait.

5. Compilez PCRE2 à l'aide de NMake :

```
nmake
```

6. Installez PCRE2 :

```
cmake --install .
```

Compilation de l'agent Zabbix

1. Téléchargez l'archive des sources de Zabbix et extrayez-la dans `C:\Zabbix\zabbix-8.0.0`.

Si vous devez générer une archive des sources à partir du dépôt de sources brutes (par exemple, pour appliquer des correctifs personnalisés ou compiler à partir du code source le plus récent), exécutez les commandes suivantes sur une machine **Linux** :

```
git clone https://git.zabbix.com/scm/zbx/zabbix.git
cd zabbix
./bootstrap.sh
./configure
make dist
```

Une archive des sources sera créée. Vous pourrez ensuite la copier sur une machine Windows.

2. Accédez au répertoire de compilation de Zabbix, ouvrez l'« invite de commandes des outils natifs x64 » et compilez l'agent Zabbix (ou d'autres composants) ; veillez à spécifier correctement les répertoires dans lesquels OpenSSL et PCRE2 sont installés :

```
"%ProgramFiles(x86)%\Microsoft Visual Studio\2022\BuildTools\VC\Auxiliary\Build\vcvarsall.bat" x64

cd C:\Zabbix\zabbix-8.0.0\build\win32\project
```

```
#### Avec la prise en charge de TLS :
nmake /K -f Makefile_agent PCRE2INCDIR="C:\Program Files\Zabbix\x64\PCRE2\include" PCRE2LIBDIR="C:\Program
nmake /K -f Makefile_get PCRE2INCDIR="C:\Program Files\Zabbix\x64\PCRE2\include" PCRE2LIBDIR="C:\Program F
nmake /K -f Makefile_sender PCRE2INCDIR="C:\Program Files\Zabbix\x64\PCRE2\include" PCRE2LIBDIR="C:\Progra

#### Sans la prise en charge de TLS :
nmake /K -f Makefile_agent PCRE2INCDIR="C:\Program Files\Zabbix\x64\PCRE2\include" PCRE2LIBDIR="C:\Program
nmake /K -f Makefile_get PCRE2INCDIR="C:\Program Files\Zabbix\x64\PCRE2\include" PCRE2LIBDIR="C:\Program F
nmake /K -f Makefile_sender PCRE2INCDIR="C:\Program Files\Zabbix\x64\PCRE2\include" PCRE2LIBDIR="C:\Progra
```

Placez entre guillemets les chemins qui contiennent des espaces.

Après la compilation, les fichiers binaires des composants Zabbix se trouvent dans `C:\Zabbix\zabbix-8.0.0\bin\win64`. Le fichier de configuration de l'agent Zabbix se trouve dans `C:\Zabbix\zabbix-8.0.0\conf`.

Pour exécuter l'agent, copiez `zabbix_agentd.exe` et son fichier de configuration dans un dossier dédié (par exemple, `C:\Zabbix\agent`), puis exécutez l'agent :

```
mkdir C:\Zabbix\agent
copy C:\Zabbix\zabbix-8.0.0\bin\win64\zabbix_agentd.exe C:\Zabbix\agent\
copy C:\Zabbix\zabbix-8.0.0\conf\zabbix_agentd.conf C:\Zabbix\agent\

C:\Zabbix\agent\zabbix_agentd.exe -c C:\Zabbix\agent\zabbix_agentd.conf -f
```

Installation de l'agent macOS à partir d'un PKG

Vue d'ensemble

L'agent Zabbix peut être installé sur macOS à l'aide de packages d'installation PKG, disponibles en téléchargement.

Les packages de l'agent Zabbix sont disponibles avec ou sans chiffrement.

Installation de l'agent

L'agent peut être installé à l'aide de l'interface utilisateur graphique ou depuis la ligne de commande, par exemple :

```
sudo installer -pkg zabbix_agent-8.0.0-macos-arm64-openssl.pkg -target /
```

Veillez à utiliser la version correcte du paquet Zabbix dans la commande. Elle doit correspondre au nom du paquet téléchargé.

Exécution de l'agent

L'agent démarrera automatiquement après l'installation ou le redémarrage.

Vous pouvez modifier le fichier de configuration situé à `/usr/local/etc/zabbix/zabbix_agentd.conf` si nécessaire.

Pour démarrer l'agent manuellement, vous pouvez exécuter :

```
sudo launchctl start com.zabbix.zabbix_agentd
```

Pour arrêter l'agent manuellement :

```
sudo launchctl stop com.zabbix.zabbix_agentd
```

Lors de la mise à niveau, le fichier de configuration existant n'est pas écrasé. À la place, un nouveau fichier `zabbix_agentd.conf.NEW` est créé afin de permettre la révision et la mise à jour du fichier de configuration existant, si nécessaire. N'oubliez pas de redémarrer l'agent après avoir apporté des modifications manuelles au fichier de configuration.

Dépannage et suppression de l'agent

Cette section répertorie quelques commandes utiles pouvant être utilisées pour le dépannage et la suppression de l'installation de l'agent Zabbix.

Vérifiez si l'agent Zabbix est en cours d'exécution :

```
ps aux | grep zabbix_agentd
```

Vérifiez si l'agent Zabbix a été installé à partir de paquets :

```
pkgutil --pkgs | grep zabbix
com.zabbix.pkg.ZabbixAgent
```

Affichez les fichiers qui ont été installés à partir du paquet d'installation (notez que le / initial n'est pas affiché dans cette vue) :

```
pkgutil --only-files --files com.zabbix.pkg.ZabbixAgent
Library/LaunchDaemons/com.zabbix.zabbix_agentd.plist
usr/local/bin/zabbix_get
usr/local/bin/zabbix_sender
usr/local/etc/zabbix/zabbix_agentd/userparameter_examples.conf.NEW
usr/local/etc/zabbix/zabbix_agentd/userparameter_mysql.conf.NEW
usr/local/etc/zabbix/zabbix_agentd.conf.NEW
usr/local/sbin/zabbix_agentd
```

Arrêtez l'agent Zabbix s'il a été lancé avec `launchctl` :

```
sudo launchctl unload /Library/LaunchDaemons/com.zabbix.zabbix_agentd.plist
```

Supprimez les fichiers (y compris la configuration et les journaux) qui ont été installés avec le paquet d'installation :

```
sudo rm -f /Library/LaunchDaemons/com.zabbix.zabbix_agentd.plist
sudo rm -f /usr/local/sbin/zabbix_agentd
sudo rm -f /usr/local/bin/zabbix_get
sudo rm -f /usr/local/bin/zabbix_sender
sudo rm -rf /usr/local/etc/zabbix
sudo rm -rf /var/log/zabbix
```

Faites oublier que l'agent Zabbix a été installé :

```
sudo pkgutil --forget com.zabbix.pkg.ZabbixAgent
```

Compilation de l'agent Zabbix sur macOS

Aperçu

Cette section montre comment créer des binaires d'agent Zabbix macOS à partir de sources avec ou sans TLS.

Prérequis

Vous aurez besoin des outils de développement en ligne de commande (Xcode n'est pas requis), d'Automake, de pkg-config et de PCRE (v8.x) ou PCRE2 (v10.x). Si vous souhaitez compiler des binaires d'agent avec TLS, vous aurez également besoin d'OpenSSL ou de GnuTLS.

Pour installer Automake et pkg-config, vous aurez besoin du gestionnaire de paquets Homebrew disponible sur https://brew.sh/. Pour l'installer, ouvrez le terminal et exécutez la commande suivante :

```
/bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"
```

Installez ensuite Automake et pkg-config :

```
brew install automake
brew install pkg-config
```

La préparation des bibliothèques PCRE, OpenSSL et GnuTLS dépend de la manière dont elles seront liées à l'agent.

Si vous prévoyez d'exécuter des binaires d'agent sur une machine macOS qui dispose déjà de ces bibliothèques, vous pouvez utiliser les bibliothèques précompilées fournies par Homebrew. Il s'agit généralement de machines macOS qui utilisent Homebrew pour compiler des binaires d'agent Zabbix ou à d'autres fins.

Si les binaires d'agent doivent être utilisés sur des machines macOS qui ne disposent pas de la version partagée des bibliothèques, vous devez compiler des bibliothèques statiques à partir des sources et lier l'agent Zabbix avec celles-ci.

Compilation des binaires de l'agent avec des bibliothèques partagées

Installez PCRE2 (remplacez *pcre2* par *pcre* dans les commandes ci-dessous, si nécessaire) :

```
brew install pcre2
```

Lors de la compilation avec TLS, installez OpenSSL et/ou GnuTLS :

```
brew install openssl
brew install gnutls
```

Téléchargez les sources de Zabbix :

```
git clone https://git.zabbix.com/scm/zbx/zabbix.git
```

Compilez l'agent sans TLS :

```
cd zabbix
./bootstrap.sh
./configure --sysconfdir=/usr/local/etc/zabbix --enable-agent --enable-ipv6
make
make install
```

Compilez l'agent avec OpenSSL :

```
cd zabbix
./bootstrap.sh
./configure --sysconfdir=/usr/local/etc/zabbix --enable-agent --enable-ipv6 --with-openssl=/usr/local/opt/
make
make install
```

Compilez l'agent avec GnuTLS :

```
cd zabbix-source/
./bootstrap.sh
./configure --sysconfdir=/usr/local/etc/zabbix --enable-agent --enable-ipv6 --with-gnutls=/usr/local/opt/g
make
make install
```

Compilation des binaires de l'agent avec des bibliothèques statiques sans TLS

Supposons que les bibliothèques statiques PCRE seront installées dans `$HOME/static-libs`. Nous utiliserons PCRE2 10.39.

```
PCRE_PREFIX="$HOME/static-libs/pcre2-10.39"
```

Téléchargez et compilez PCRE avec la prise en charge des propriétés Unicode :

```
mkdir static-libs-source
cd static-libs-source
curl --remote-name https://github.com/PhilipHazel/pcre2/releases/download/pcre2-10.39/pcre2-10.39.tar.gz
tar xf pcre2-10.39.tar.gz
cd pcre2-10.39
./configure --prefix="$PCRE_PREFIX" --disable-shared --enable-static --enable-unicode-properties
make
make check
make install
```

Téléchargez le code source de Zabbix et compilez l'agent :

```
git clone https://git.zabbix.com/scm/zbx/zabbix.git
cd zabbix
./bootstrap.sh
./configure --sysconfdir=/usr/local/etc/zabbix --enable-agent --enable-ipv6 --with-libpcre2="$PCRE_PREFIX"
make
make install
```

Compilation des binaires de l'agent avec des bibliothèques statiques et OpenSSL

Lors de la compilation d'OpenSSL, il est recommandé d'exécuter `make test` après une compilation réussie. Même si la compilation a réussi, les tests échouent parfois. Si c'est le cas, les problèmes doivent être analysés et résolus avant de continuer.

Supposons que les bibliothèques statiques PCRE et OpenSSL seront installées dans `$HOME/static-libs`. Nous utiliserons PCRE2 10.39 et OpenSSL 1.1.1a.

```
PCRE_PREFIX="$HOME/static-libs/pcre2-10.39"
OPENSSL_PREFIX="$HOME/static-libs/openssl-1.1.1a"
```

Compilons les bibliothèques statiques dans `static-libs-source` :

```
mkdir static-libs-source
cd static-libs-source
```

Téléchargez et compilez PCRE avec la prise en charge des propriétés Unicode :

```
curl --remote-name https://github.com/PhilipHazel/pcre2/releases/download/pcre2-10.39/pcre2-10.39.tar.gz
tar xf pcre2-10.39.tar.gz
cd pcre2-10.39
./configure --prefix="$PCRE_PREFIX" --disable-shared --enable-static --enable-unicode-properties
make
```

```
make check
make install
cd ..
```

Téléchargez et compilez OpenSSL :

```
curl --remote-name https://www.openssl.org/source/openssl-1.1.1a.tar.gz
tar xf openssl-1.1.1a.tar.gz
cd openssl-1.1.1a
./Configure --prefix="$OPENSSL_PREFIX" --openssldir="$OPENSSL_PREFIX" --api=1.1.0 no-shared no-capieng no-
make
make test
make install_sw
cd ..
```

Téléchargez le code source de Zabbix et compilez l'agent :

```
git clone https://git.zabbix.com/scm/zbx/zabbix.git
cd zabbix
./bootstrap.sh
./configure --sysconfdir=/usr/local/etc/zabbix --enable-agent --enable-ipv6 --with-libpcre2="$PCRE_PREFIX"
make
make install
```

Compilation des binaires de l'agent avec des bibliothèques statiques avec GnuTLS

GnuTLS dépend du backend cryptographique Nettle et de la bibliothèque arithmétique GMP. Au lieu d'utiliser la bibliothèque GMP complète, ce guide utilisera mini-gmp, qui est incluse dans Nettle.

Lors de la compilation de GnuTLS et de Nettle, il est recommandé d'exécuter `make check` après une compilation réussie. Même si la compilation a réussi, les tests échouent parfois. Si c'est le cas, les problèmes doivent être analysés et résolus avant de continuer.

Supposons que les bibliothèques statiques PCRE, Nettle et GnuTLS seront installées dans `$HOME/static-libs`. Nous utiliserons PCRE2 10.39, Nettle 3.4.1 et GnuTLS 3.6.5.

```
PCRE_PREFIX="$HOME/static-libs/pcre2-10.39"
NETTLE_PREFIX="$HOME/static-libs/nettle-3.4.1"
GNUTLS_PREFIX="$HOME/static-libs/gnutls-3.6.5"
```

Compilons les bibliothèques statiques dans `static-libs-source` :

```
mkdir static-libs-source
cd static-libs-source
```

Téléchargez et compilez Nettle :

```
curl --remote-name https://ftp.gnu.org/gnu/nettle/nettle-3.4.1.tar.gz
tar xf nettle-3.4.1.tar.gz
cd nettle-3.4.1
./configure --prefix="$NETTLE_PREFIX" --enable-static --disable-shared --disable-documentation --disable-a
make
make check
make install
cd ..
```

Téléchargez et compilez GnuTLS :

```
curl --remote-name https://www.gnupg.org/ftp/gcrypt/gnutls/v3.6/gnutls-3.6.5.tar.xz
tar xf gnutls-3.6.5.tar.xz
cd gnutls-3.6.5
PKG_CONFIG_PATH="$NETTLE_PREFIX/lib/pkgconfig" ./configure --prefix="$GNUTLS_PREFIX" --enable-static --dis
make
make check
make install
cd ..
```

Téléchargez le code source de Zabbix et compilez l'agent :

```
git clone https://git.zabbix.com/scm/zbx/zabbix.git
cd zabbix
./bootstrap.sh
```

```
CFLAGS="-Wno-unused-command-line-argument -framework Foundation -framework Security" \
> LIBS="-lgnutls -lhogweed -lnettle" \
> LDFLAGS="-L$GNUTLS_PREFIX/lib -L$NETTLE_PREFIX/lib" \
> ./configure --sysconfdir=/usr/local/etc/zabbix --enable-agent --enable-ipv6 --with-libpcre2="$PCRE_PREFI
make
make install
```

Guide de déploiement d'AWS pour le serveur Zabbix

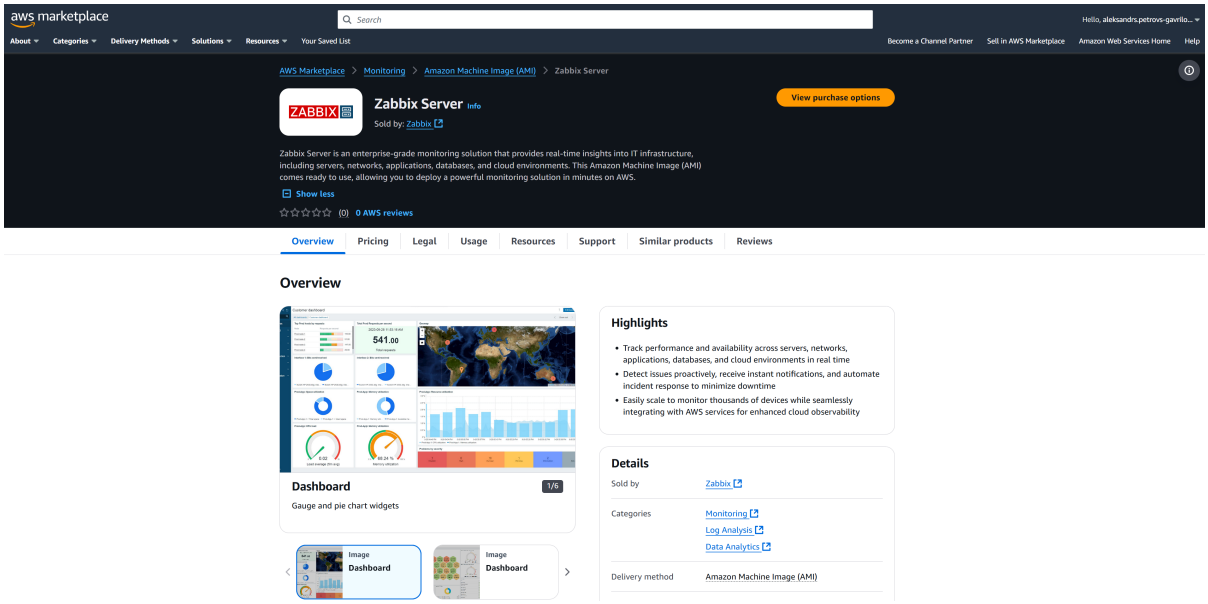
Présentation générale

Cette page décrit comment déployer le serveur Zabbix à l'aide d'AWS.

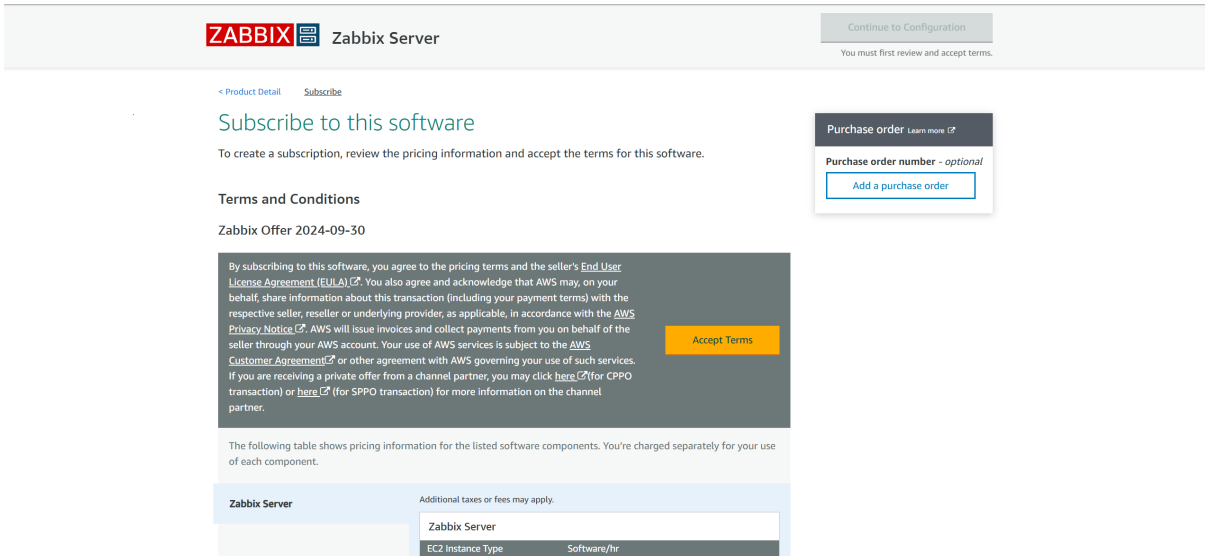
Attention:
Plusieurs versions de Zabbix sont disponibles pour le déploiement sur la plateforme. Veillez à consulter la documentation correspondant à la même version de Zabbix que celle que vous avez déployée. La documentation des différentes versions de Zabbix peut être sélectionnée dans le menu déroulant des versions.

Déploiement de l'appliance

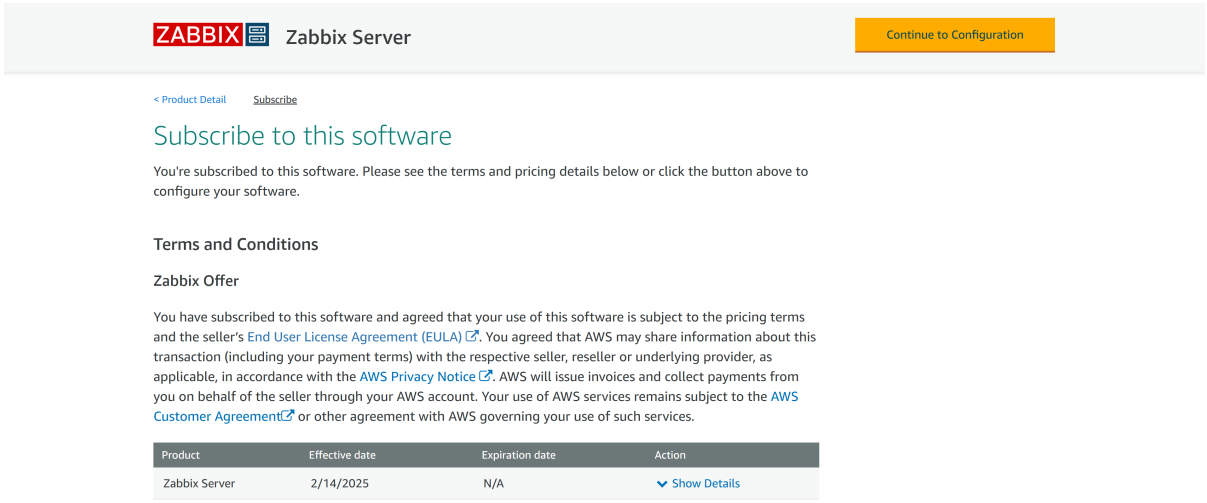
Pour déployer l'appliance Zabbix, accédez d'abord à la page Zabbix AWS marketplace. Cliquez ensuite sur *View purchase options* pour déployer l'image et accéder à la page d'abonnement.



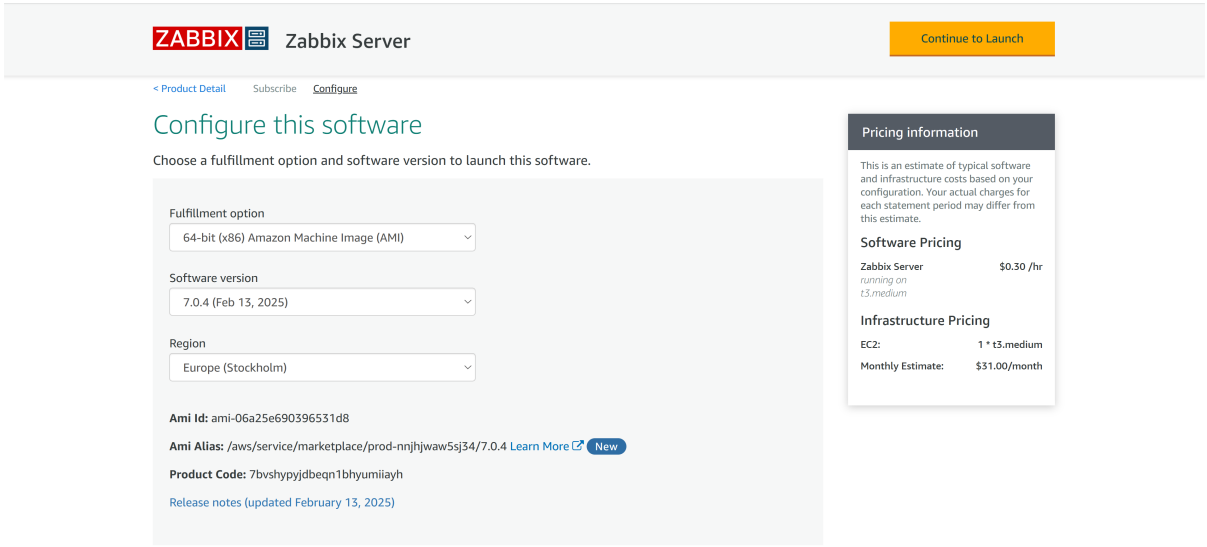
Cliquez sur le bouton *Accept terms* pour accepter les *Terms and Conditions*.



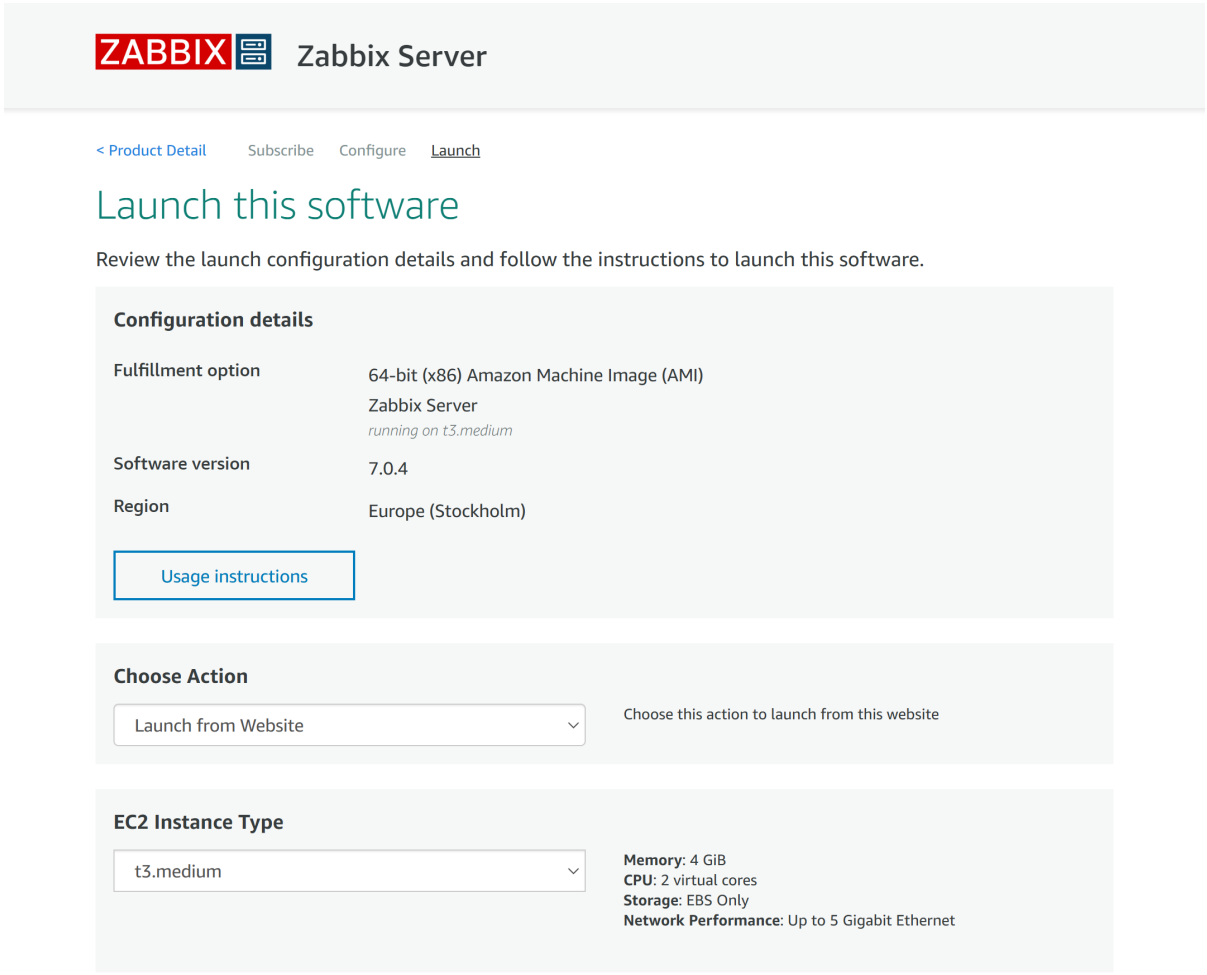
Cliquez sur le bouton *Continue to Configuration* pour accéder à la section Configuration de l'appliance Zabbix.



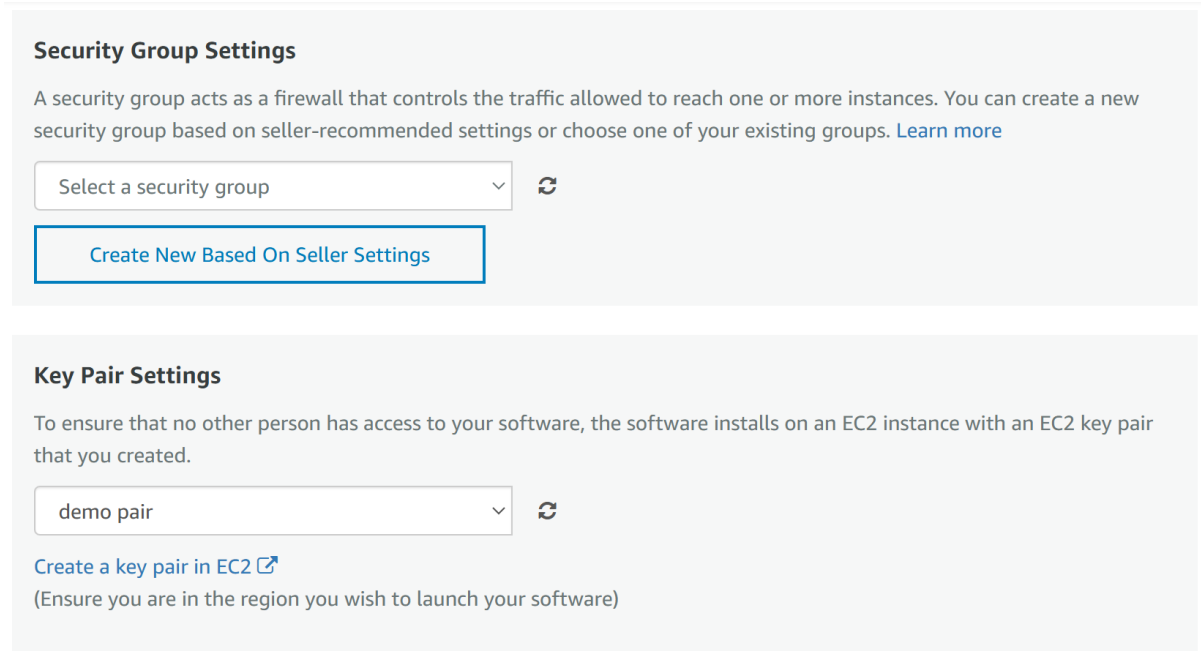
Sélectionnez l'option Fulfilment, la version de Zabbix et la Région de votre choix.



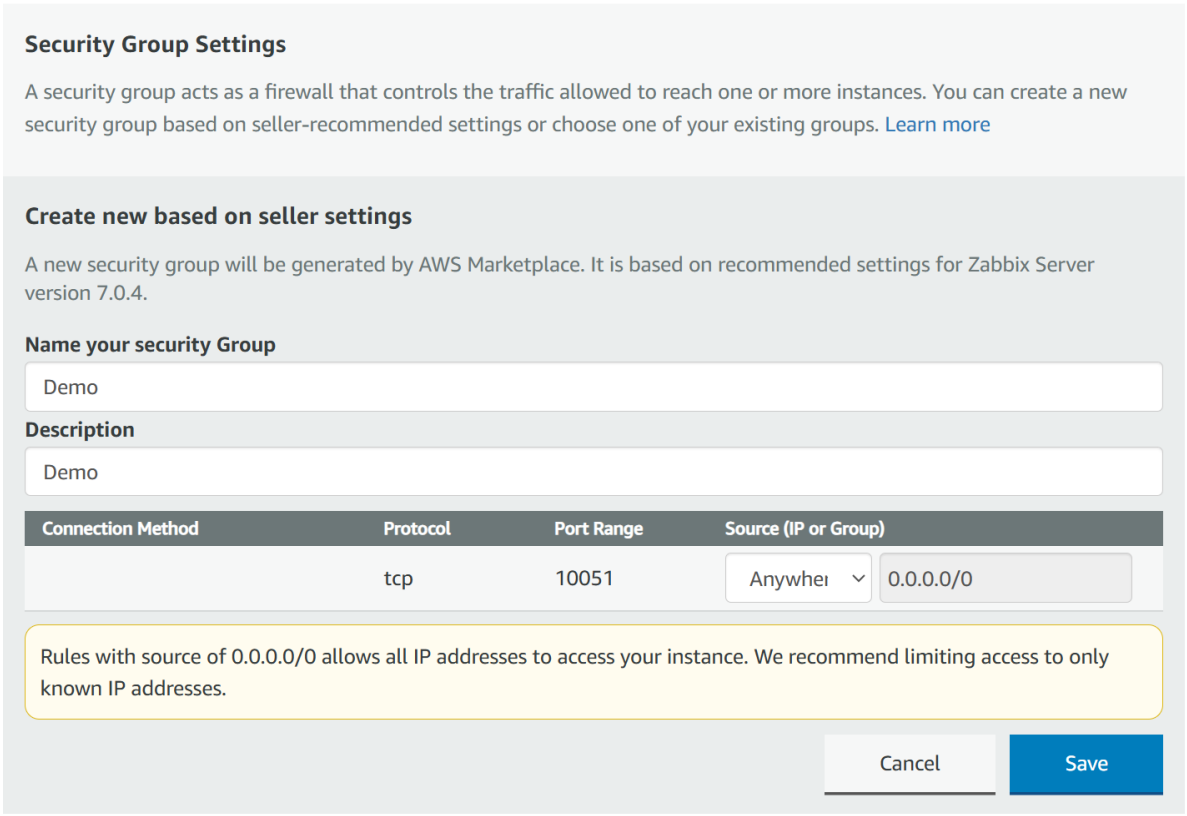
Cliquez sur le bouton *Continue to Launch*. Sur la page Launch, sélectionnez l'option Launch, le type d'instance EC2, le VPC et les paramètres de Subnet de votre choix.



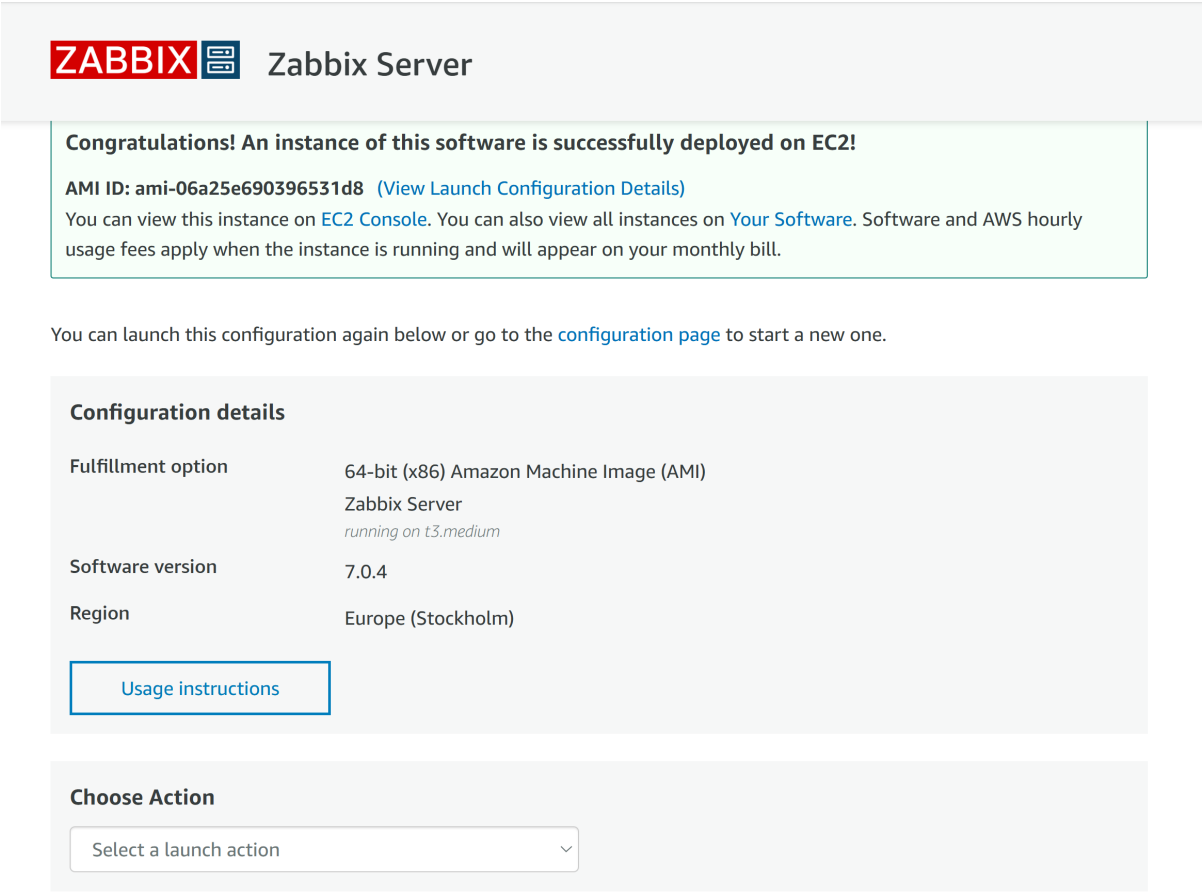
Au bas de la page Launch, indiquez les paramètres du Security Group et sélectionnez ou créez une paire de clés EC2 - veillez à enregistrer votre clé privée dans un emplacement sûr !



Si vous sélectionnez *Create a new Security Group based on seller settings*, le port TCP 10051 (serveur Zabbix) sera autorisé. Remarque : si vous souhaitez accéder à l'instance via SSH et HTTP/HTTPS (interface Web), vous devrez également autoriser les connexions SSH et HTTP/HTTPS dans les paramètres de votre Security Group.

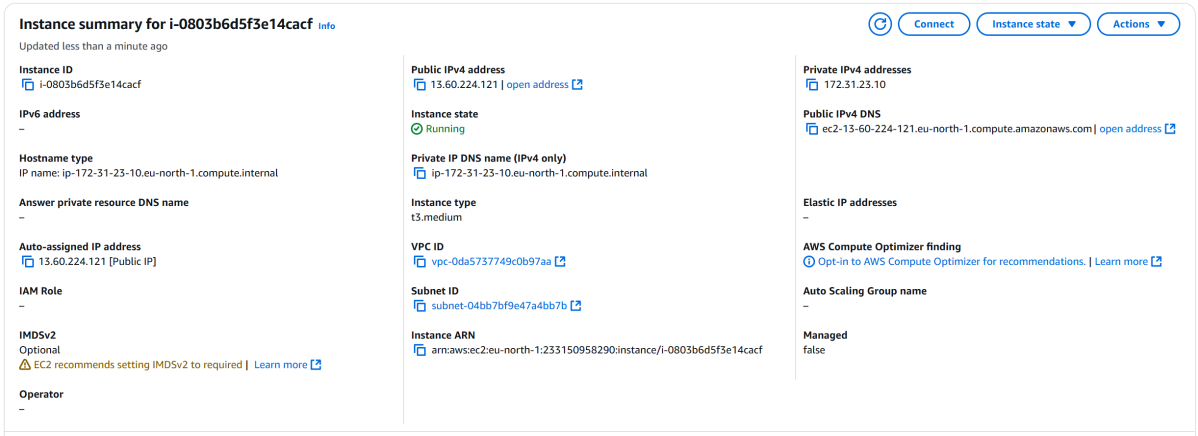


Félicitations, votre appliance Zabbix a été déployée avec succès.

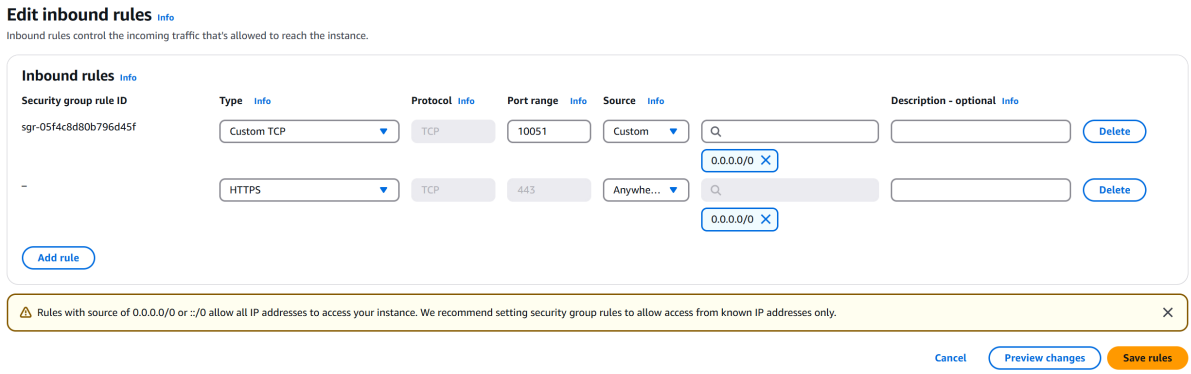


Accéder à l'appliance

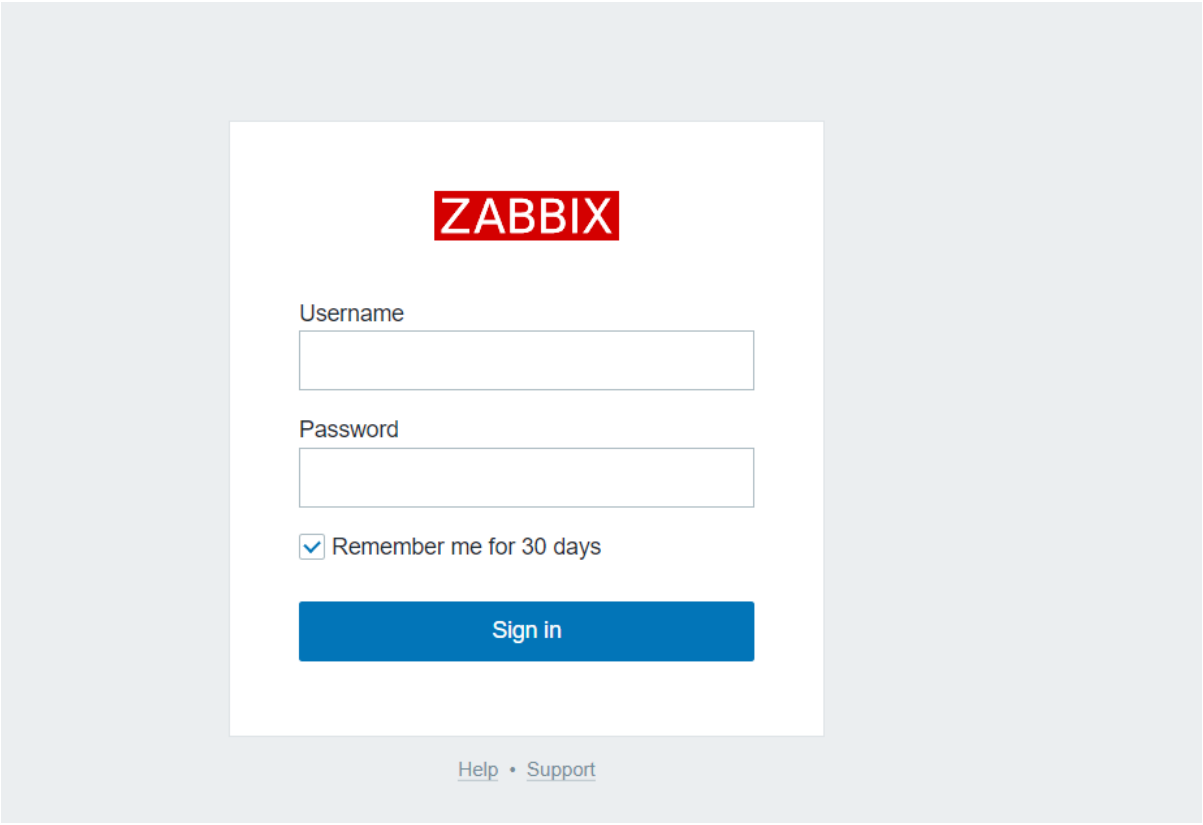
Ouvrez la section *Instances* et sélectionnez l'instance de l'appliance Zabbix nouvellement déployée.



Pour accéder à l'interface Zabbix à l'aide de l'adresse IPv4 publique ou du DNS IPv4 public de l'instance, autorisez le trafic HTTPS entrant sur le port 443 dans le groupe de sécurité.



Vous pouvez maintenant accéder à l'interface Zabbix en ouvrant l'adresse IPv4 publique ou le DNS IPv4 public de l'appliance Zabbix.



Pour obtenir le mot de passe, vous devez vous connecter à la machine à l'aide de la console série ou de SSH. Le nom d'utilisateur SSH par défaut est rocky. N'oubliez pas d'utiliser la paire de clés que vous avez sélectionnée ou générée précédemment. Veuillez noter que la génération des identifiants peut prendre du temps. La génération des identifiants peut prendre quelques minutes. Attendez quelques minutes avant d'essayer de vous connecter.

```
Last login: Tue Nov 16 15:16:07 2021 from ec2-13-48-4-200.eu-north-1.compute.amazonaws.com
********************************************************************************

Zabbix frontend credentials:

Username: Admin

Password: gJ8f7XpgJbsnc2BU48g

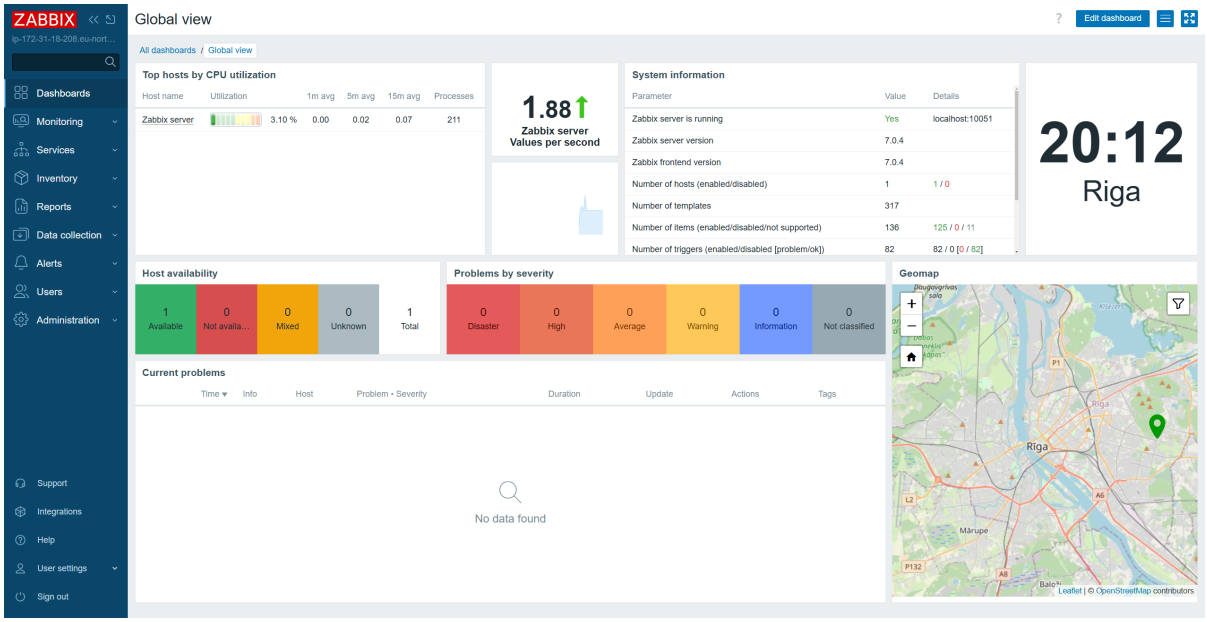
To learn about available professional services, including technical suppport and training, please visit https://www.zabbix.com/services

Official Zabbix documentation available at https://www.zabbix.com/documentation/current/

Note! Do not forget to change timezone PHP variable in /etc/php.d/99-zabbix.ini file.

********************************************************************************
```

Vous pouvez maintenant utiliser ces informations pour vous connecter à votre interface Zabbix.



Accéder à la base de données

Vous pouvez accéder à la base de données Zabbix avec l'utilisateur `root` sans saisir de mot de passe.

```
[root@ip-172-31-29-138 ~]# mysql -uroot
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 124
Server version: 8.0.26 MySQL Community Server - GPL

Copyright (c) 2000, 2021, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

Les identifiants sont stockés dans le fichier de configuration `/root/.my.cnf`.

Le fuseau horaire par défaut de l'interface Zabbix peut être modifié dans *Administration > Général > Interface graphique*.

Guide de déploiement AWS pour le proxy Zabbix

Présentation générale

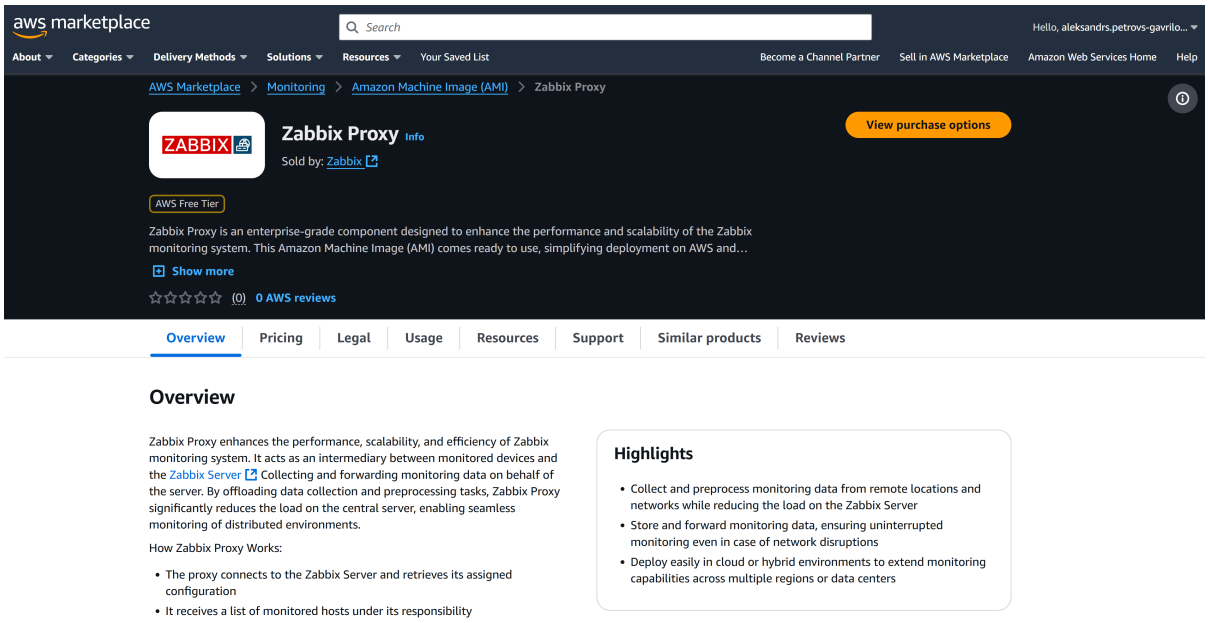
Cette page décrit comment déployer un proxy Zabbix à l'aide d'AWS.

Attention:
Plusieurs versions de Zabbix sont disponibles pour un déploiement sur la plateforme. Veillez à consulter la documentation correspondant à la même version de Zabbix que celle que vous avez déployée. La documentation des différentes versions de Zabbix peut être sélectionnée dans le menu déroulant des versions.

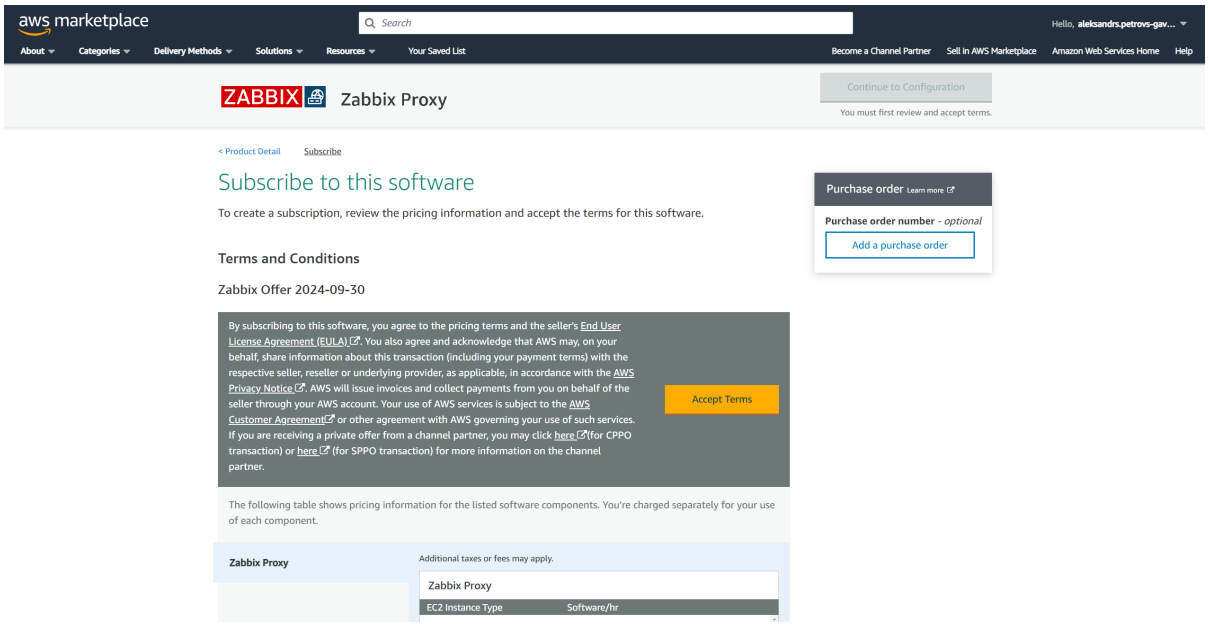
Déploiement du proxy Zabbix

Attention:
Une instance fonctionnelle du serveur Zabbix est requise pour utiliser le composant proxy Zabbix.

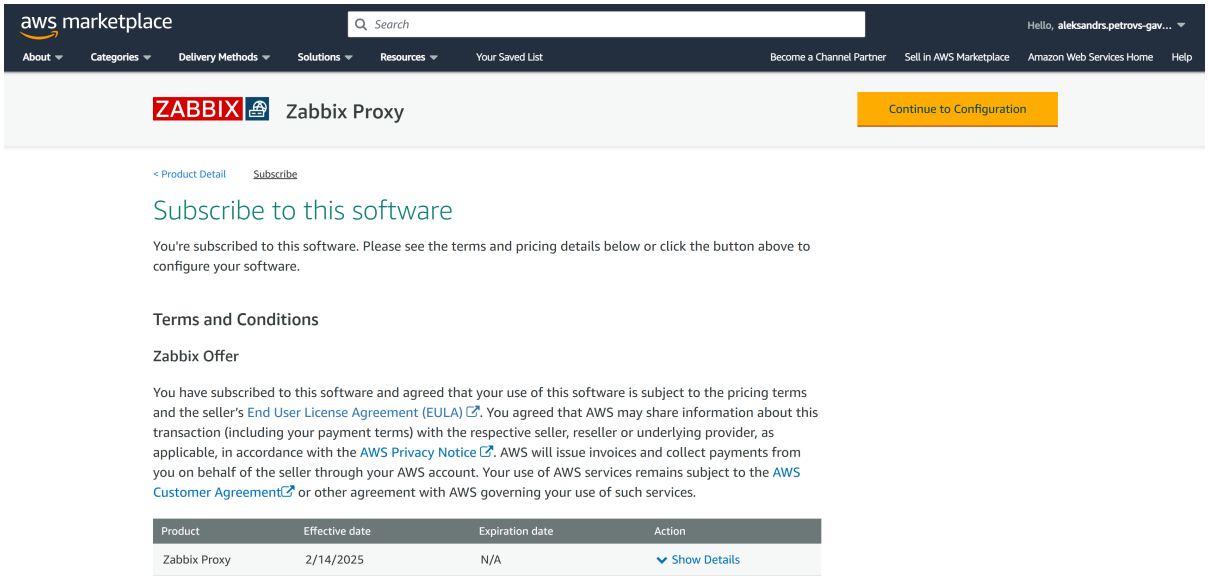
Pour déployer le proxy Zabbix, accédez d'abord à la page Zabbix sur AWS Marketplace. Cliquez ensuite sur *View purchase options* pour déployer l'image et accéder à la page d'abonnement.



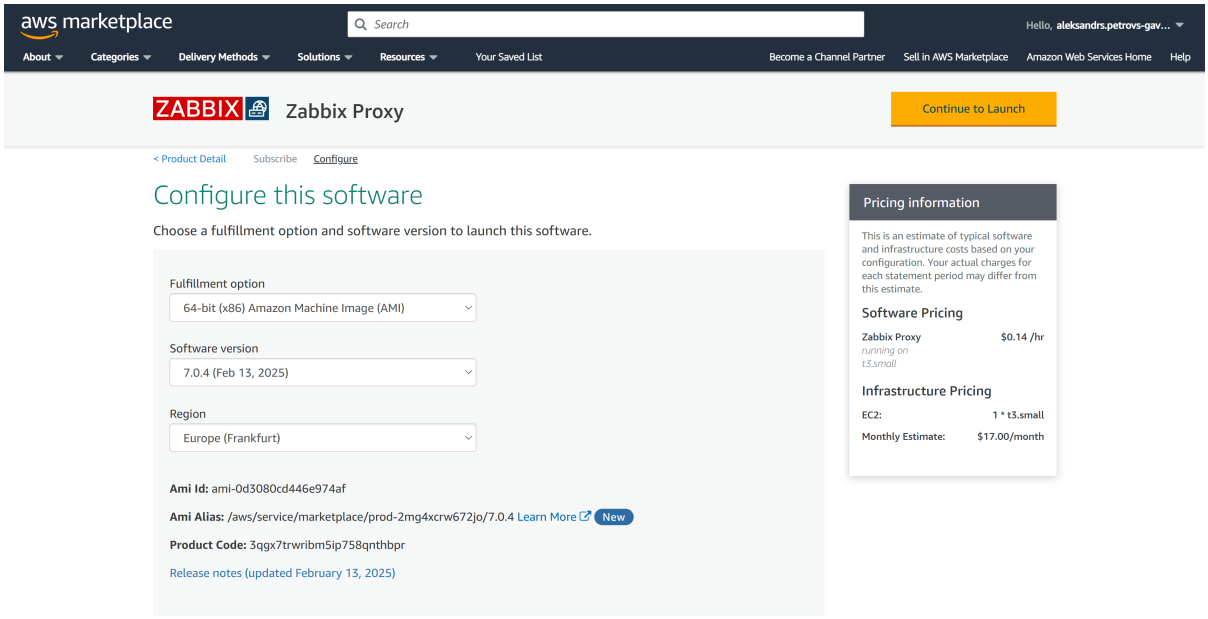
Cliquez sur le bouton *Accept terms* pour accepter les *Terms and Conditions*.



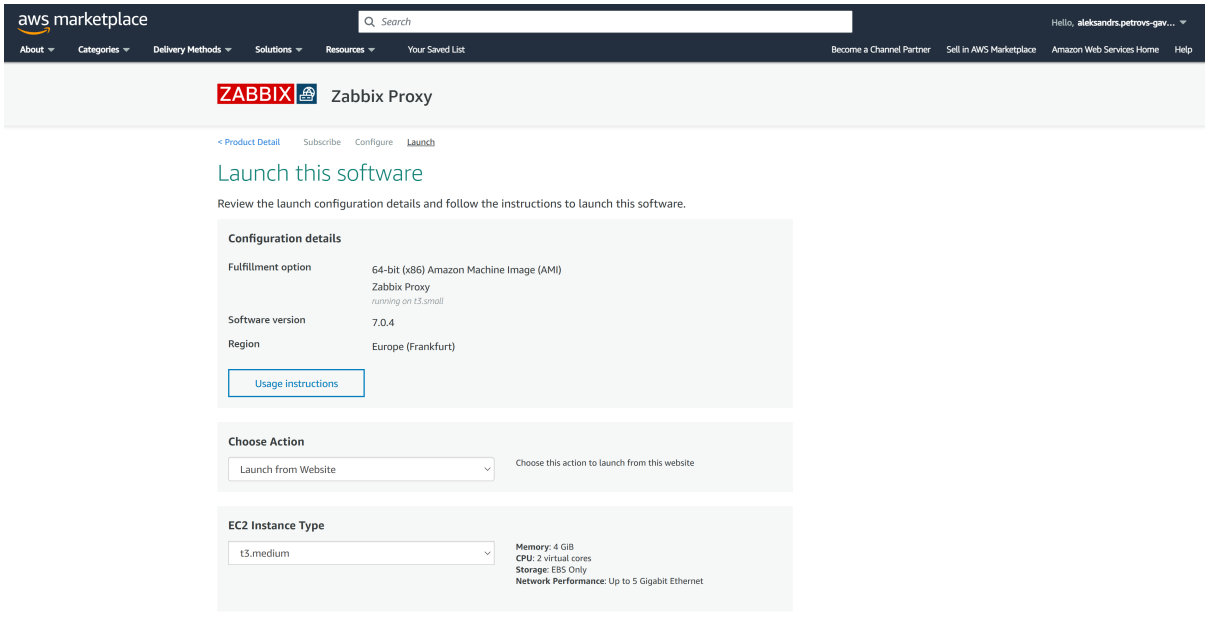
Cliquez sur le bouton *Continue to Configuration* pour accéder à la section Configuration du proxy Zabbix.



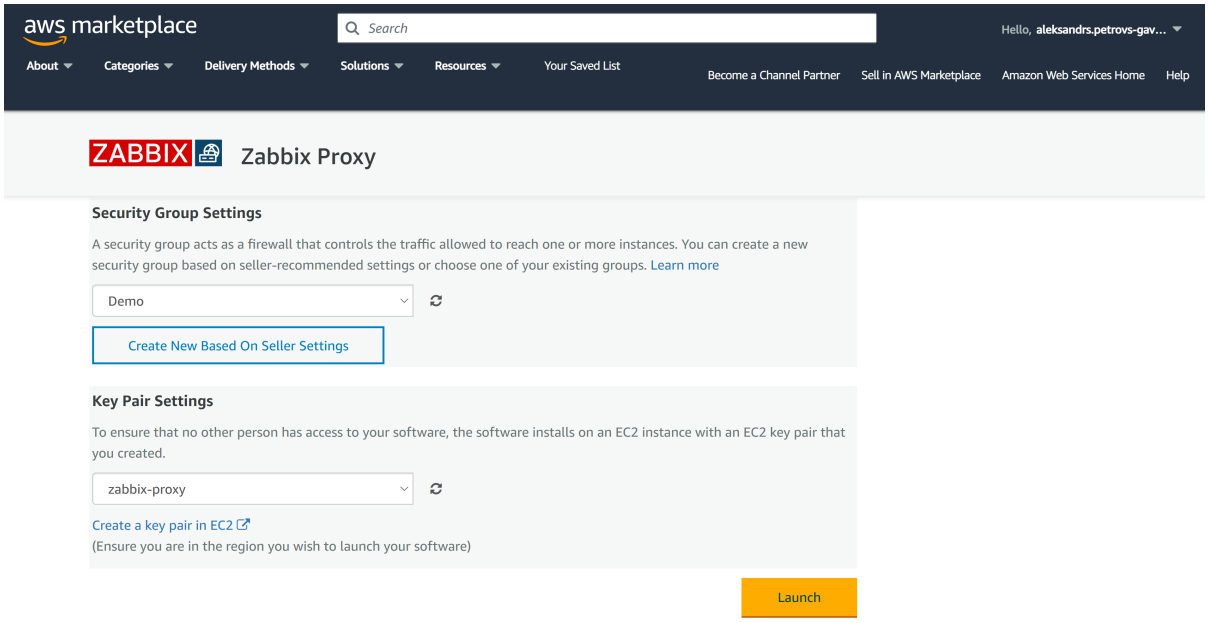
Sélectionnez l'option Fulfilment, la version de Zabbix et la région de votre choix.



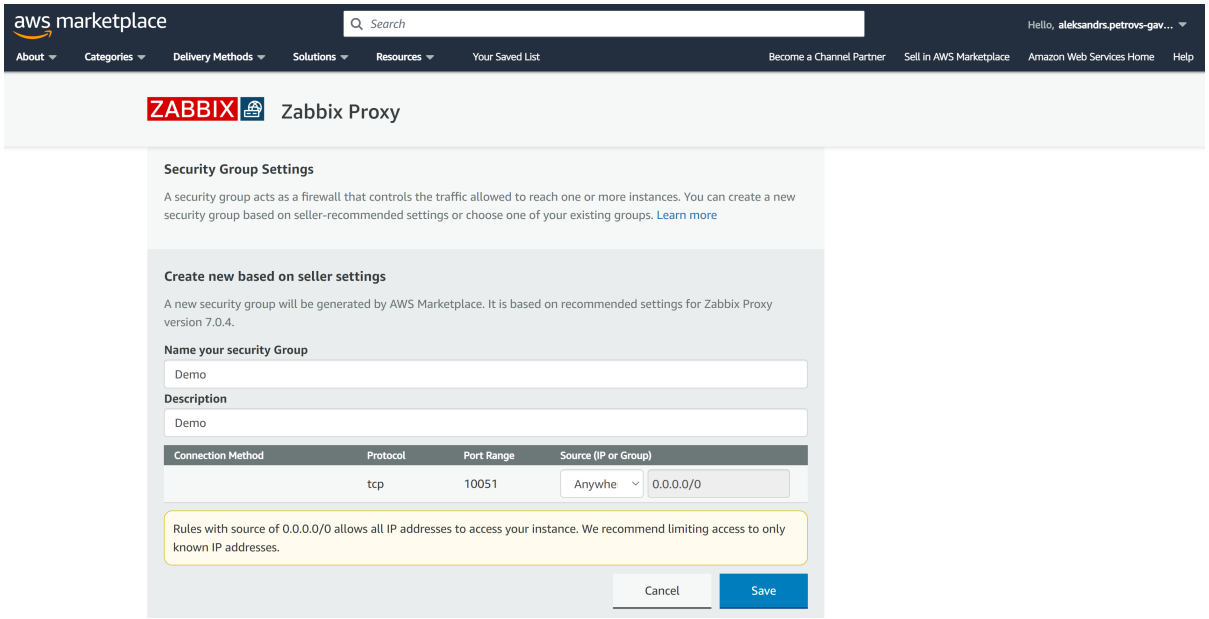
Cliquez sur le bouton *Continue to Launch*. Sur la page Launch, sélectionnez l'option Launch, le type d'instance EC2, le VPC et les paramètres de sous-réseau de votre choix.



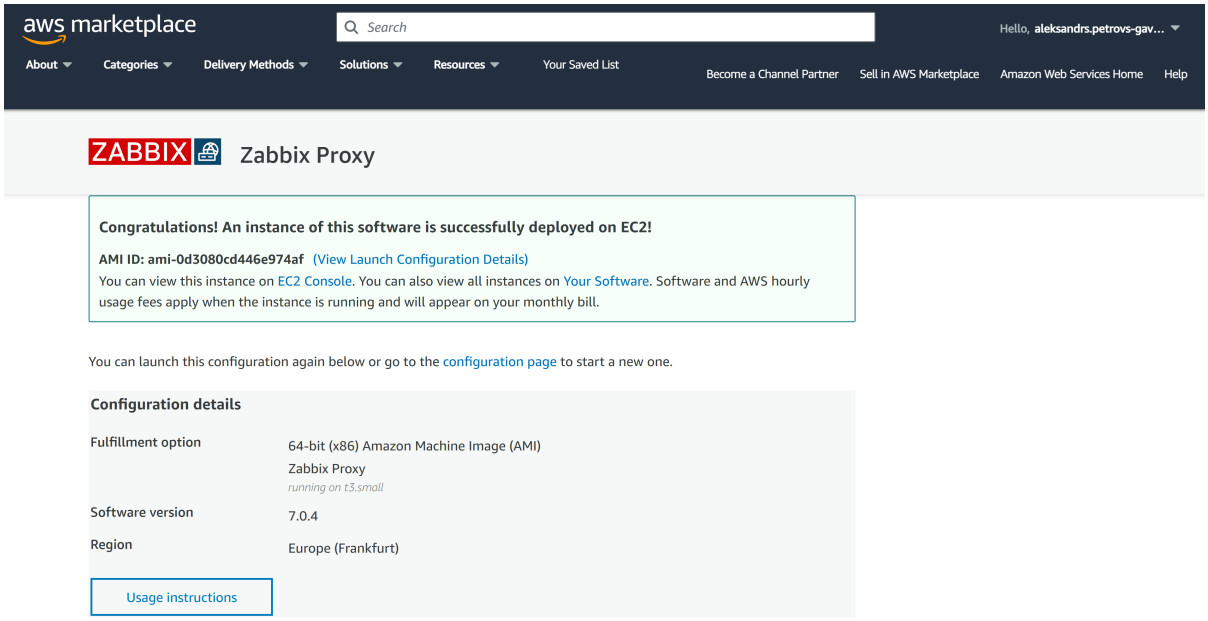
Au bas de la page Launch, indiquez les paramètres du groupe de sécurité et sélectionnez ou créez une paire de clés EC2. Veillez à enregistrer votre clé privée dans un emplacement sûr.



Si vous sélectionnez *Create a new Security Group based on the Seller settings*, le port TCP 10051 (serveur Zabbix) sera autorisé. Remarque : si vous souhaitez accéder à l'instance via SSH, vous devrez également autoriser les connexions SSH dans les paramètres de votre groupe de sécurité.

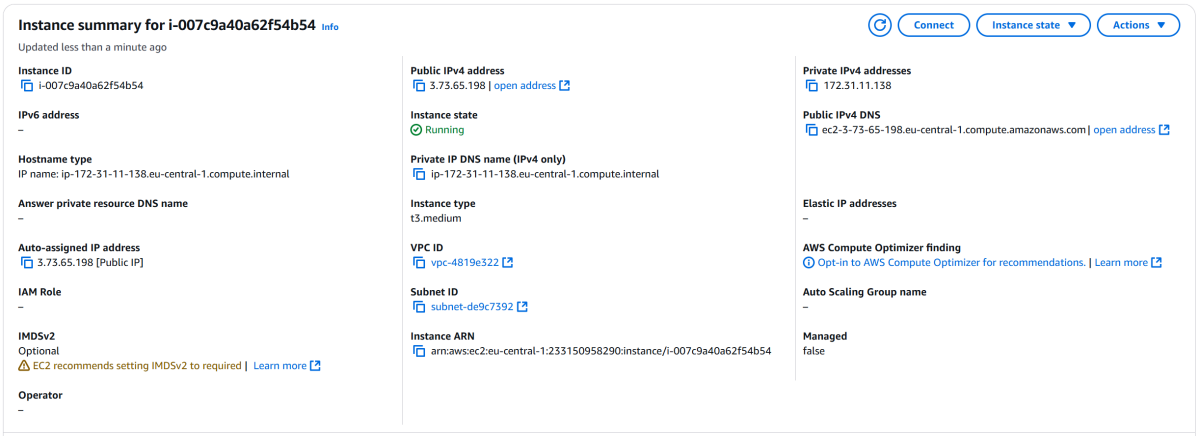


Enfin, cliquez sur le bouton *Launch*. Félicitations, votre proxy Zabbix est maintenant en cours de déploiement.



Configuration et accès au proxy Zabbix

Ouvrez la section Instances dans la console EC2 et sélectionnez l'instance du proxy Zabbix nouvellement déployée.



Connectez-vous à l'adresse IPv4 publique ou au DNS IPv4 public du proxy Zabbix à l'aide de SSH. Le nom d'utilisateur SSH par défaut est `rocky`. N'oubliez pas d'utiliser la paire de clés que vous avez générée précédemment.

La génération des identifiants peut prendre quelques minutes. Attendez quelques minutes avant d'essayer de vous connecter.



Le fichier de configuration du proxy Zabbix se trouve dans `/etc/zabbix/zabbix_proxy.conf`.

Le fichier de configuration peut être ouvert avec l'éditeur de texte de votre choix, par exemple vi :

```
sudo vi /etc/zabbix/zabbix_proxy.conf
```

La configuration initiale du proxy consiste à modifier les paramètres `ProxyMode`, `Server` et `Hostname` dans le fichier de configuration du proxy Zabbix.

Le mode du proxy peut être modifié en éditant le paramètre `ProxyMode`. Définissez `ProxyMode=0` pour le mode actif ou `ProxyMode=1` pour le mode passif.

Le paramètre `Server` doit contenir l'adresse IP du serveur Zabbix. Par exemple :

```
Server=192.168.50.141
```

Le paramètre `Hostname` doit contenir le nom d'hôte du proxy Zabbix. Le nom d'hôte défini ici doit correspondre au nom du proxy fourni lors de la configuration du proxy dans l'interface Zabbix. Par exemple :

```
Hostname=EC2-Frankfurt
```

Une fois la configuration terminée, redémarrez le proxy Zabbix à l'aide de la commande suivante :

```
sudo systemctl restart zabbix-proxy
```

Remarque : utilisez sudo pour modifier les fichiers de configuration et exécuter des commandes depuis la ligne de commande.

```
### Option: ProxyMode
#       Proxy operating mode.
#       0 - proxy in the active mode
#       1 - proxy in the passive mode
#
# Mandatory: no
# Default:
# ProxyMode=0

### Option: Server
#       If ProxyMode is set to active mode:
#               IP address or DNS name of Zabbix server to get configuration data from and send data to.
#       If ProxyMode is set to passive mode:
#               List of comma delimited IP addresses, optionally in CIDR notation, or DNS names of Zabbix
server.
#               Incoming connections will be accepted only from the addresses listed here.
#               If IPv6 support is enabled then '127.0.0.1', '::127.0.0.1', '::ffff:127.0.0.1' are treated
 equally
#               and '::/0' will allow any IPv4 or IPv6 address.
#               '0.0.0.0/0' can be used to allow any IPv4 address.
#               Example: Server=127.0.0.1,192.168.1.0/24,::1,2001:db8::/32,zabbix.example.com
#
# Mandatory: yes
# Default:
# Server=

Server=127.0.0.1
```

```
### Option: Hostname
#       Unique, case sensitive Proxy name. Make sure the Proxy name is known to the server!
#       Value is acquired from HostnameItem if undefined.
#
# Mandatory: no
# Default:
# Hostname=

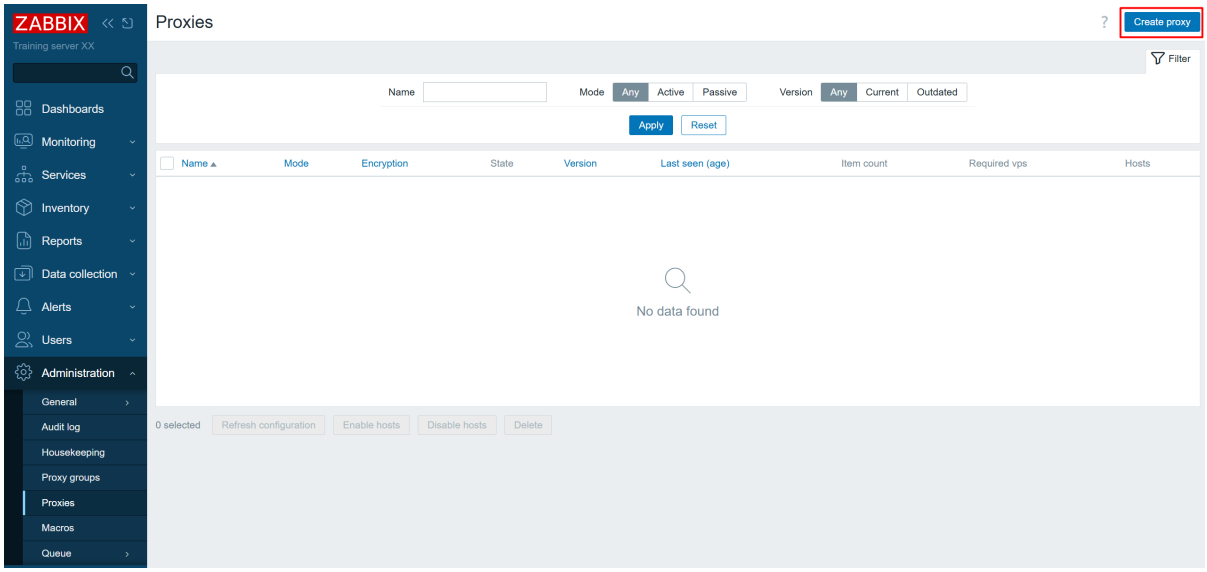
Hostname=EC2-Frankfurt
```

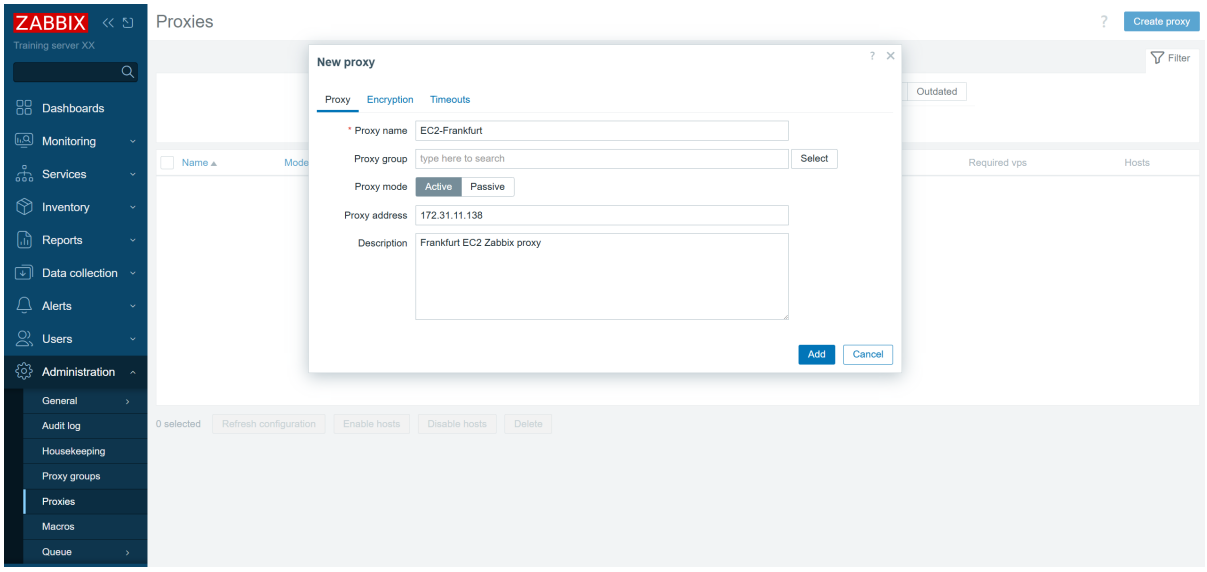
Ouvrez l'interface Zabbix et accédez à *Administration* > *Proxies*. Cliquez sur le bouton *Create proxy*.

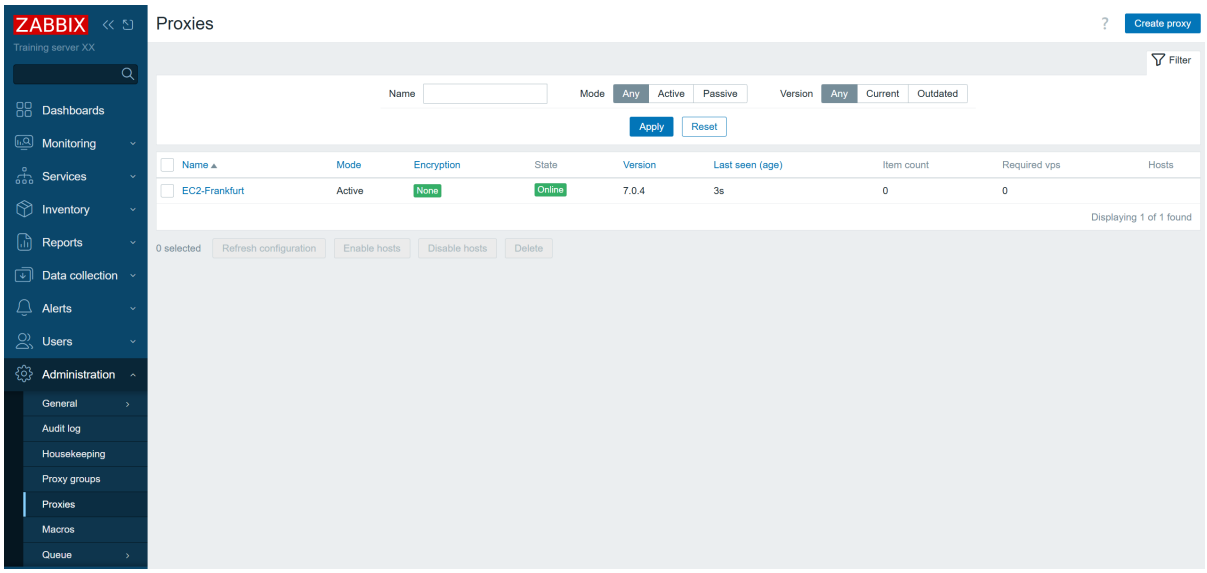
Saisissez le même nom de proxy que la valeur du paramètre `Hostname` dans le fichier de configuration du proxy.

Sélectionnez le mode du proxy et indiquez l'adresse IP du proxy (facultatif pour les proxies actifs).

Cliquez sur *Create proxy*.







Emplacement de la base de données du proxy

La base de données SQLite du proxy est stockée dans `/tmp/zabbix_proxy.sqlite3` et est spécifiée par le paramètre `DBName` dans le fichier de configuration du proxy Zabbix.

Guide de déploiement d'Azure pour le serveur Zabbix

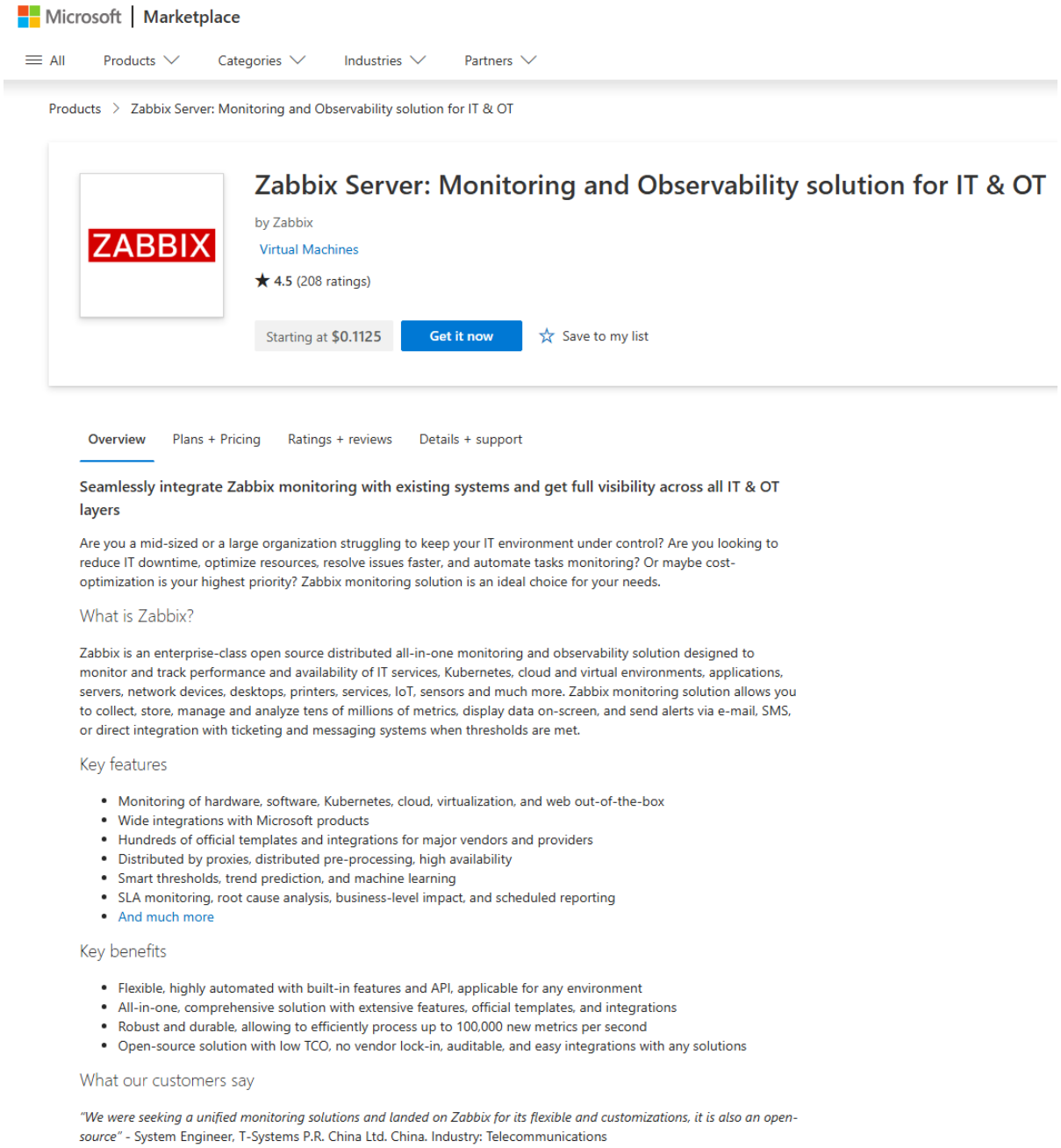
Présentation générale

Cette page décrit comment déployer le serveur Zabbix à l'aide d'Azure.

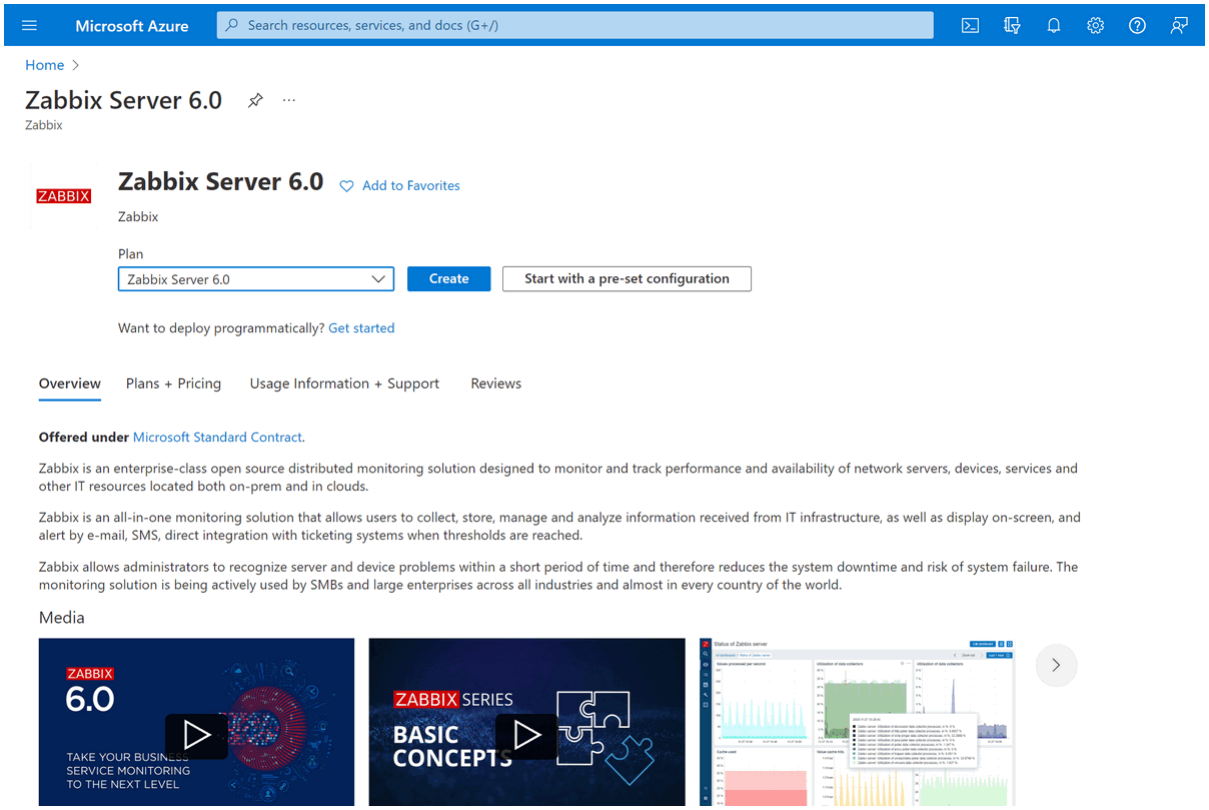
Attention:
Plusieurs versions de Zabbix sont disponibles pour le déploiement sur la plateforme. Veillez à consulter la documentation correspondant à la version de Zabbix que vous avez déployée. La documentation des différentes versions de Zabbix peut être sélectionnée dans le menu déroulant des versions.

Déploiement de l'appliance

Pour déployer l'appliance Zabbix, accédez d'abord à la page Zabbix Azure Marketplace. Consultez la présentation, puis cliquez sur *Get it now*.



Consultez la présentation du déploiement, les plans tarifaires et les autres détails. Cliquez ensuite sur *Create* ou *Start with a pre-set configuration* pour déployer votre appliance Zabbix.

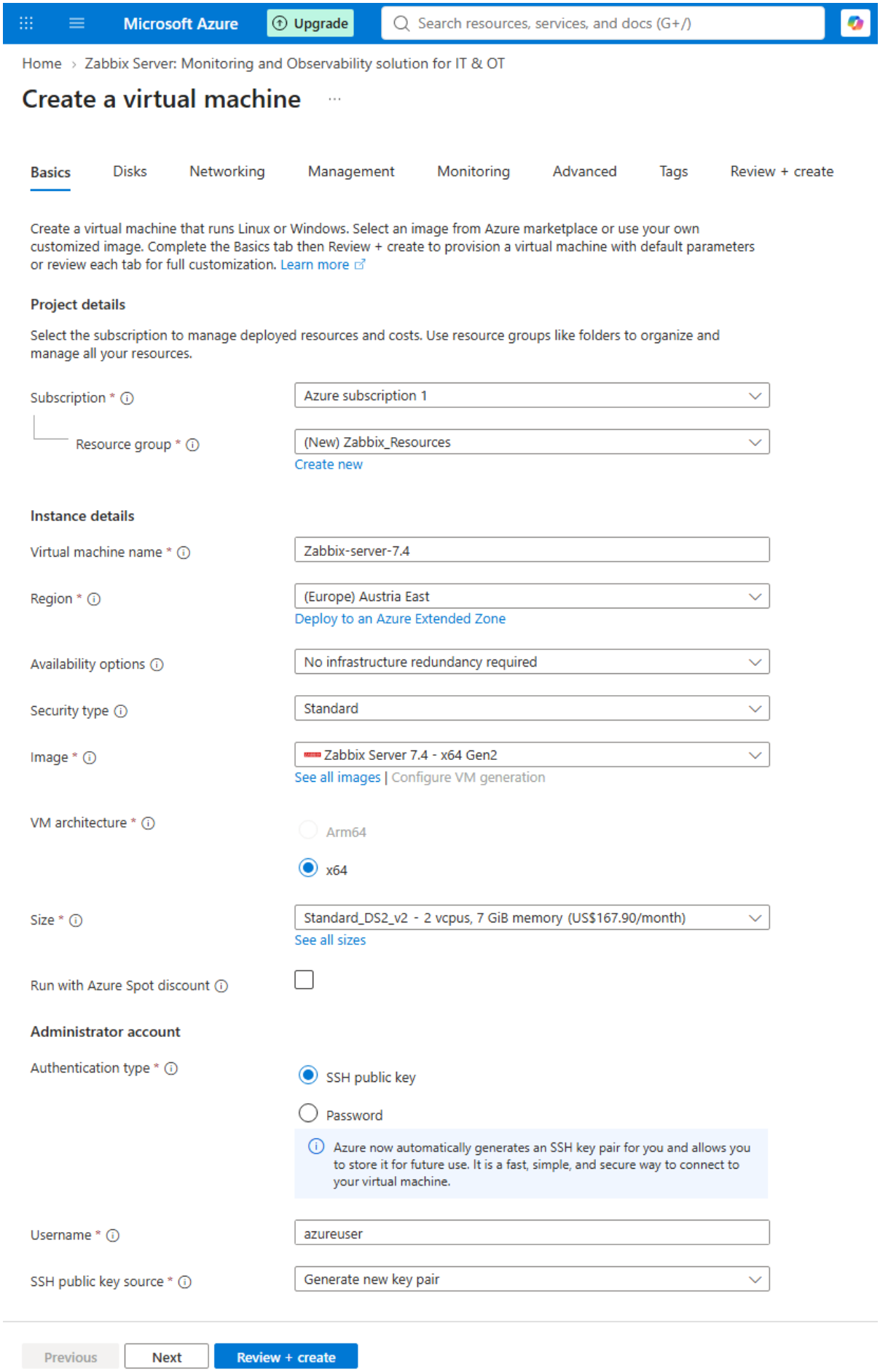


Configurez la VM en sélectionnant sa taille, en créant un compte administrateur et en définissant les paramètres du disque, du réseau, de la gestion et autres. Ces paramètres peuvent varier selon les exigences de votre infrastructure.

Vous pouvez utiliser la configuration suivante comme exemple :

Onglet *Basics* :

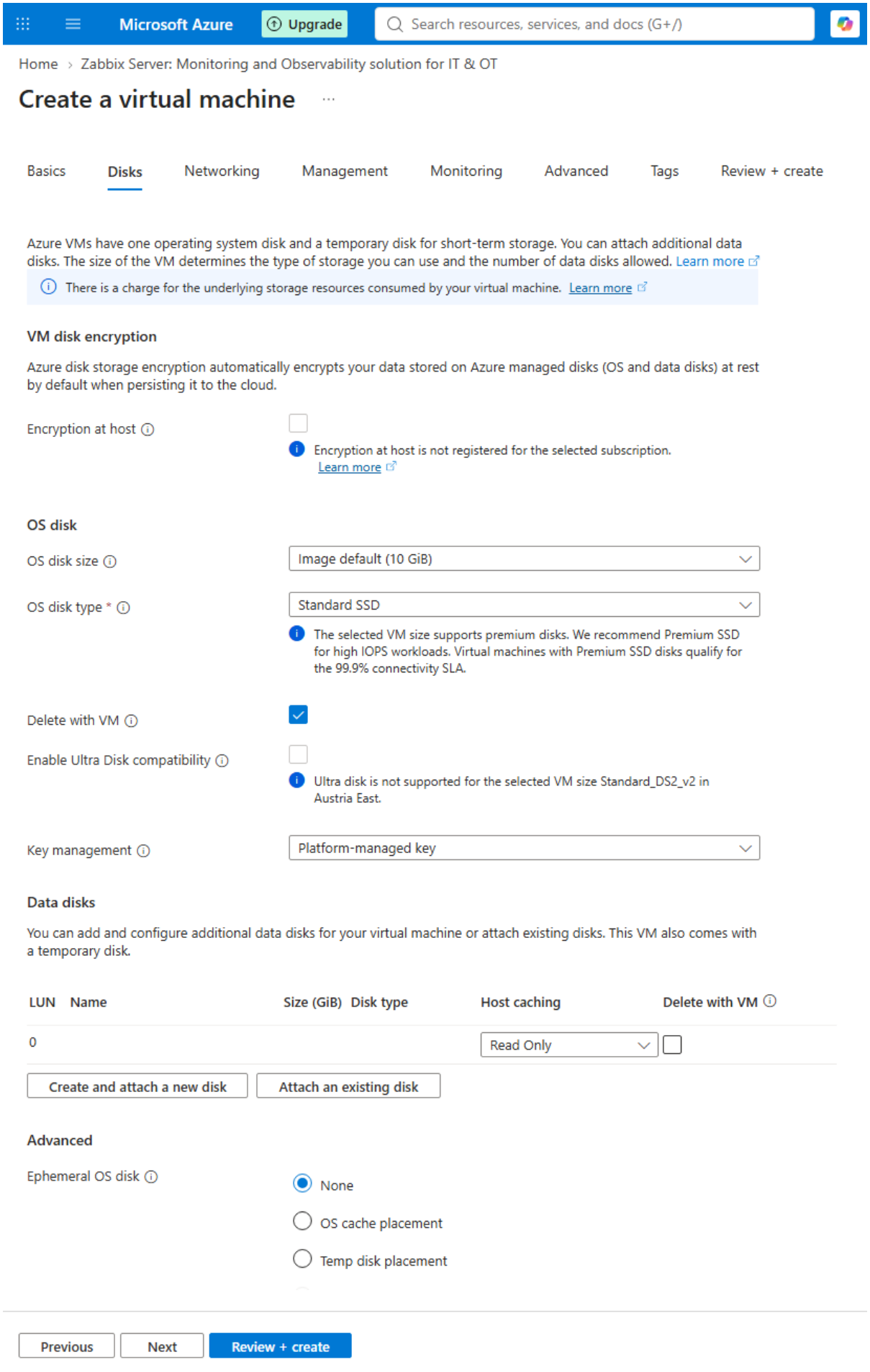
- *Subscription* : Azure subscription 1
- *Resource group* : Zabbix_Resources
- *Virtual machine name* : Zabbix-server--7.4
- *Availability options* : No infrastructure redundancy required
- *Security type* : Standard
- *Image* : Zabbix Server 7.4 - x64 Gen2
- *Size* : Standard_DS2_V2 - 2 vcpus, 7 GiB memory
- *Authentication type* : SSH public key
- *Username* : azureuser
- *SSH public key source* : Generate new key pair



Onglet *Disks* :

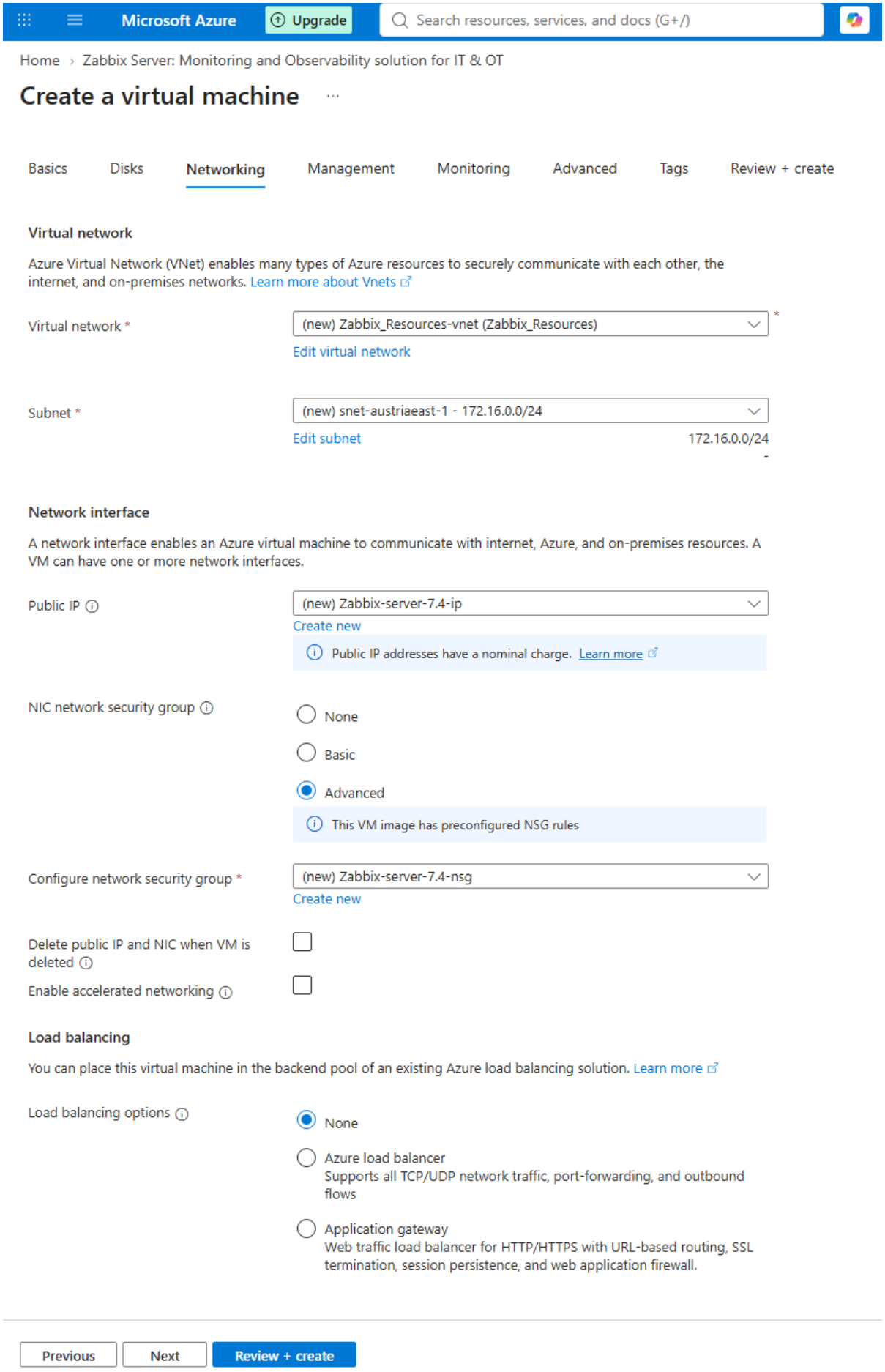
- *Encryption at host* : décoché
- *OS disk type* : Standard SSD

- *Delete with VM* : coché
- *Enable Ultra Disk compatibility* : décoché

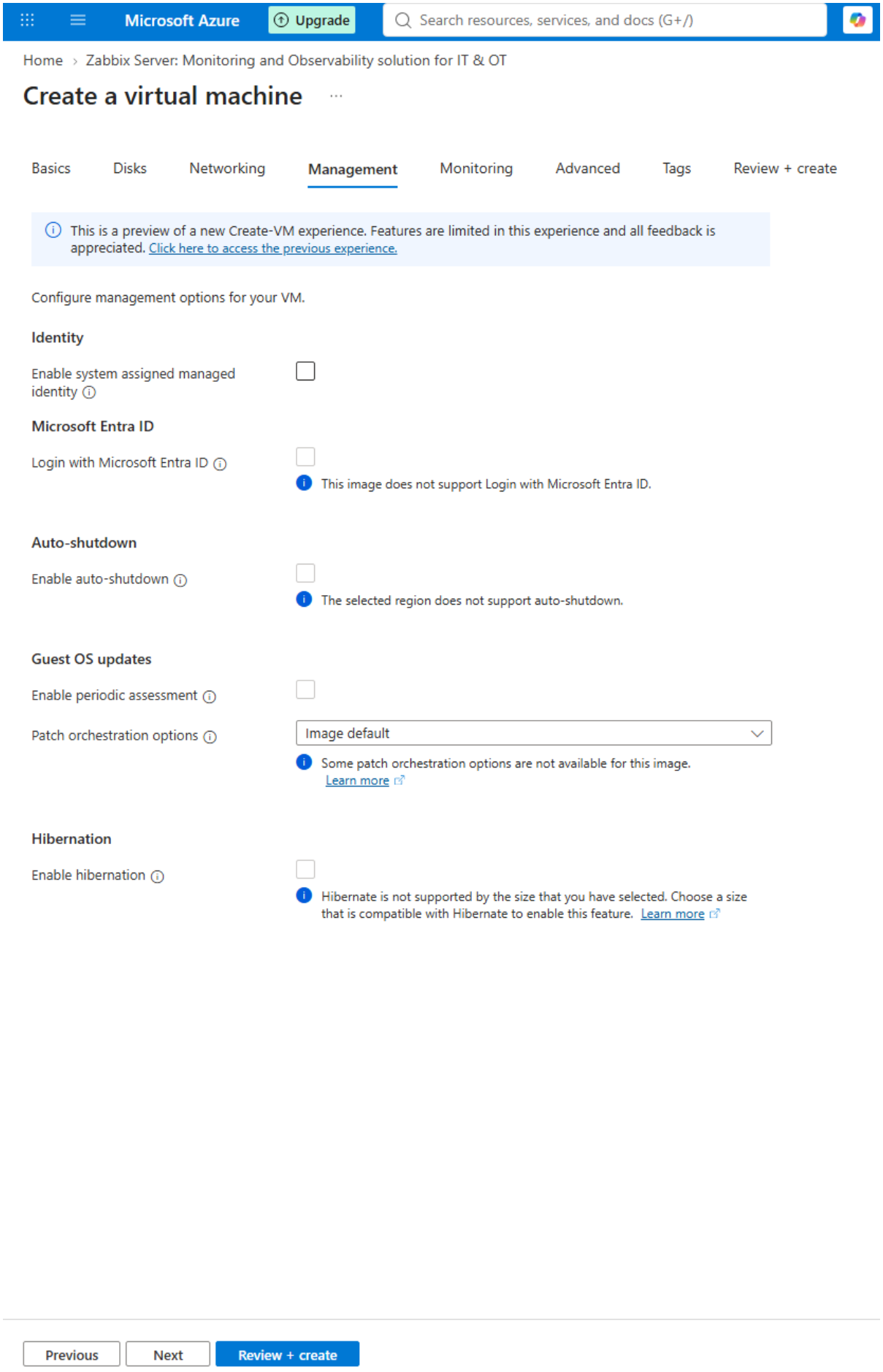


Onglet *Networking* :

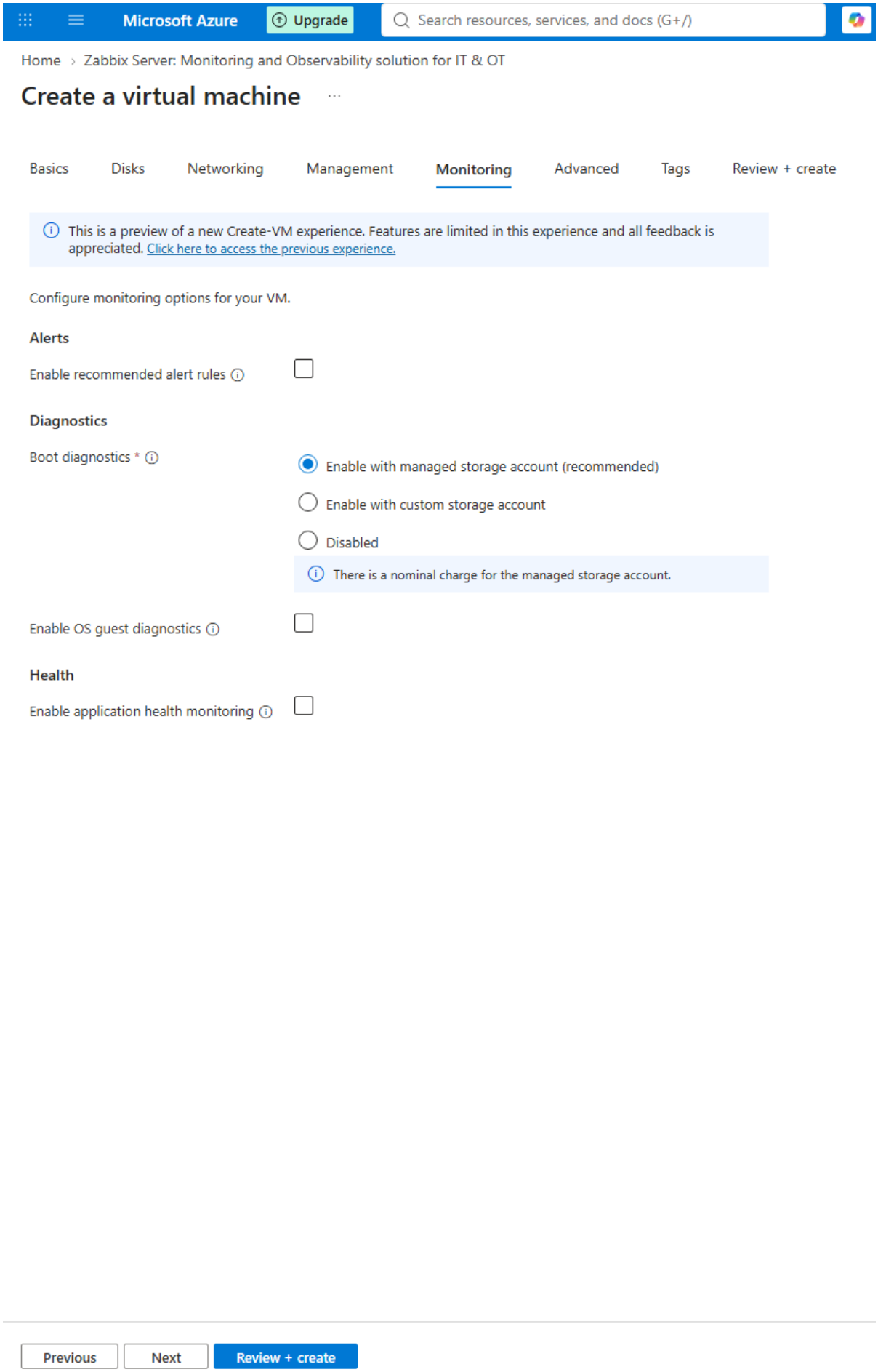
- *Virtual network* : Zabbix_Resources-vnet
- *Public IP* : Zabbix-server-7.4-ip
- *NIC network security group* : Advanced
- *Configure network security group* : Zabbix-server-7.4-nsg



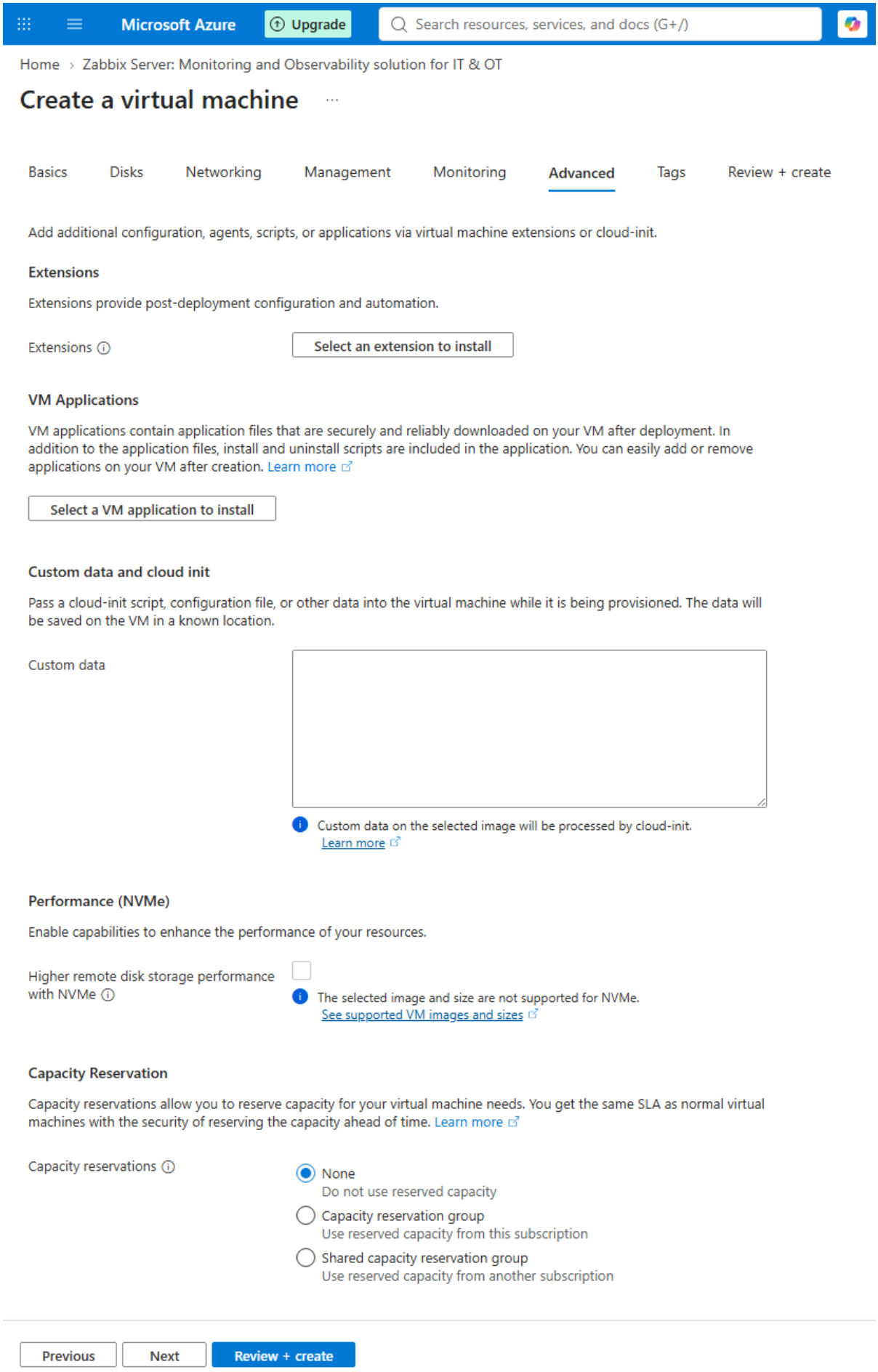
Onglet *Management* : les paramètres peuvent être laissés par défaut.



Onglet *Monitoring* : les paramètres peuvent être laissés par défaut.



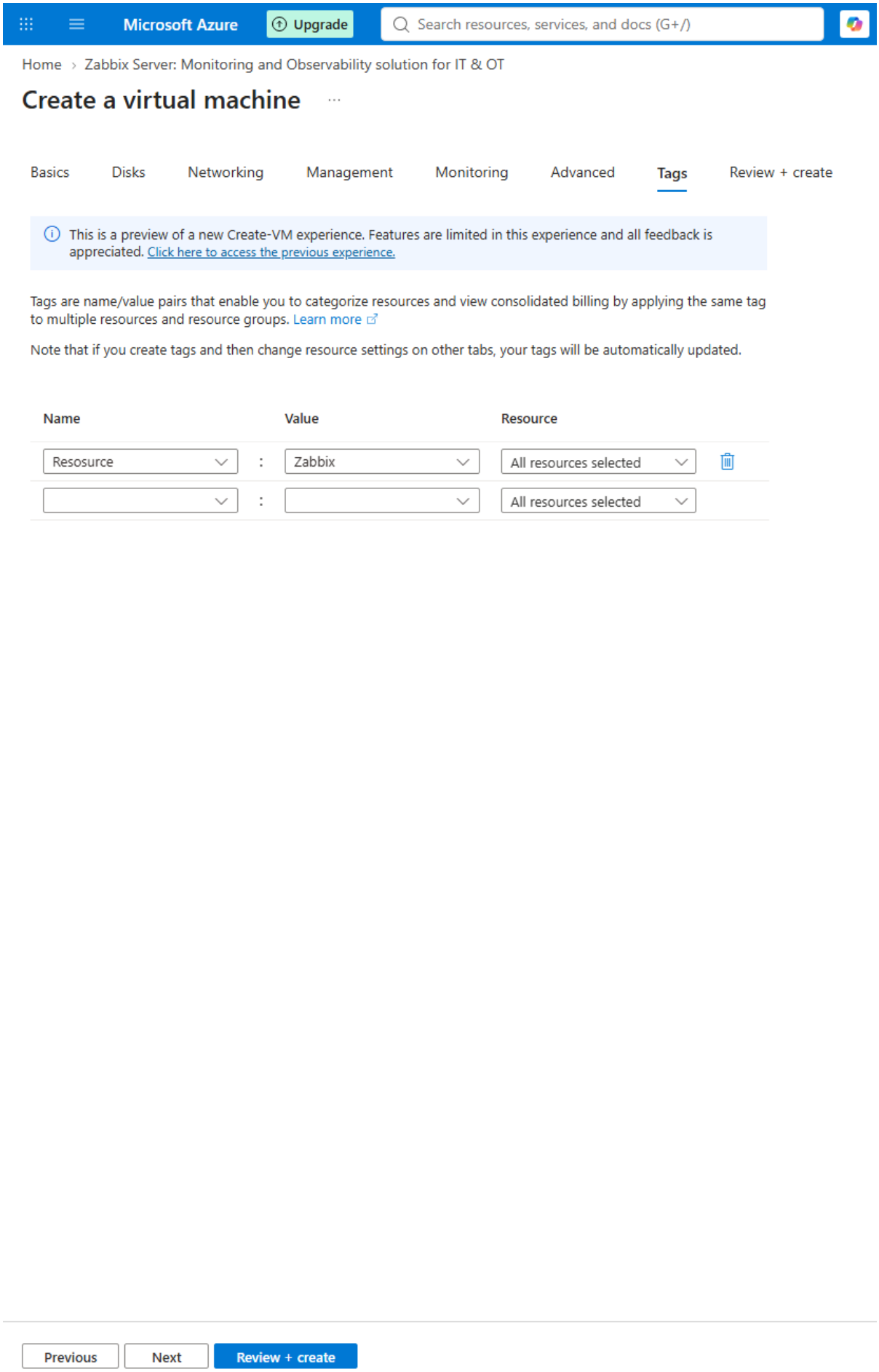
Onglet *Advanced* : les paramètres peuvent être laissés par défaut.



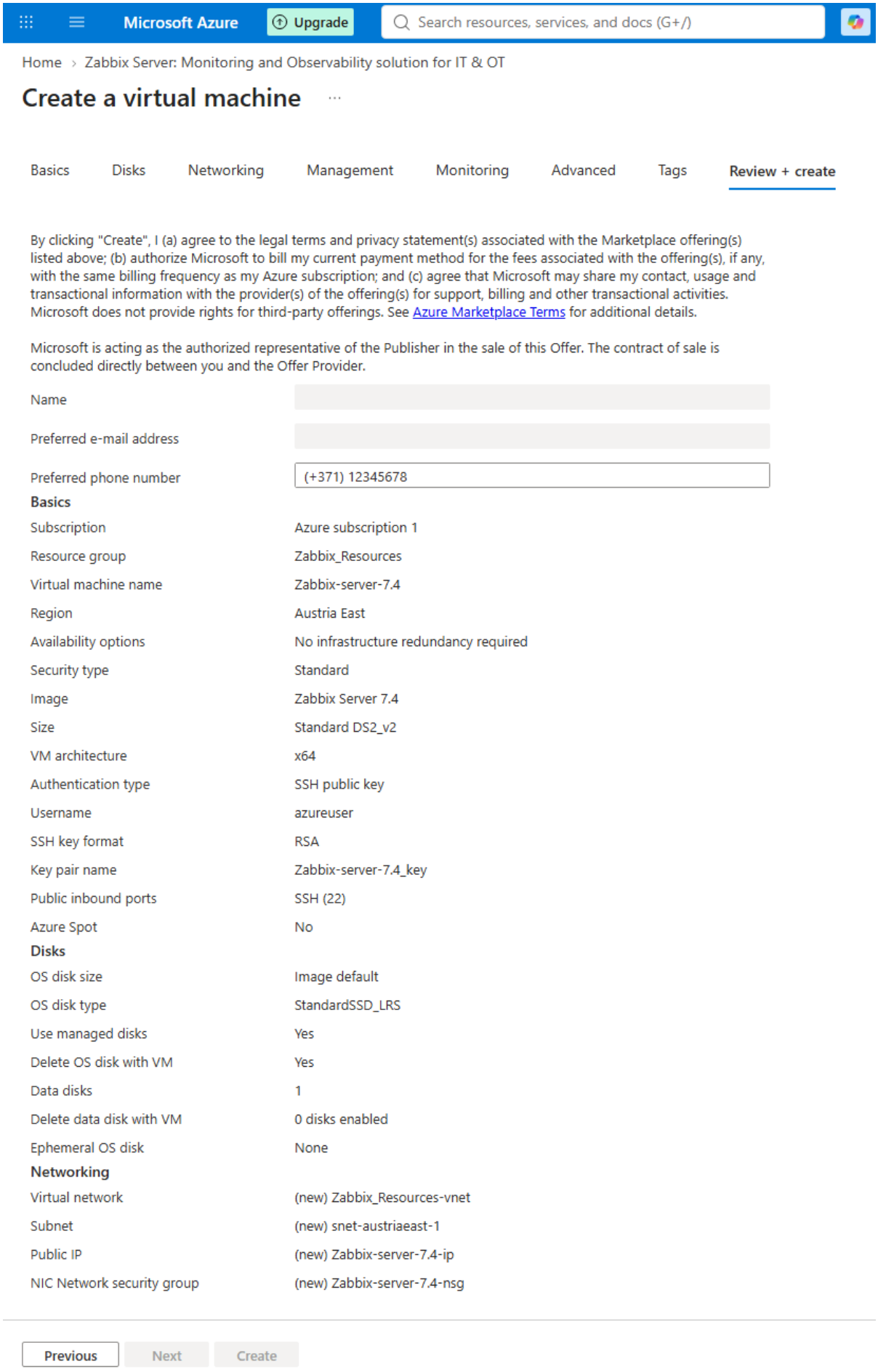
Onglet *Tags* :

- *Name* : Resource
- *Value* : Zabbix

- *Resource* : All resource selected



Confirmez les détails de la VM et créez la VM.



Vous serez redirigé vers la page d'état du déploiement de la VM. Attendez que le déploiement soit terminé.

Une fois le déploiement terminé, ouvrez la ressource et notez son adresse IP publique. Vous devrez maintenant vous connecter au serveur Zabbix à l'aide de SSH (ou d'une autre méthode disponible) afin d'obtenir les identifiants de connexion à l'interface

Zabbix.

```
********************************************************************************

Zabbix frontend credentials:

Username: Admin

Password: zfpvagcs0s+srUUK+EM

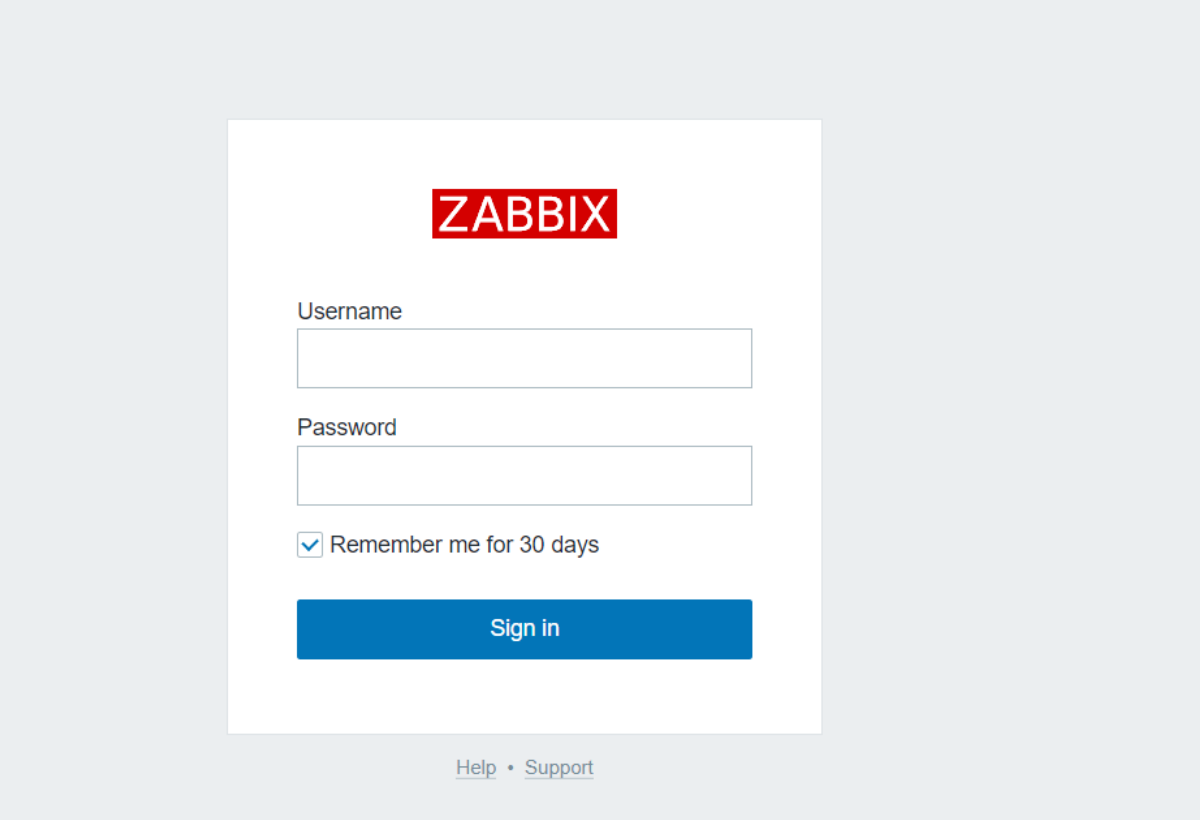

To learn about available professional services, including technical suppport and
 training, please visit https://www.zabbix.com/services

Official Zabbix documentation available at https://www.zabbix.com/documentation/
current/

Note! Do not forget to change timezone PHP variable in /etc/php-fpm.d/zabbix.con
f file.


********************************************************************************
[azureuser@Zabbix-server-6 ~]$
```

Accédez à l'interface Zabbix en ouvrant l'adresse IP publique de l'appliance Zabbix dans un navigateur web. Pour vous connecter, utilisez le nom d'utilisateur et le mot de passe obtenus à l'étape précédente.



Accéder à la base de données

Vous pouvez accéder à la base de données Zabbix avec l'utilisateur `root` sans saisir de mot de passe. Les identifiants sont stockés dans le fichier de configuration `/root/.my.cnf`.

Guide de déploiement d'Azure pour le proxy Zabbix

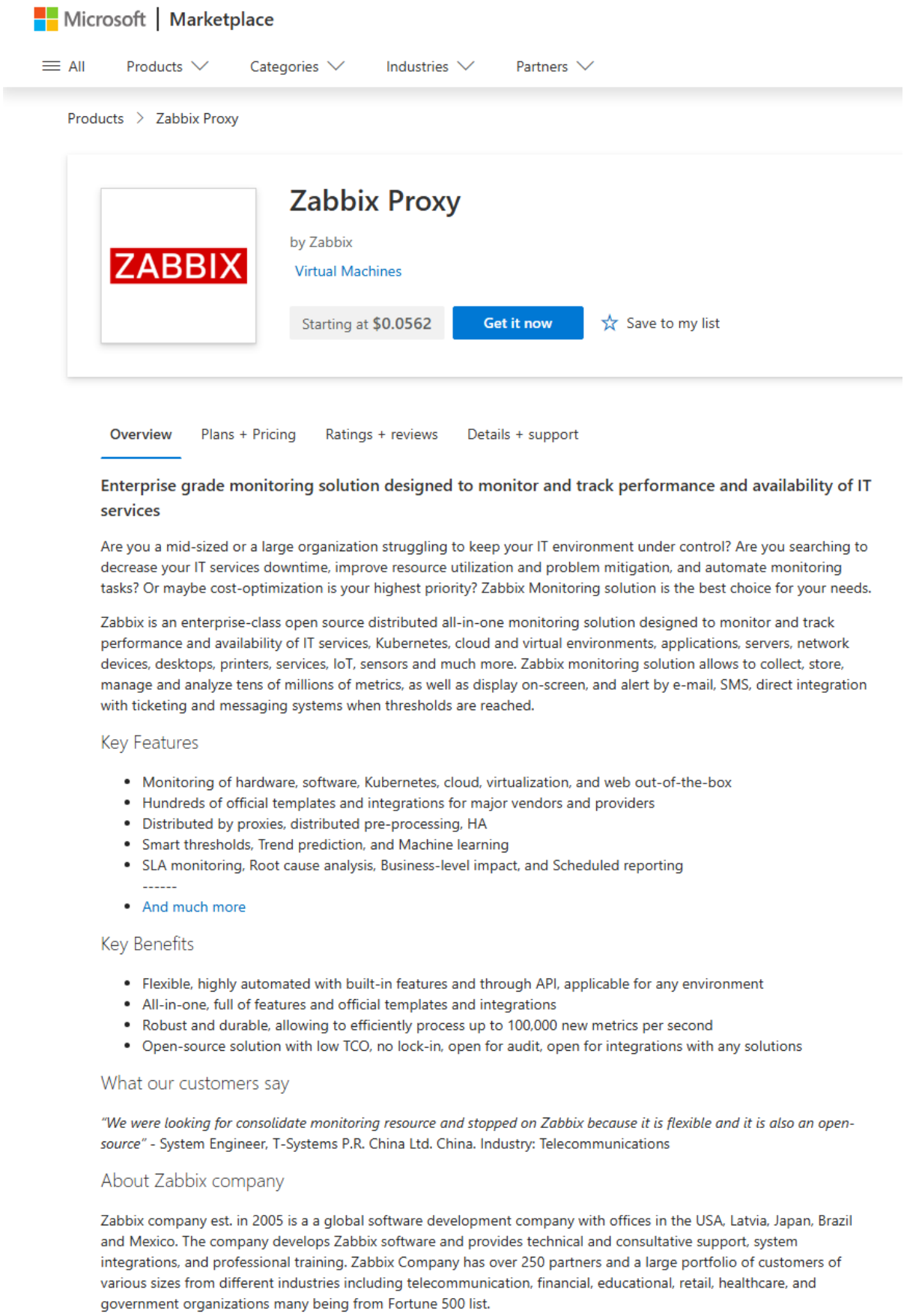
Présentation générale

Cette page décrit comment déployer un proxy Zabbix à l'aide d'Azure.

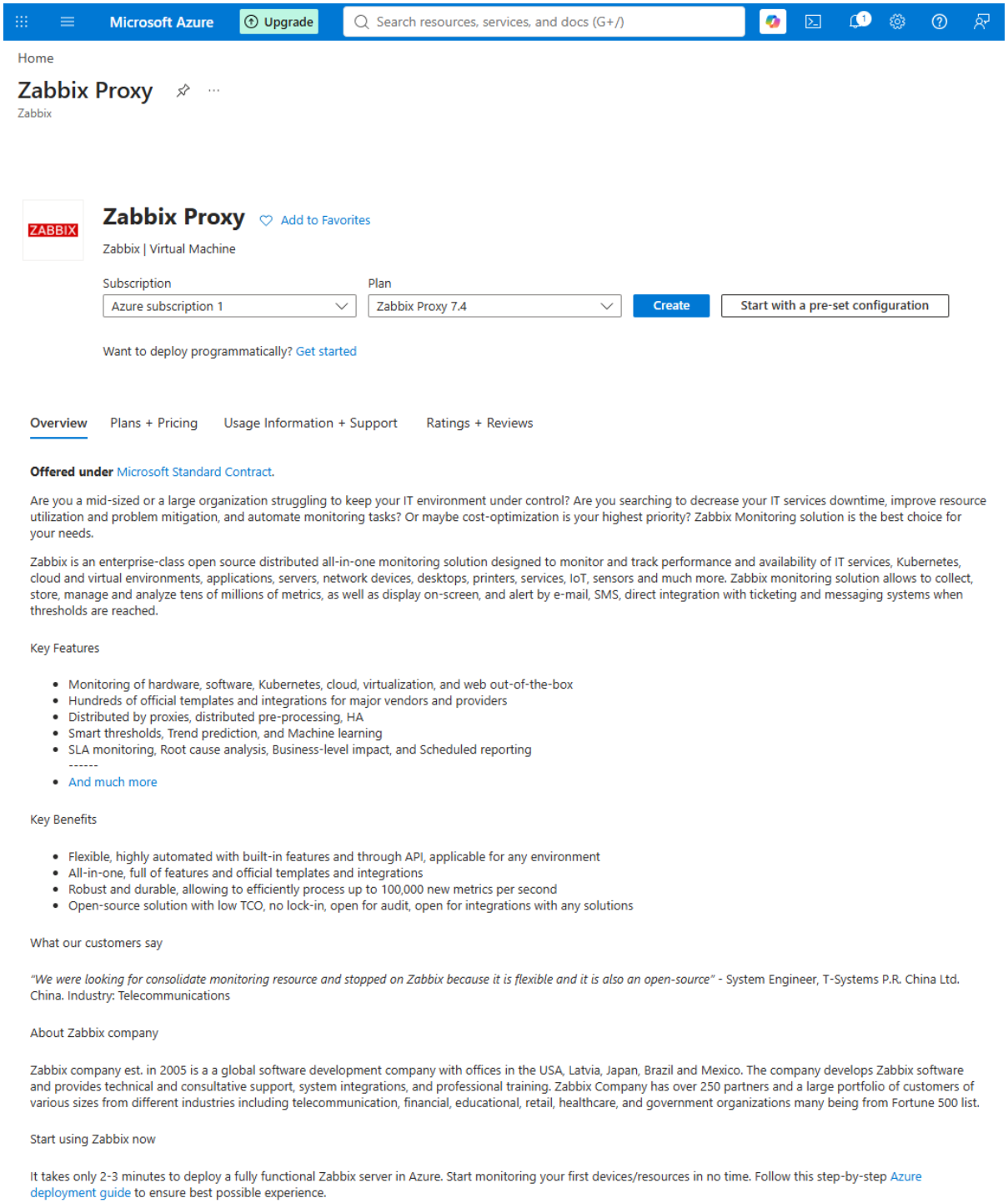
Attention:
Plusieurs versions de Zabbix sont disponibles pour le déploiement sur la plateforme. Veillez à consulter la documentation correspondant à la même version de Zabbix que celle que vous avez déployée. La documentation des différentes versions de Zabbix peut être sélectionnée dans le menu déroulant des versions.

Déploiement d'un proxy Zabbix

Pour déployer un proxy Zabbix, accédez d'abord à la page Zabbix sur la marketplace Azure. Cliquez sur *Get it now*.



Vous pouvez consulter ici des informations supplémentaires sur le déploiement actuel, telles que la présentation générale, les plans tarifaires, etc. Cliquez sur *Create* ou sur *Start with a pre-set configuration*.

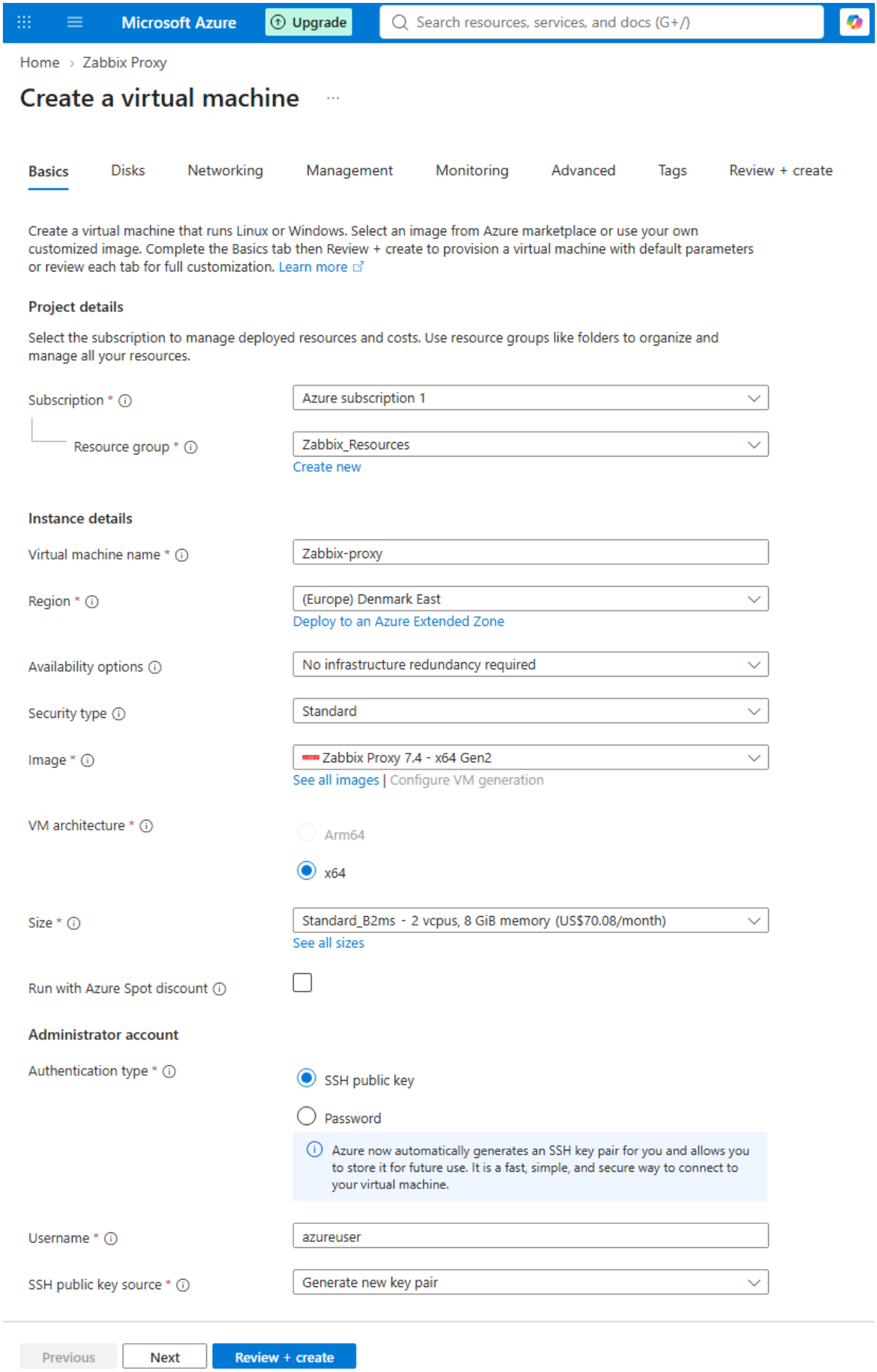


Procédez à la configuration détaillée de la VM : sélectionnez la taille de la VM, le compte administrateur, le disque, le réseau, la gestion et les autres paramètres. Ces paramètres peuvent varier en fonction des exigences de votre infrastructure.

Vous pouvez utiliser la configuration suivante comme exemple :

Onglet *Basics* :

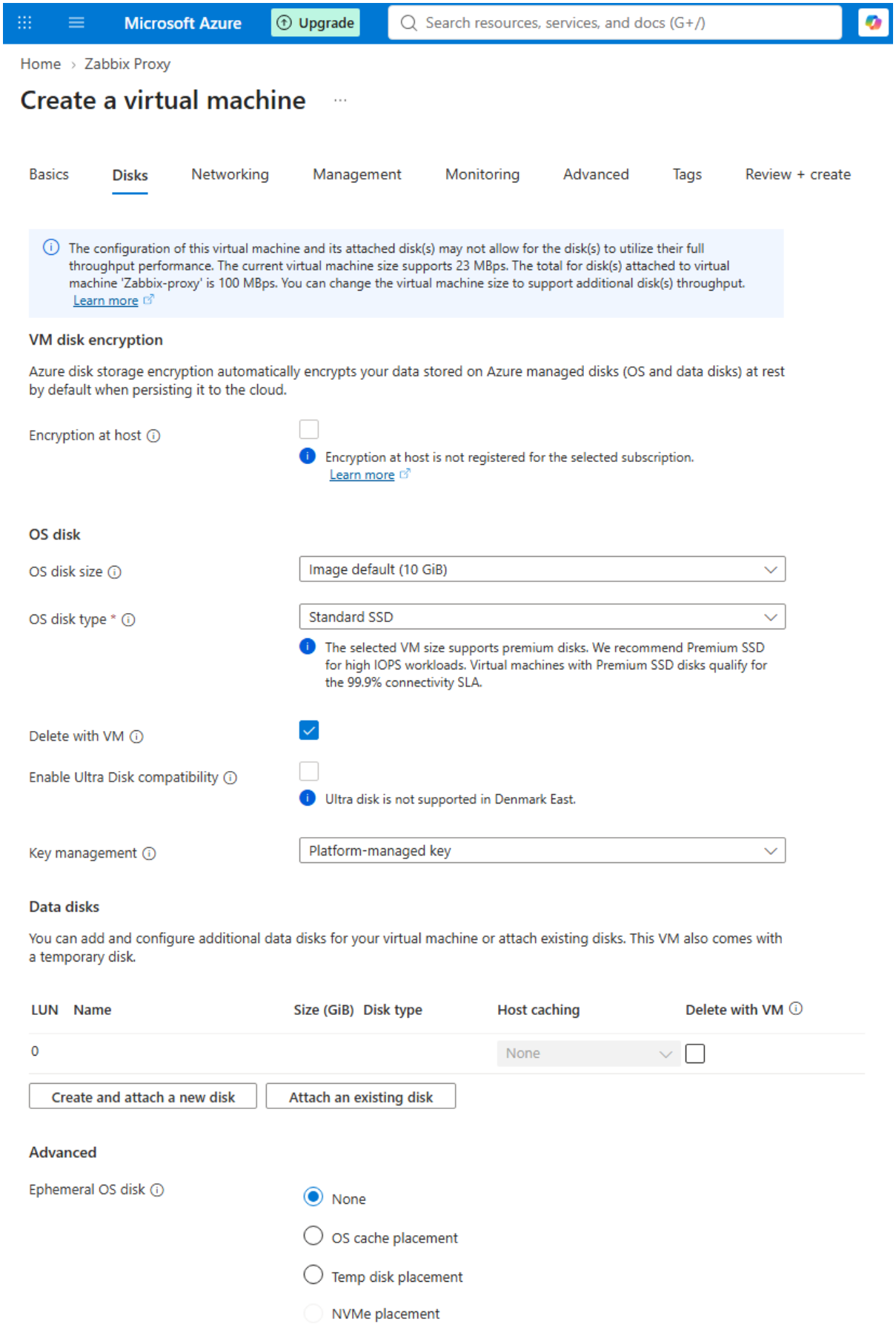
- *Subscription* : Azure subscription 1
- *Resource group* : Zabbix_Resources
- *Virtual machine name* : Zabbix-proxy
- *Availability options* : No infrastructure redundancy required
- *Security type* : Standard
- *Image* : Zabbix Proxy 7.4 - x64 Gen2
- *Size* : Standard_B2ms - 2 vcpus, 8 GiB memory
- *Authentication type* : SSH public key
- *Username* : azureuser
- *SSH public key source* : Generate new key pair



Onglet *Disks* :

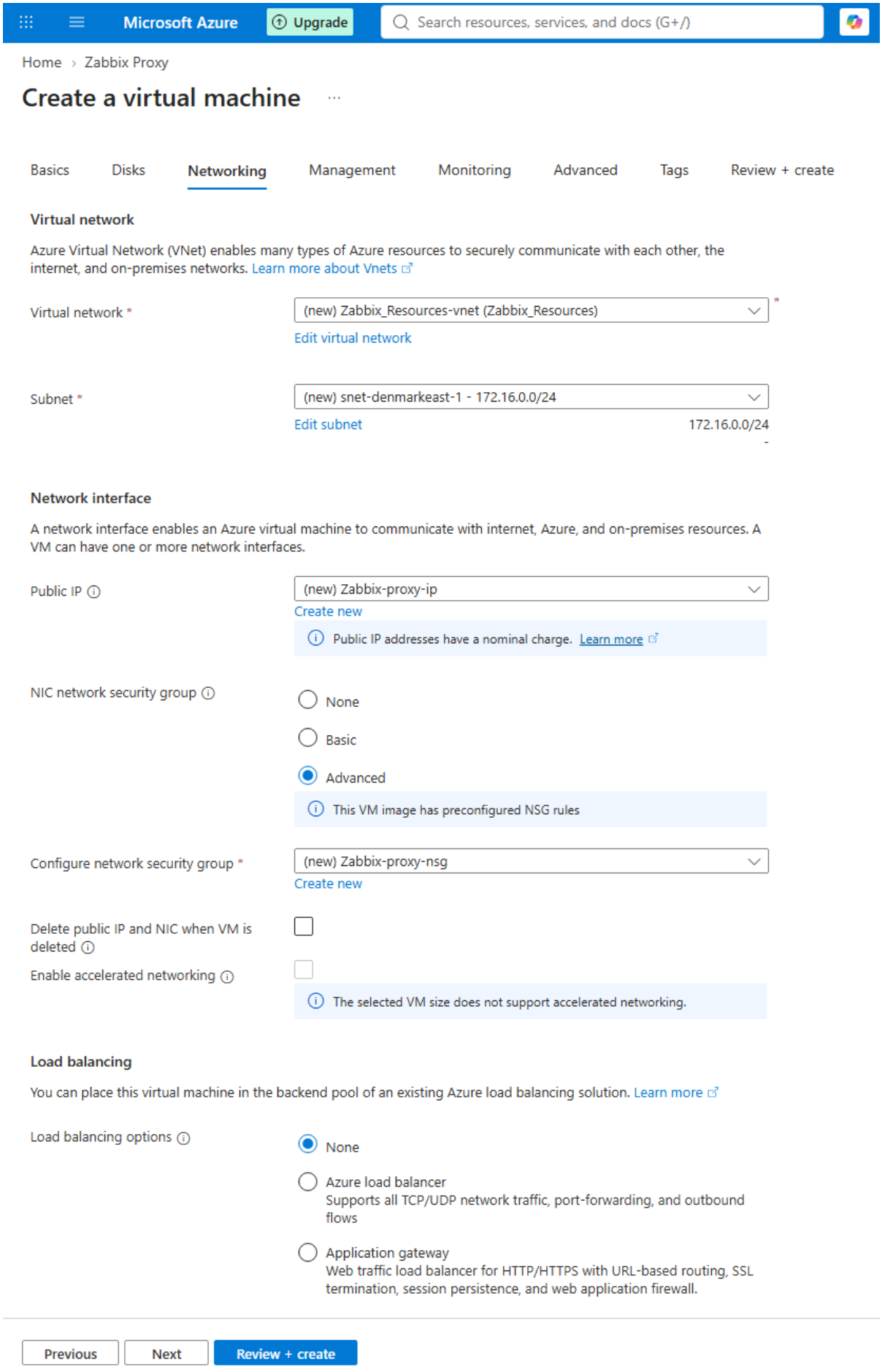
- *Encryption at host* : décoché

- *OS disk type* : Standard SSD
- *Delete with VM* : coché
- *Enable Ultra Disk compatibility* : décoché

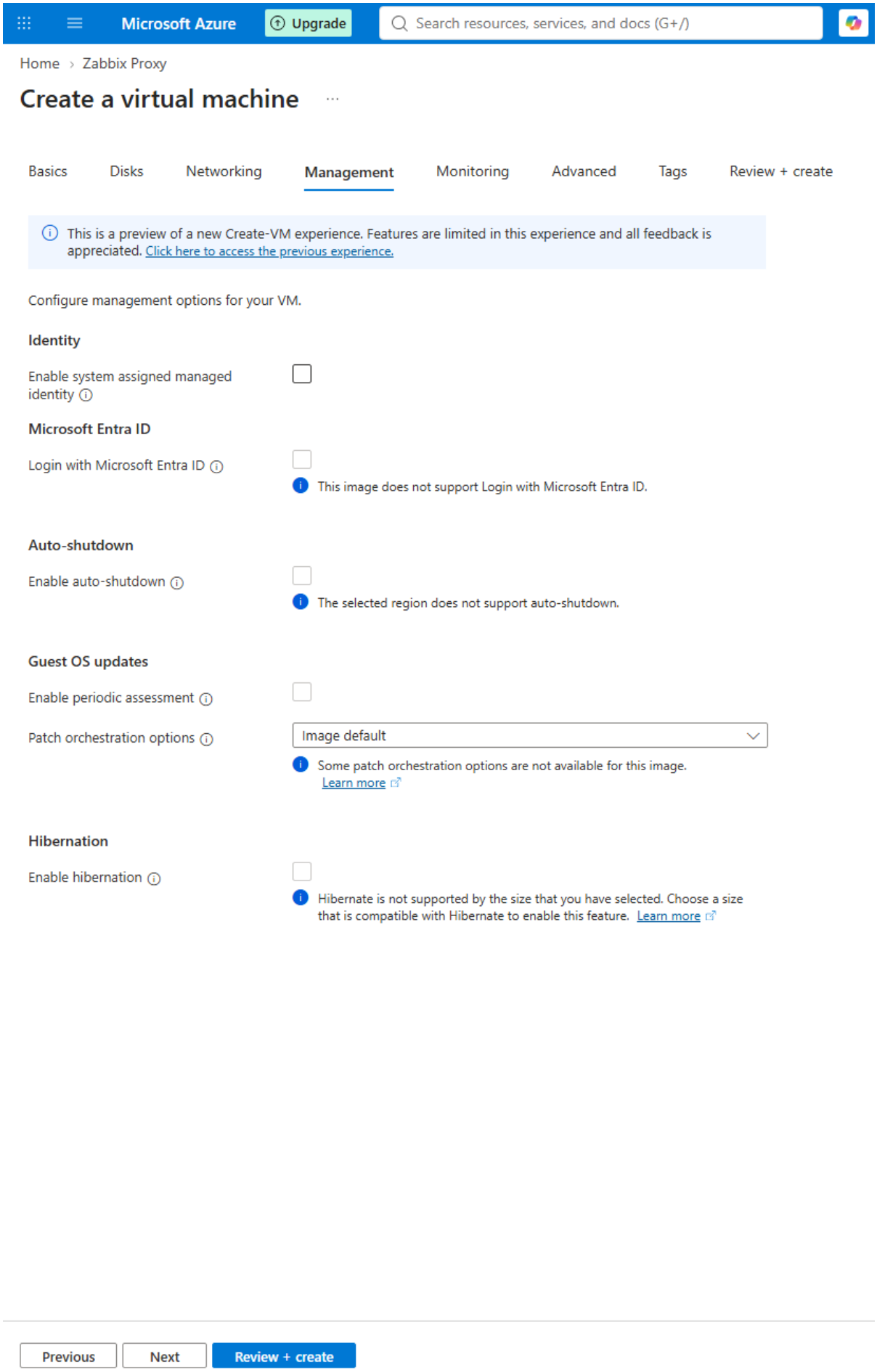


Onglet *Networking* :

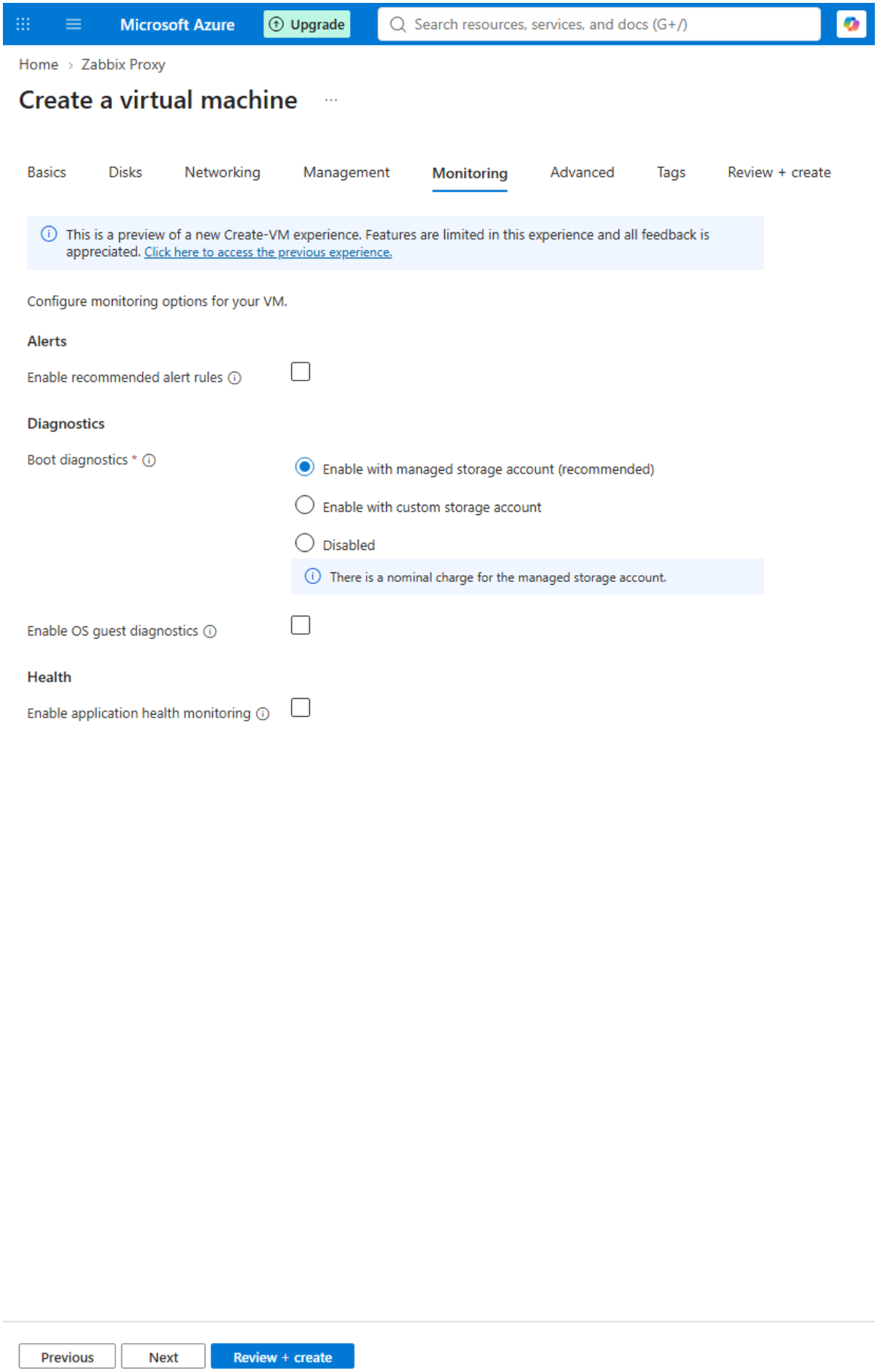
- *Virtual network* : Zabbix_Resources-vnet
- *Public IP* : Zabbix-proxy-ip
- *NIC network security group* : Advanced
- *Configure network security group* : Zabbix-proxy-nsg



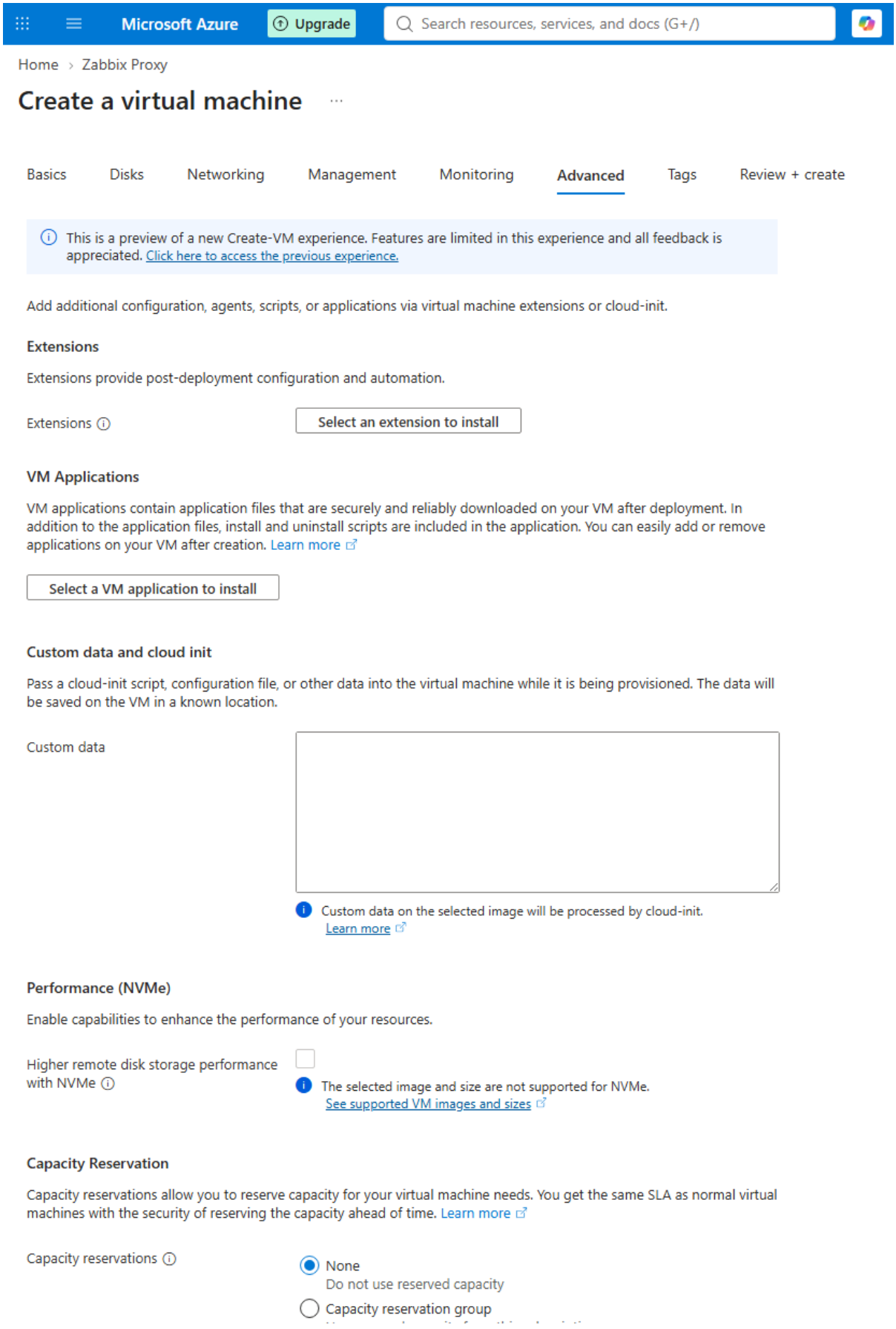
Onglet *Management* : les paramètres peuvent être conservés par défaut.



Onglet *Monitoring* : les paramètres peuvent être conservés par défaut.



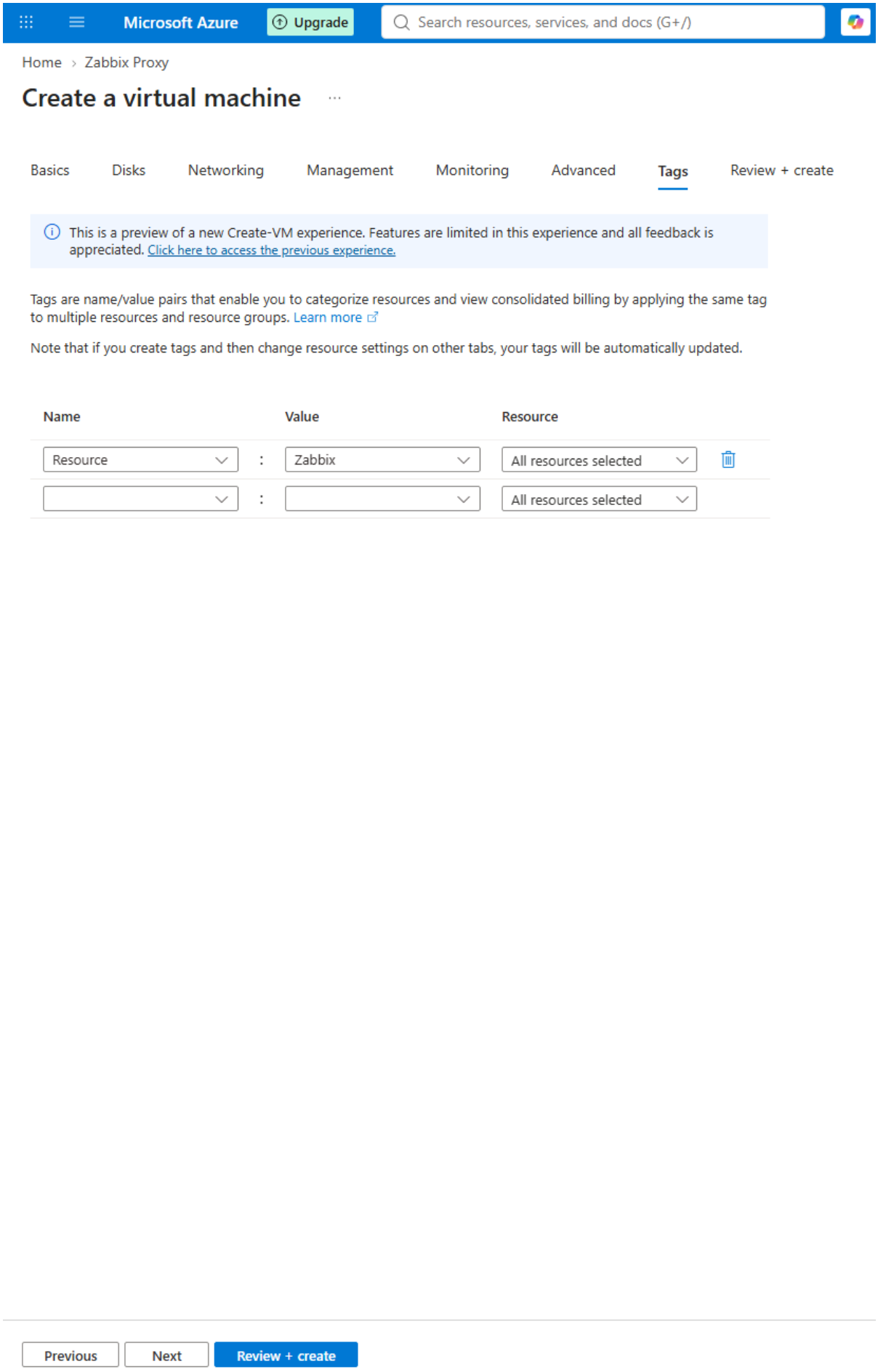
Onglet *Advanced* : les paramètres peuvent être conservés par défaut.



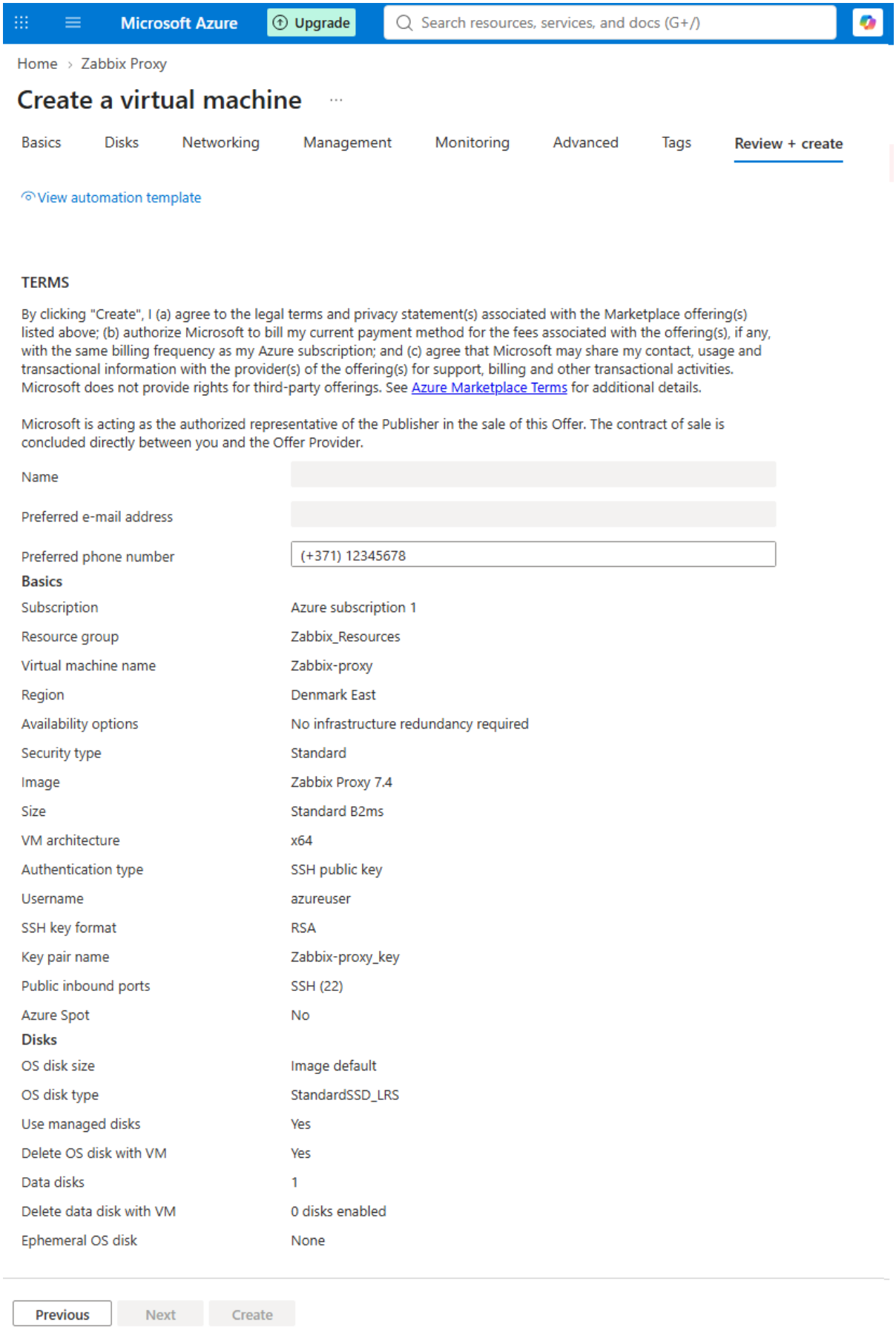
Onglet *Tags* :

- *Name* : Resource
- *Value* : Zabbix

- *Resource* : All resource selected



Confirmez les détails de la VM et créez la VM.



Vous serez redirigé vers la page d'état du déploiement de la VM. Attendez que le déploiement soit terminé.

Une fois le déploiement terminé, ouvrez la ressource et notez son adresse IP publique. Connectez-vous au proxy Zabbix via SSH, en utilisant les identifiants administrateur créés lors du déploiement de la VM, afin de configurer le proxy.

```
********************************************************************************

To learn about available professional services, including technical suppport and
 training, please visit https://www.zabbix.com/services

Official Zabbix documentation available at https://www.zabbix.com/documentation/
current/

********************************************************************************
[azureuser@Zabbix-proxy ~]$
```

Le fichier de configuration du proxy Zabbix se trouve dans `/etc/zabbix/zabbix_proxy.conf`.

La configuration initiale du proxy consiste à modifier les paramètres `ProxyMode`, `Server` et `Hostname` dans le fichier de configuration du proxy Zabbix.

Le mode du proxy peut être modifié en éditant le paramètre `ProxyMode`. Définissez `ProxyMode=0` pour le mode actif ou `ProxyMode=1` pour le mode passif.

Le paramètre `Server` doit contenir l'adresse IP du serveur Zabbix.

Le paramètre `Hostname` doit contenir le nom d'hôte du proxy Zabbix. Le nom d'hôte défini ici doit correspondre au nom du proxy indiqué lors de la configuration du proxy dans l'interface Zabbix.

```
############ GENERAL PARAMETERS #################

### Option: ProxyMode
#       Proxy operating mode.
#       0 - proxy in the active mode
#       1 - proxy in the passive mode
#
# Mandatory: no
# Default:
# ProxyMode=0

### Option: Server
#       If ProxyMode is set to active mode:
#               IP address or DNS name (address:port) or cluster (address:port;address2:port) of Zabbix server to get
configuration data from and send data to.
#               If port is not specified, default port is used.
#               Cluster nodes need to be separated by semicolon.
#       If ProxyMode is set to passive mode:
#               List of comma delimited IP addresses, optionally in CIDR notation, or DNS names of Zabbix server.
#               Incoming connections will be accepted only from the addresses listed here.
#               If IPv6 support is enabled then '127.0.0.1', '::127.0.0.1', '::ffff:127.0.0.1' are treated equally
#               and '::/0' will allow any IPv4 or IPv6 address.
#               '0.0.0.0/0' can be used to allow any IPv4 address.
#               Example: Server=127.0.0.1,192.168.1.0/24,::1,2001:db8::/32,zabbix.example.com
#
# Mandatory: yes
# Default:
# Server=

Server=127.0.0.1
```

```
### Option: Hostname
#       Unique, case sensitive Proxy name. Make sure the Proxy name is known to the server!
#       Value is acquired from HostnameItem if undefined.
#
# Mandatory: no
# Default:
# Hostname=

Hostname=Zabbix proxy
```

Ouvrez l'interface Zabbix et accédez à *Administration* > *Proxies*. Cliquez sur le bouton *Create proxy*.

Saisissez le même nom de proxy que la valeur du paramètre *Hostname* dans le fichier de configuration du proxy.

Sélectionnez le mode du proxy et indiquez l'adresse IP du proxy (facultatif pour les proxies actifs).

Cliquez sur le bouton *Create proxy* pour créer un proxy.

Emplacement de la base de données du proxy

La base de données SQLite du proxy se trouve dans `/tmp/zabbix_proxy.sqlite3` et est spécifiée par le paramètre `DBName` dans le fichier de configuration du proxy Zabbix.

Guide de déploiement Google Cloud

Vue d'ensemble

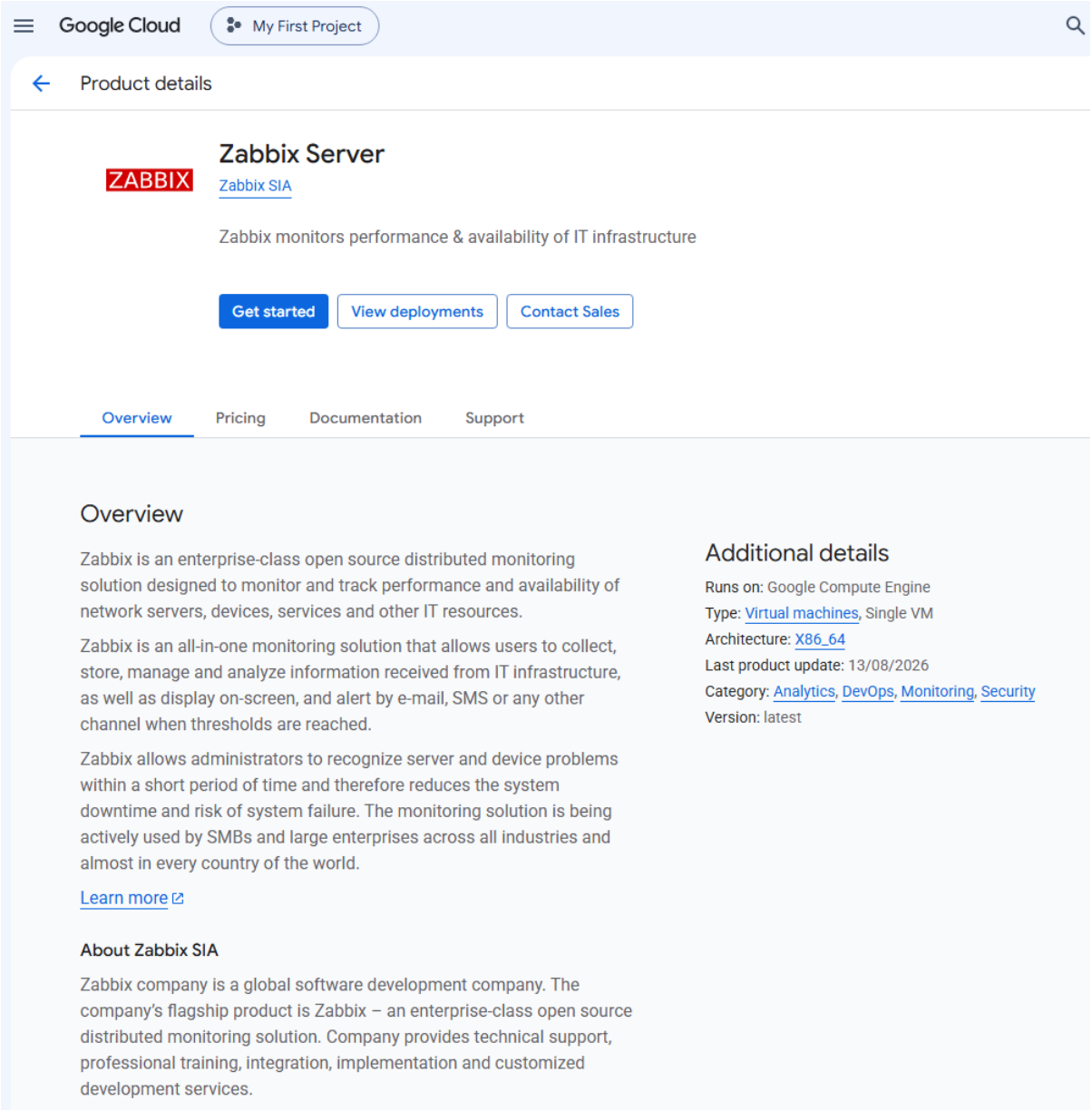
Cette page décrit comment déployer Zabbix à l'aide de Google Cloud Marketplace.

Attention:
Plusieurs versions de Zabbix sont disponibles pour être déployées sur la plateforme. Veillez à consulter la documentation correspondant à la même version de Zabbix que celle que vous avez déployée. La documentation des différentes versions de Zabbix peut être sélectionnée dans le menu déroulant des versions.

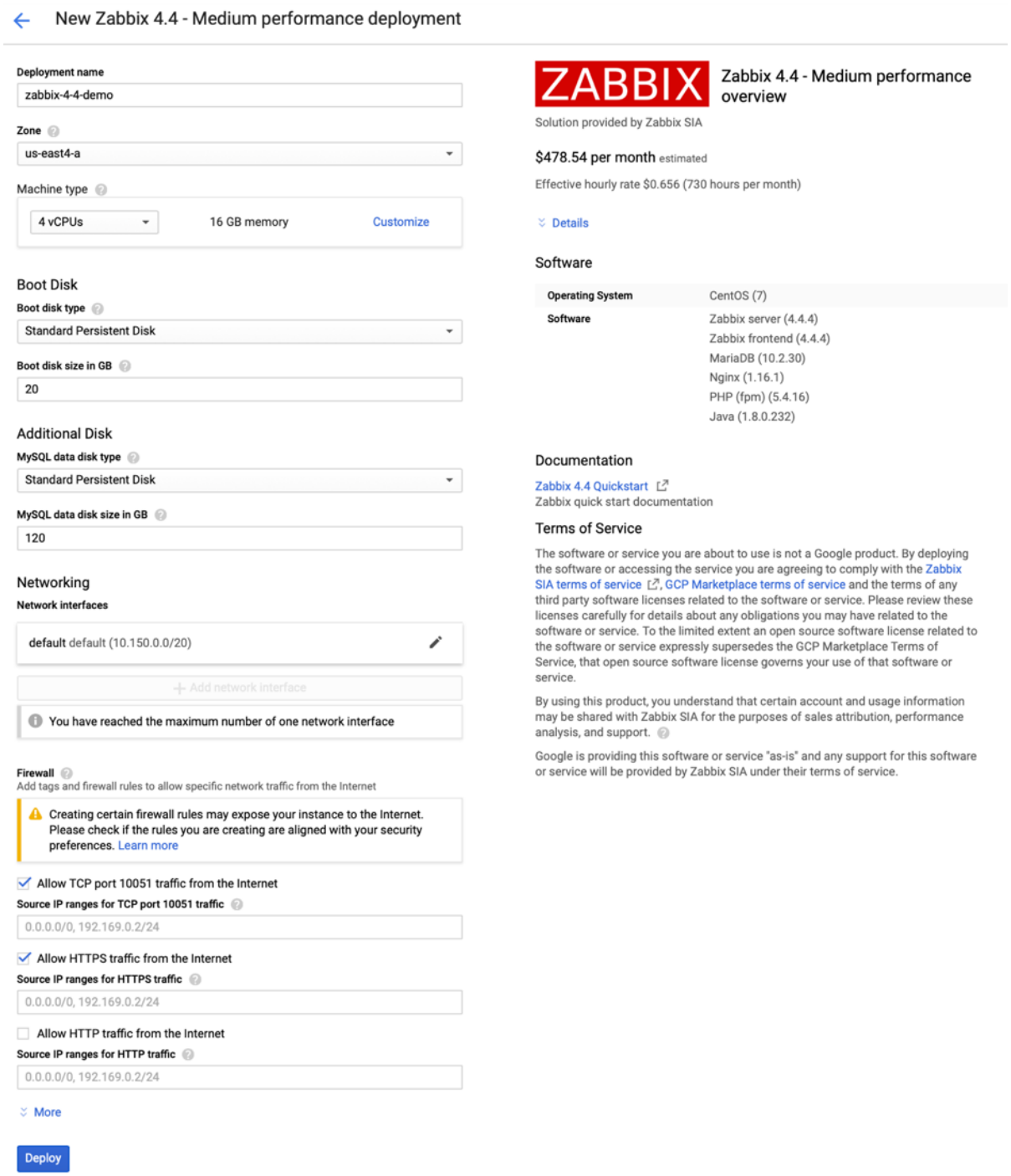
Déploiement de l'appliance

Commencez par déployer l'appliance Zabbix. C'est très simple avec Google Cloud Marketplace. Accédez simplement au produit Zabbix Server sur la marketplace.

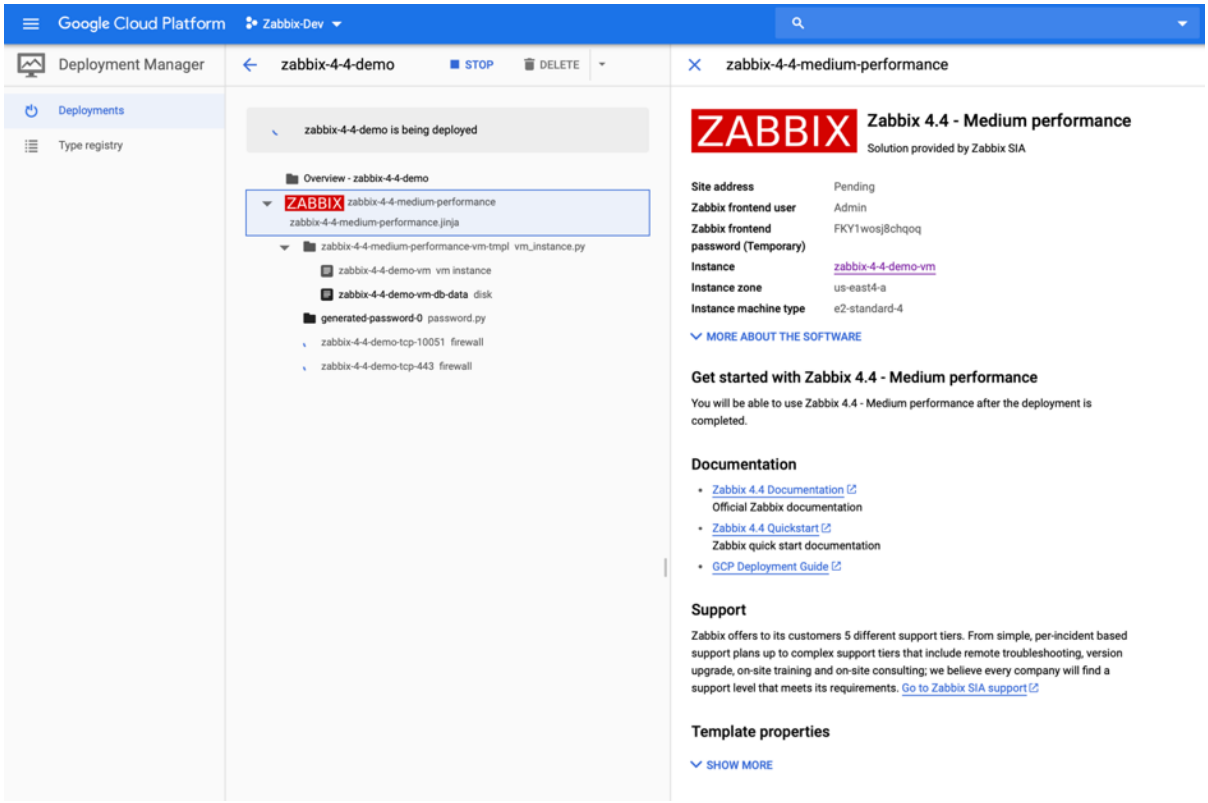
Cliquez sur *Get started*.



Vous pouvez conserver les paramètres par défaut ou les personnaliser. Lorsque vous avez terminé, cliquez sur *Deploy*.



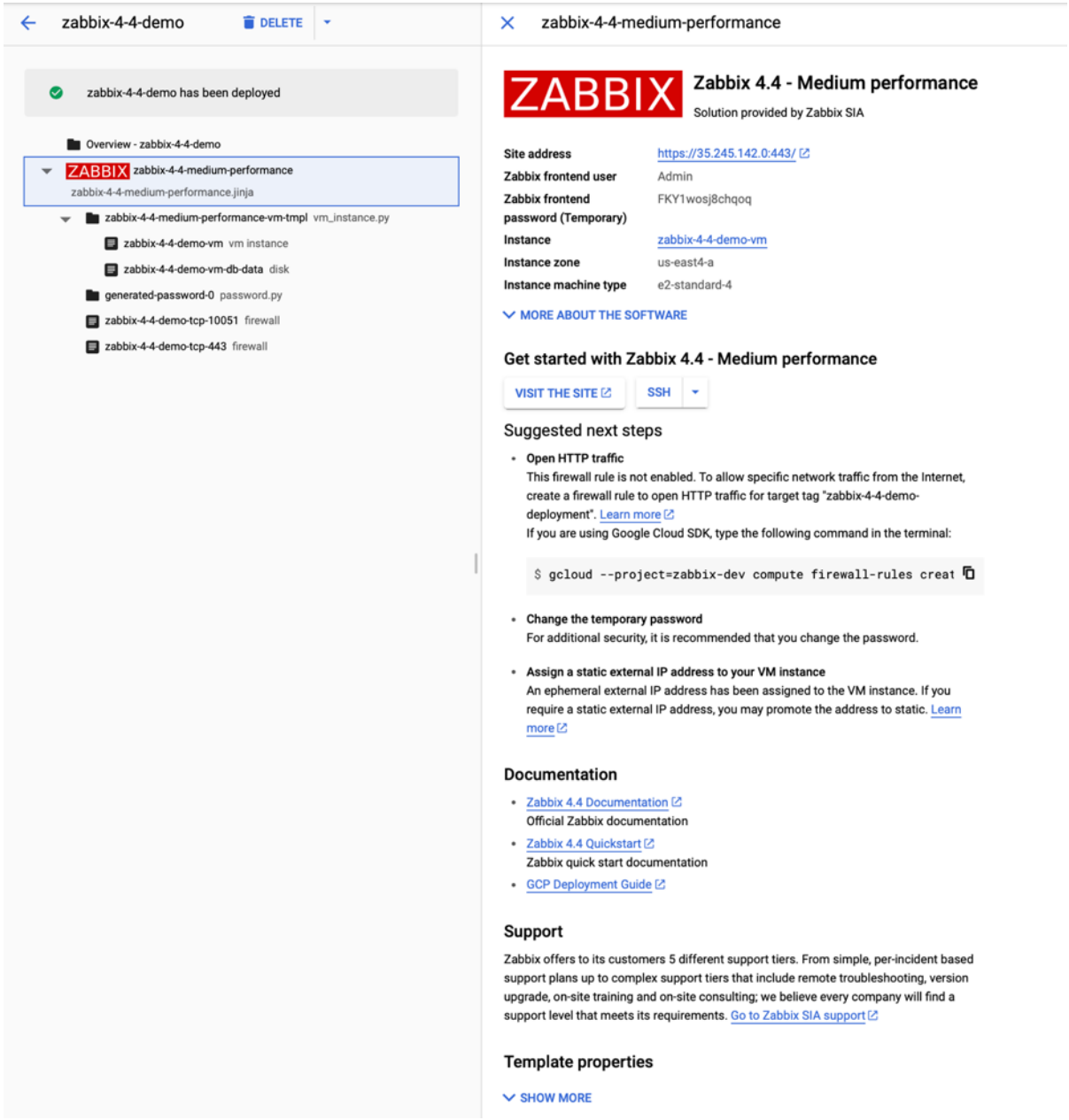
Le déploiement est maintenant en cours. Attendez qu'il soit terminé.



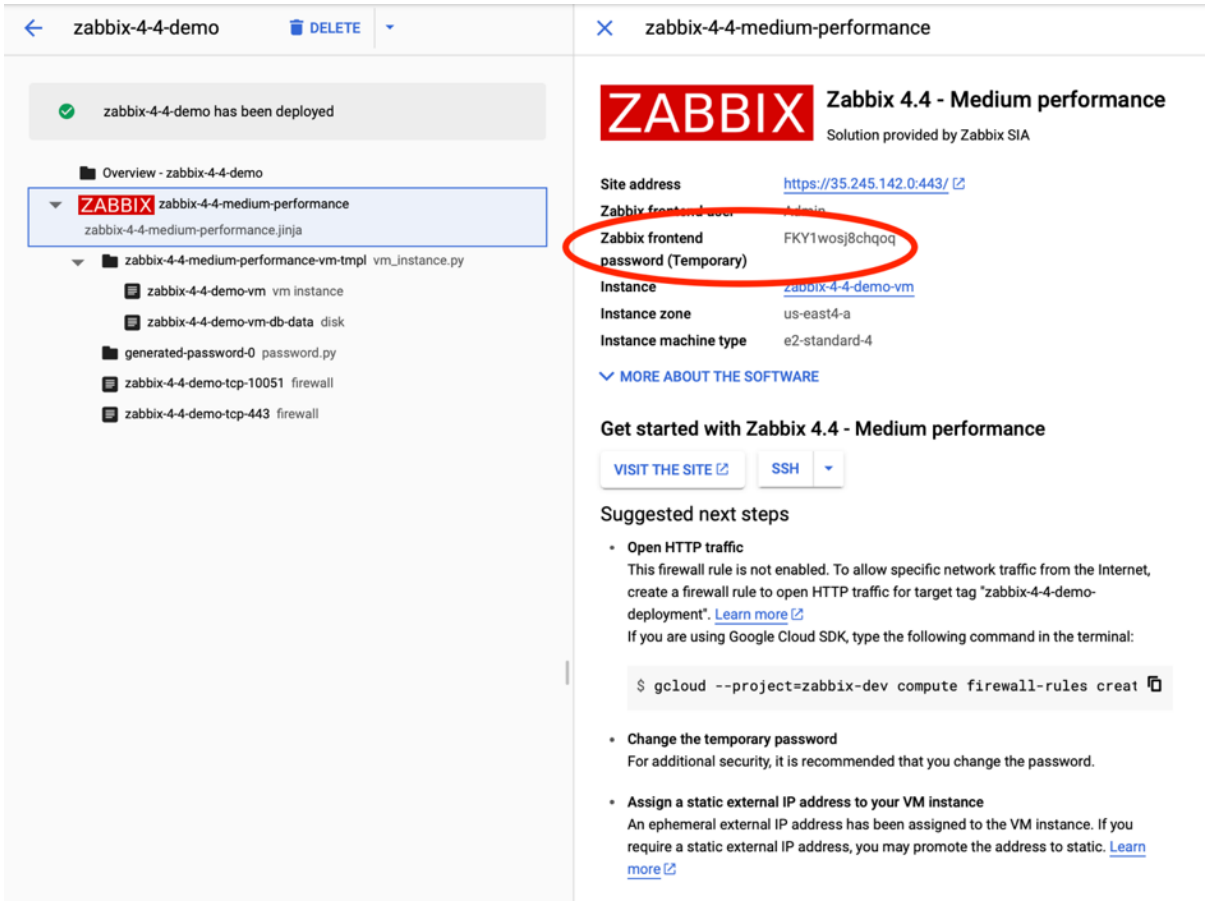
C’est tout ! Le déploiement a maintenant commencé.

Accès à l’appliance

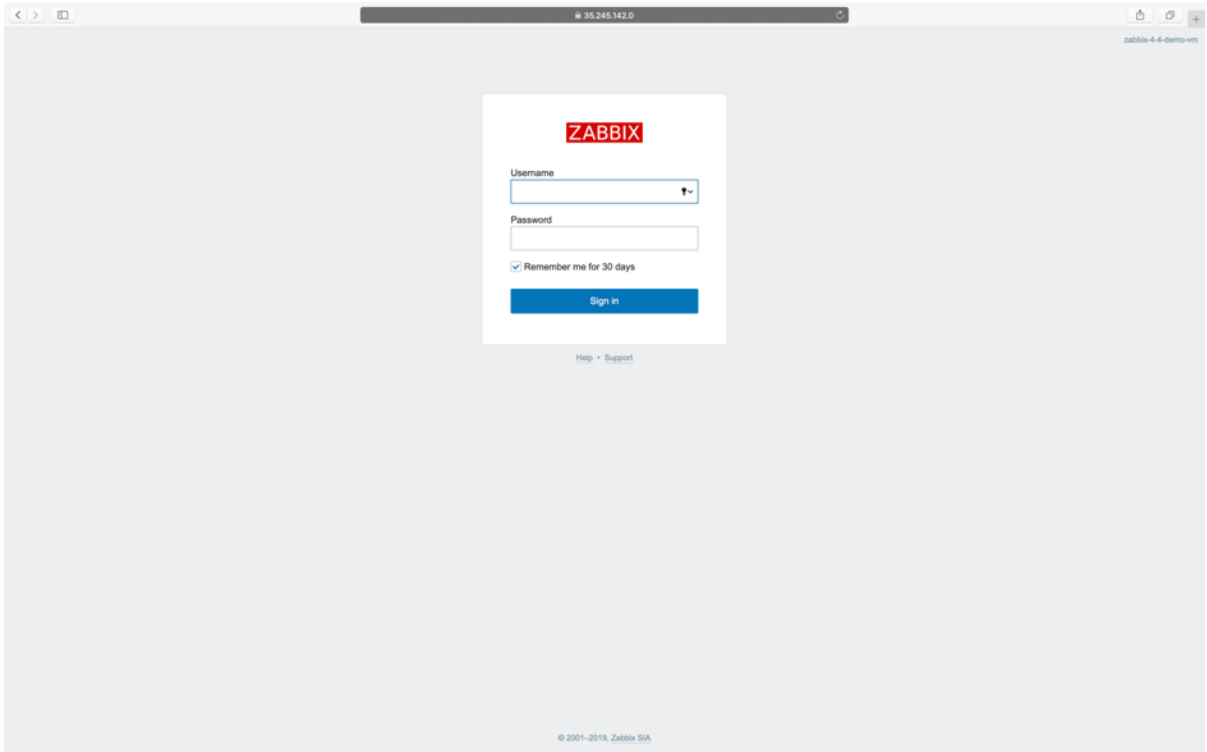
Une fois le déploiement terminé, vous devriez voir un message indiquant que l’appliance est en cours de déploiement.

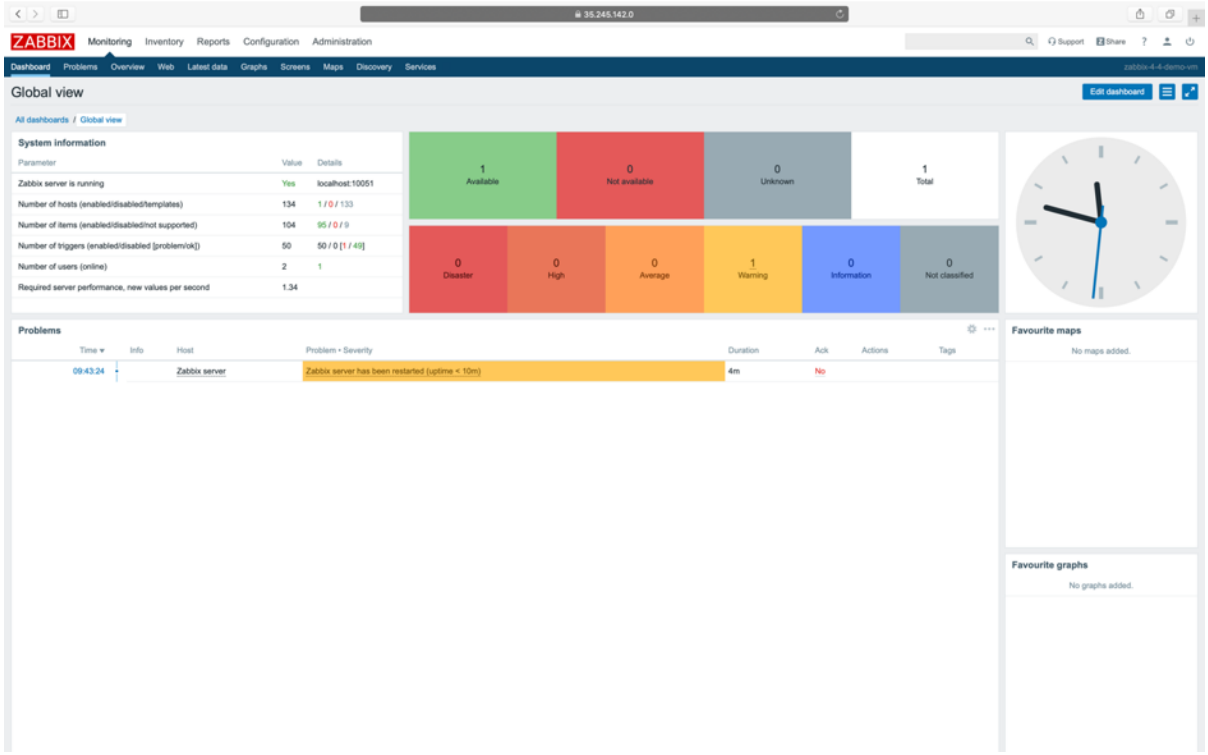


Ouvrez un navigateur web à l'adresse *Site Address* pour afficher l'interface Zabbix. Avant cela, récupérez le mot de passe de l'utilisateur *Admin* de Zabbix pour vous connecter à votre appliance Zabbix.

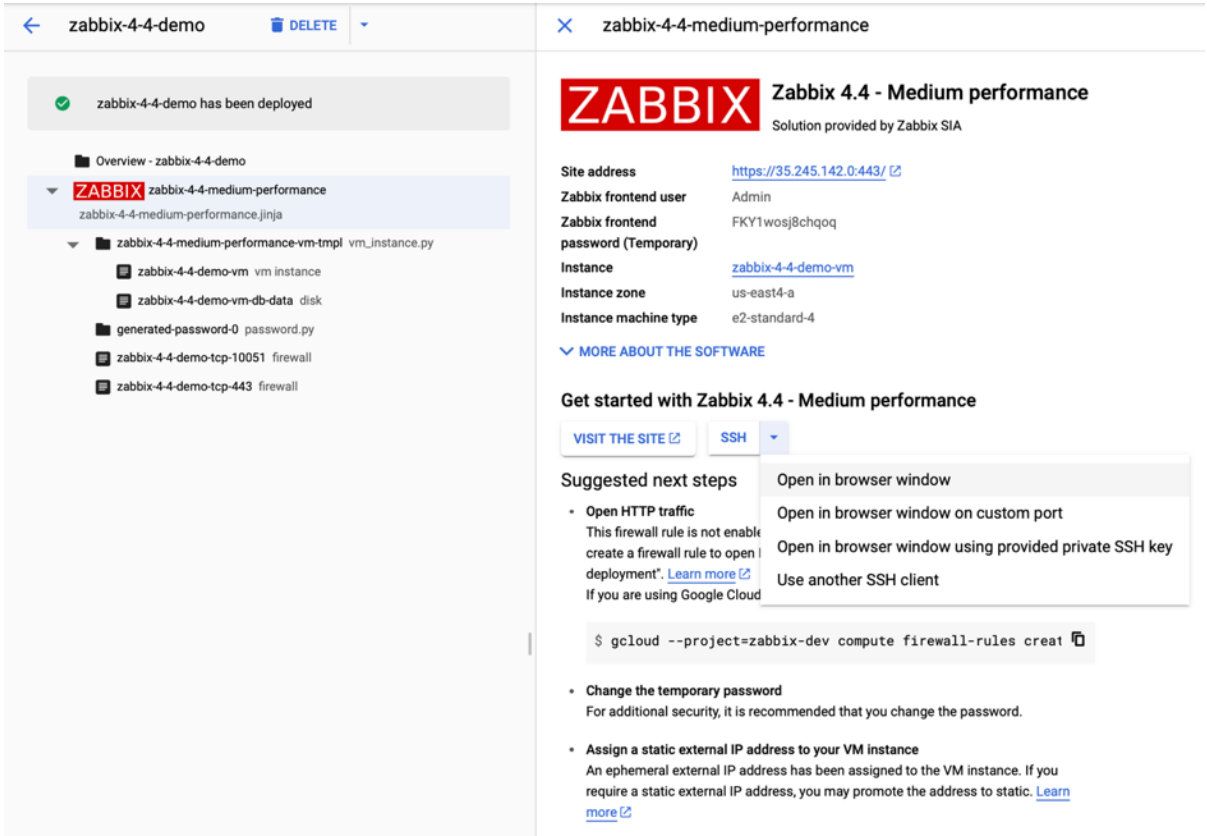


Parfait ! Vous disposez maintenant d'une appliance Zabbix.





Vous pouvez également vous connecter à l'instance pour interagir avec la base de données. Pour ce faire, retournez dans la console Google et suivez la flèche rouge comme indiqué ci-dessous afin de démarrer une session SSH à l'aide de l'option *Open in browser window*.



MySQL est disponible sans mot de passe `root`. Le mot de passe est stocké dans le fichier de configuration `/root/.my.cnf`. Par défaut, l'appliance Zabbix utilise le fuseau horaire UTC. Si vous devez le modifier, éditez la directive de configuration PHP `date.timezone` dans `/etc/php.d/99-zabbix.ini`, puis redémarrez le processus php-fpm : `systemctl restart php-fpm`.

Mise à niveau

Avant de procéder à la mise à niveau, il est fortement recommandé de consulter toutes les notes de mise à niveau pertinentes.

Il sera également utile de savoir quels composants vous devez mettre à niveau.

Vue d'ensemble

Des instructions de mise à niveau détaillées sont disponibles pour :

- Red Hat Enterprise Linux (à l'aide de paquets)
- Debian/Ubuntu (à l'aide de paquets)
- Mise à niveau à l'aide de conteneurs
- Mise à niveau à l'aide des sources

Instructions associées :

- Pour les serveurs dans un cluster à haute disponibilité (HA), consultez Mise à niveau d'un cluster HA
- Pour une base de données TimescaleDB, consultez Mise à niveau du schéma TimescaleDB

Composants à mettre à niveau

La mise à niveau des agents Zabbix est recommandée, mais pas obligatoire.

La mise à niveau des proxies Zabbix est fortement recommandée. Le serveur Zabbix prend entièrement en charge les proxies qui utilisent la même version majeure que le serveur. Le serveur Zabbix prend également en charge les proxies qui ne sont **pas antérieurs** à la version LTS précédente du serveur Zabbix, mais avec des fonctionnalités limitées (collecte de données, exécution de commandes à distance, vérifications immédiates de la valeur des éléments). La mise à jour de la configuration est également désactivée, et les proxies obsolètes ne fonctionneront qu'avec l'ancienne configuration.

> **Attention:**
> Les proxy dont la version est antérieure à celle de la précédente version LTS du serveur Zabbix ou ultérieure à la version majeure du serveur Zabbix ne sont pas pris en charge. Le serveur Zabbix ignorera les données provenant des proxy non pris en charge et toute communication avec le serveur Zabbix échouera avec un avertissement. Pour plus d'informations, consultez Compatibilité des versions.

Pour réduire au minimum les temps d'arrêt et les pertes de données pendant la mise à niveau, il est recommandé d'arrêter, de mettre à niveau et de démarrer le serveur Zabbix, puis d'arrêter, de mettre à niveau et de démarrer les proxy l'un après l'autre. Pendant l'arrêt du serveur, les proxy en fonctionnement continueront à collecter les données. Une fois le serveur opérationnel, les proxy obsolètes enverront les données au serveur plus récent (la configuration des proxy ne sera toutefois pas mise à jour) et resteront partiellement fonctionnels. Les notifications relatives aux problèmes survenus pendant l'arrêt du serveur Zabbix seront générées uniquement après le démarrage du serveur mis à niveau.

Si le proxy Zabbix est démarré pour la première fois et que le fichier de base de données SQLite est absent, le proxy le crée automatiquement.

Notez que si le proxy Zabbix utilise SQLite3 et détecte au démarrage que la version du fichier de base de données existant est antérieure à celle requise, il **supprimera automatiquement le fichier de base de données** et en créera un nouveau. Par conséquent, les données d'historique stockées dans le fichier de base de données SQLite seront perdues. Si la version du proxy Zabbix est antérieure à celle du fichier de base de données, Zabbix consignera une erreur et quittera.

Selon la taille de la base de données, la mise à niveau de la base de données vers la version 8.0 peut prendre beaucoup de temps.

Notes de mise à niveau

La mise à niveau directe vers Zabbix 8.0.x est prise en charge depuis les versions **2.0**.x. Pour effectuer une mise à niveau depuis des versions antérieures, consultez la documentation Zabbix relative à la version 2.0 et aux versions antérieures.

> **Note:**
> Veuillez noter qu'après la mise à niveau, certaines intégrations de logiciels tiers dans Zabbix peuvent être affectées si le logiciel externe n'est pas compatible avec la version mise à niveau de Zabbix.

Les notes de mise à niveau suivantes sont disponibles :

Mise à niveau depuis	Lire les notes complètes de mise à niveau	Modifications les plus importantes entre les versions
7.4.x	Pour : Zabbix 8.0	Les agents Zabbix antérieurs à la version 1.6 ne sont plus pris en charge. Les versions minimales requises des bases de données ont été relevées. La version minimale requise de PHP passe de 8.0.0 à 8.2.0. Le caractère % a été ajouté à la liste `UnsafeUserParameters` pour Zabbix agent et Zabbix agent 2.
7.2.x	Pour : Zabbix 7.4 Zabbix 8.0	La prise en charge de la bibliothèque PCRE a été supprimée.
7.0.x	Pour : Zabbix 7.2 Zabbix 7.4 Zabbix 8.0	La prise en charge d'Oracle DB a été supprimée.
6.4.x	Pour : Zabbix 7.0 Zabbix 7.2 Zabbix 7.4 Zabbix 8.0	La version minimale requise de PHP passe de 7.4.0 à 8.0.0. Interrogateurs asynchrones pour les vérifications agent, agent HTTP, SNMP walk[OID] et SNMP get[OID]. Table de base de données distincte pour les proxy. L'emplacement par défaut du fichier de configuration de l'agent Windows a été modifié. Oracle DB est obsolète. L'ancien type de valeur numérique (flottante) a été supprimé.
6.2.x	Pour : Zabbix 6.4 Zabbix 7.0 Zabbix 7.2 Zabbix 7.4 Zabbix 8.0	La version minimale requise de MySQL passe de 8.0.0 à 8.0.30. La bibliothèque 'libevent_pthreads' est requise pour le serveur/proxy Zabbix. Lors du premier démarrage après une mise à niveau, le proxy Zabbix utilisant SQLite3 supprime automatiquement l'ancienne version de la base de données (avec tout l'historique) et en crée une nouvelle.
6.0.x LTS	Pour : Zabbix 6.2 Zabbix 6.4 Zabbix 7.0 Zabbix 7.2 Zabbix 7.4 Zabbix 8.0	La version minimale requise de PHP passe de 7.2.5 à 7.4.0. La supervision des services a été considérablement remaniée. Les déclencheurs déterministes doivent être créés pendant la mise à niveau. Si la journalisation binaire est activée pour MySQL/MariaDB, cela nécessite des privilèges de superutilisateur ou la définition de la variable/du paramètre de configuration *log_bin_trust_function_creators = 1*. Consultez Scripts de création de la base de données pour savoir comment définir cette variable.
5.4.x	Pour : Zabbix 6.0 Zabbix 6.2 Zabbix 6.4 Zabbix 7.0 Zabbix 7.2 Zabbix 7.4 Zabbix 8.0	Les versions minimales requises des bases de données ont été relevées. Le serveur/proxy ne démarre pas si la base de données est obsolète. Des enregistrements du journal d'audit ont été perdus en raison d'une modification de la structure de la base de données.
5.2.x	Pour : Zabbix 5.4 Zabbix 6.0 Zabbix 6.2 Zabbix 6.4 Zabbix 7.0 Zabbix 7.2 Zabbix 7.4 Zabbix 8.0	Les versions minimales requises des bases de données ont été relevées. Les éléments agrégés ont été supprimés en tant que type distinct.

Mise à niveau depuis	Lire les notes complètes de mise à niveau	Modifications les plus importantes entre les versions
5.0.x LTS	Pour : Zabbix 5.2 Zabbix 5.4 Zabbix 6.0 Zabbix 6.2 Zabbix 6.4 Zabbix 7.0 Zabbix 7.2 Zabbix 7.4 Zabbix 8.0	La version minimale requise de PHP passe de 7.2.0 à 7.2.5. L'algorithme de hachage des mots de passe passe de MD5 à bcrypt.
4.4.x	Pour : Zabbix 5.0 Zabbix 5.2 Zabbix 5.4 Zabbix 6.0 Zabbix 6.2 Zabbix 6.4 Zabbix 7.0 Zabbix 7.2 Zabbix 7.4 Zabbix 8.0	La prise en charge d'IBM DB2 a été supprimée. La version minimale requise de PHP passe de 5.4.0 à 7.2.0. Les versions minimales requises des bases de données ont été relevées. Le répertoire des fichiers PHP de Zabbix a été modifié.
4.2.x	Pour : Zabbix 4.4 Zabbix 5.0 Zabbix 5.2 Zabbix 5.4 Zabbix 6.0 Zabbix 6.2 Zabbix 6.4 Zabbix 7.0 Zabbix 7.2 Zabbix 7.4 Zabbix 8.0	Les types de média Jabber et Ez Texting ont été supprimés.
4.0.x LTS	Pour : Zabbix 4.2 Zabbix 4.4 Zabbix 5.0 Zabbix 5.2 Zabbix 5.4 Zabbix 6.0 Zabbix 6.2 Zabbix 6.4 Zabbix 7.0 Zabbix 7.2 Zabbix 7.4 Zabbix 8.0	Les proxy plus anciens ne peuvent plus transmettre de données à un serveur mis à niveau. Les agents plus récents ne peuvent plus fonctionner avec un serveur Zabbix plus ancien.
3.4.x	Pour : Zabbix 4.0 Zabbix 4.2 Zabbix 4.4 Zabbix 5.0 Zabbix 5.2 Zabbix 5.4 Zabbix 6.0 Zabbix 6.2 Zabbix 6.4 Zabbix 7.0 Zabbix 7.2 Zabbix 7.4 Zabbix 8.0	Les bibliothèques 'libpthread' et 'zlib' sont désormais obligatoires. La prise en charge du protocole en texte brut a été supprimée et l'en-tête est obligatoire. Les agents Zabbix antérieurs à la version 1.4 ne sont plus pris en charge. Le paramètre Server dans la configuration d'un proxy passif est désormais obligatoire.

Mise à niveau depuis	Lire les notes complètes de mise à niveau	Modifications les plus importantes entre les versions
3.2.x	Pour : Zabbix 3.4 Zabbix 4.0 Zabbix 4.2 Zabbix 4.4 Zabbix 5.0 Zabbix 5.2 Zabbix 5.4 Zabbix 6.0 Zabbix 6.2 Zabbix 6.4 Zabbix 7.0 Zabbix 7.2 Zabbix 7.4 Zabbix 8.0	La prise en charge de SQLite comme base de données principale a été supprimée pour le serveur/l'interface Zabbix. Les expressions régulières compatibles Perl (PCRE) sont prises en charge à la place des expressions étendues POSIX. Les bibliothèques 'libpcre' et 'libevent' sont obligatoires pour le serveur Zabbix. Des vérifications du code de sortie ont été ajoutées pour les paramètres utilisateur, les commandes distantes et les éléments system.run[] sans l'indicateur 'nowait', ainsi que pour les scripts exécutés par le serveur Zabbix. Java gateway Zabbix doit être mis à niveau pour prendre en charge les nouvelles fonctionnalités.
3.0.x LTS	Pour : Zabbix 3.2 Zabbix 3.4 Zabbix 4.0 Zabbix 4.2 Zabbix 4.4 Zabbix 5.0 Zabbix 5.2 Zabbix 5.4 Zabbix 6.0 Zabbix 6.2 Zabbix 6.4 Zabbix 7.0 Zabbix 7.2 Zabbix 7.4 Zabbix 8.0	La mise à niveau de la base de données peut être lente, selon la taille de la table d'historique.
2.4.x	Pour : Zabbix 3.0 Zabbix 3.2 Zabbix 3.4 Zabbix 4.0 Zabbix 4.2 Zabbix 4.4 Zabbix 5.0 Zabbix 5.2 Zabbix 5.4 Zabbix 6.0 Zabbix 6.2 Zabbix 6.4 Zabbix 7.0 Zabbix 7.2 Zabbix 7.4 Zabbix 8.0	La version minimale requise de PHP passe de 5.3.0 à 5.4.0. Le paramètre d'agent LogFile doit être spécifié.

Mise à niveau depuis	Lire les notes complètes de mise à niveau	Modifications les plus importantes entre les versions
2.2.x LTS	Pour : Zabbix 2.4 Zabbix 3.0 Zabbix 3.2 Zabbix 3.4 Zabbix 4.0 Zabbix 4.2 Zabbix 4.4 Zabbix 5.0 Zabbix 5.2 Zabbix 5.4 Zabbix 6.0 Zabbix 6.2 Zabbix 6.4 Zabbix 7.0 Zabbix 7.2 Zabbix 7.4 Zabbix 8.0	La supervision distribuée basée sur les nœuds a été supprimée.
2.0.x	Pour : Zabbix 2.2 Zabbix 2.4 Zabbix 3.0 Zabbix 3.2 Zabbix 3.4 Zabbix 4.0 Zabbix 4.2 Zabbix 4.4 Zabbix 5.0 Zabbix 5.2 Zabbix 5.4 Zabbix 6.0 Zabbix 6.2 Zabbix 6.4 Zabbix 7.0 Zabbix 7.2 Zabbix 7.4 Zabbix 8.0	La version minimale requise de PHP passe de 5.1.6 à 5.3.0. Une base de données MySQL sensible à la casse est requise pour le bon fonctionnement du serveur ; le jeu de caractères utf8 et la collation utf8_bin sont requis pour que le serveur Zabbix fonctionne correctement avec une base de données MySQL. Consultez les scripts de création de la base de données. L'extension PHP 'mysqli' est requise à la place de 'mysql'.

Mise à niveau à partir de paquets

Aperçu

Cette section fournit les étapes requises pour effectuer une mise à niveau réussie à l'aide des paquets RPM et DEB officiels fournis par Zabbix pour :

- Red Hat Enterprise Linux
- Debian/Ubuntu

Paquets Zabbix provenant des dépôts du système d'exploitation

Certaines distributions du système d'exploitation (en particulier les distributions basées sur Debian) fournissent leurs propres paquets Zabbix. Ces paquets **ne sont pas pris en charge par Zabbix** et peuvent être obsolètes ou ne pas inclure les dernières fonctionnalités et corrections de bogues. Il est recommandé d'utiliser uniquement les paquets officiels du dépôt officiel Zabbix.

Si vous effectuez une mise à niveau à partir de paquets fournis par des distributions du système d'exploitation (ou si vous les avez eus installés à un moment donné), suivez cette procédure pour passer aux paquets officiels Zabbix :

1. Désinstallez les anciens paquets.
2. Vérifiez s'il reste des fichiers de l'ancienne installation et supprimez-les.
3. Installez les paquets officiels Zabbix en suivant les instructions d'installation fournies par Zabbix.

N'effectuez pas de mise à niveau directe, car cela pourrait entraîner une installation défectueuse.

Debian/Ubuntu

Aperçu

Cette section fournit des instructions pour effectuer la mise à niveau de Zabbix **7.4.x** vers la dernière version de Zabbix **8.0.x** à l'aide des paquets officiels Zabbix pour Debian/Ubuntu.

Warning:
Avant la mise à niveau, veuillez consulter les notes de mise à niveau correspondantes et vous assurer que votre système répond aux exigences de Zabbix 8.0.

Note:
Envisagez d'exécuter deux sessions SSH parallèles pendant la mise à niveau : l'une pour exécuter les étapes de mise à niveau et l'autre pour surveiller les journaux du serveur/proxy. Par exemple, exécutez `tail -f zabbix_server.log` ou `tail -f zabbix_proxy.log` dans la seconde session pour afficher en temps réel les dernières entrées du journal et les erreurs éventuelles. Cela peut être essentiel pour les instances de production.

Pour obtenir des instructions sur la mise à niveau entre les versions mineures de Zabbix 8.0.x (par exemple, de 8.0.1 à 8.0.3), consultez Mise à niveau entre versions mineures.

Procédure de mise à niveau

1 Arrêter les processus Zabbix

Arrêtez le serveur Zabbix pour vous assurer qu'aucune nouvelle donnée n'est insérée dans la base de données :

```
systemctl stop zabbix-server
```

Si vous mettez à niveau Zabbix proxy, l'agent ou agent 2, arrêtez également ces composants :

```
systemctl stop zabbix-proxy
systemctl stop zabbix-agent
systemctl stop zabbix-agent2
```

2 Sauvegarder la base de données Zabbix

Sauvegardez votre base de données Zabbix existante afin de vous prémunir contre les échecs de mise à niveau (par exemple, des problèmes d'espace disque, une panne de courant ou des problèmes inattendus).

3 Sauvegarder les fichiers de configuration Zabbix, les fichiers PHP et les binaires Zabbix

Sauvegardez les fichiers de configuration Zabbix, les fichiers PHP et les binaires Zabbix existants.

Pour les fichiers de configuration, exécutez :

```
mkdir /opt/zabbix-backup/
cp /etc/zabbix/zabbix_server.conf /opt/zabbix-backup/
##### Pour Apache :
cp /etc/apache2/conf-enabled/zabbix.conf /opt/zabbix-backup/
##### Pour Nginx :
cp /etc/nginx/conf.d/zabbix.conf /opt/zabbix-backup/
cp /etc/zabbix/nginx.conf /opt/zabbix-backup/
```

Pour les fichiers PHP et les binaires Zabbix, exécutez :

```
cp -R /usr/share/zabbix/ /opt/zabbix-backup/
cp -R /usr/share/zabbix-* /opt/zabbix-backup/
```

4 Mettre à jour le paquet de configuration du dépôt

Avant de procéder à la mise à niveau, désinstallez votre paquet de dépôt Zabbix actuel :

```
rm -Rf /etc/apt/sources.list.d/zabbix.list
```

Vous devrez peut-être aussi supprimer manuellement d'anciens paquets Zabbix de votre répertoire de travail (par exemple, `rm zabbix-release_latest+debian12_all.deb`) avant de télécharger le nouveau, afin d'empêcher le gestionnaire de paquets de réutiliser une version obsolète pendant le processus de mise à niveau.

Ensuite, installez le dernier paquet de configuration du dépôt afin de garantir la compatibilité avec les paquets les plus récents et d'inclure les correctifs de sécurité ou de bogues les plus récents.

Sur **Debian 13**, exécutez :

```
wget https://repo.zabbix.com/zabbix/8.0/release/debian/pool/main/z/zabbix-release/zabbix-release_latest+de
dpkg -i zabbix-release_latest+debian13_all.deb
```

Sur **Debian 12**, exécutez :

```
wget https://repo.zabbix.com/zabbix/8.0/release/debian/pool/main/z/zabbix-release/zabbix-release_latest+de
dpkg -i zabbix-release_latest+debian12_all.deb
```

> **Note:**
> Pour les anciennes versions de Debian, remplacez le lien ci-dessus par le bon lien depuis le dépôt Zabbix. Notez toutefois que les paquets pour ces versions peuvent ne pas inclure tous les composants Zabbix, et que pour mettre à niveau ces composants à partir des paquets, il est préférable de mettre à niveau votre système d'exploitation. Pour la liste des composants inclus, voir Paquets Zabbix.

Sur **Ubuntu 26.04**, exécutez :

```
wget https://repo.zabbix.com/zabbix/8.0/release/ubuntu/pool/main/z/zabbix-release/zabbix-release_latest+ub
dpkg -i zabbix-release_latest+ubuntu26.04_all.deb
```

Sur **Ubuntu 24.04**, exécutez :

```
wget https://repo.zabbix.com/zabbix/8.0/release/ubuntu/pool/main/z/zabbix-release/zabbix-release_latest+ub
dpkg -i zabbix-release_latest+ubuntu24.04_all.deb
```

Sur **Ubuntu 22.04**, exécutez :

```
wget https://repo.zabbix.com/zabbix/8.0/release/ubuntu/pool/main/z/zabbix-release/zabbix-release_latest+ub
dpkg -i zabbix-release_latest+ubuntu22.04_all.deb
```

> **Note:**
> Pour les anciennes versions d'Ubuntu, remplacez le lien ci-dessus par le bon lien depuis le dépôt Zabbix. Notez toutefois que les paquets pour ces versions peuvent ne pas inclure tous les composants Zabbix, et que pour mettre à niveau ces composants à partir des paquets, il est préférable de mettre à niveau votre système d'exploitation. Pour la liste des composants inclus, voir Paquets Zabbix.

Vous pouvez voir une invite concernant la configuration du dépôt Zabbix :

```
Configuration file '/etc/apt/sources.list.d/zabbix.list'
==> Deleted (by you or by a script) since installation.
==> Package distributor has shipped an updated version.
What would you like to do about it ?  Your options are:
Y or I  : install the package maintainer's version
N or O  : keep your currently-installed version
D       : show the differences between the versions
Z       : start a shell to examine the situation
The default action is to keep your current version.
*** zabbix.list (Y/I/N/O/D/Z) [default=N] ?
```

Entrez `Y` (ou `I`) pour installer la version du mainteneur du paquet de configuration du dépôt Zabbix.

Ensuite, mettez à jour les informations du dépôt :

```
apt update
```

5 Mettre à niveau les composants Zabbix

Pour mettre à niveau les composants Zabbix, exécutez :

```
apt install --only-upgrade zabbix-server-mysql zabbix-frontend-php zabbix-agent
```

- Si vous utilisez PostgreSQL, remplacez `mysql` par `pgsql` dans la commande.
- Si vous mettez à niveau le proxy, remplacez `server` par `proxy` dans la commande.
- Si vous mettez à niveau Zabbix agent 2, remplacez `zabbix-agent` par `zabbix-agent2 zabbix-agent2-plugin-*` dans la commande.

Attention:
La mise à niveau de Zabbix agent 2 avec la commande `apt install zabbix-agent2` peut entraîner une erreur. Pour plus d'informations, consultez *Problèmes connus*.

Une invite concernant la configuration du serveur Zabbix (ou du proxy) peut s'afficher :

```
Configuration file '/etc/zabbix/zabbix_server.conf'
==> Modified (by you or by a script) since installation.
==> Package distributor has shipped an updated version.
What would you like to do about it ?  Your options are:
Y or I  : install the package maintainer's version
N or O  : keep your currently-installed version
D       : show the differences between the versions
Z       : start a shell to examine the situation
The default action is to keep your current version.
*** zabbix_server.conf (Y/I/N/O/D/Z) [default=N] ?
```

Saisissez l'option qui correspond le mieux à votre situation. Par exemple, saisissez `D` pour comparer la configuration actuelle et la nouvelle configuration, puis décidez si vous souhaitez installer la version du responsable du paquet (`Y` ou `I`).

Ensuite, mettez à niveau l'interface Zabbix et redémarrez le serveur web.

Pour Apache, exécutez :

```
apt install zabbix-apache-conf
systemctl restart apache2
```

Pour Nginx, exécutez :

```
apt install zabbix-nginx-conf
systemctl restart nginx
```

Pour Nginx, une invite concernant le fichier de configuration `nginx.conf` peut s'afficher. Saisissez l'option qui correspond le mieux à votre situation :

```
Configuration file '/etc/zabbix/nginx.conf'
 ==> Modified (by you or by a script) since installation.
 ==> Package distributor has shipped an updated version.
   What would you like to do about it ?  Your options are:
    Y or I  : install the package maintainer's version
    N or O  : keep your currently-installed version
      D     : show the differences between the versions
      Z     : start a shell to examine the situation
 The default action is to keep your current version.
*** nginx.conf (Y/I/N/O/D/Z) [default=N] ?
```

Si, après la mise à niveau, vous devez installer un paquet Zabbix supplémentaire depuis le même dépôt, `apt` peut sélectionner les mises à niveau des paquets Zabbix déjà installés lors de la résolution des dépendances. Pour éviter de mettre à niveau un paquet de serveur Zabbix ou de proxy installé, bloquez temporairement ce paquet avant l'installation, puis débloquez-le ensuite :

```
apt-mark hold zabbix-server-mysql
apt install zabbix-sender
apt-mark unhold zabbix-server-mysql
```

6 Examiner les paramètres de configuration du composant

Consultez les notes de mise à niveau pertinentes pour vérifier si des modifications des paramètres de configuration sont nécessaires.

Pour les nouveaux paramètres facultatifs, consultez la page Nouveautés.

Si vous utilisez Nginx comme serveur web et que le fichier de configuration `/etc/zabbix/nginx.conf` a été réinstallé pendant la mise à niveau, examinez le fichier et mettez à jour les directives `listen` et `server_name` afin qu'elles correspondent à la configuration de votre interface Zabbix.

7 Démarrer les processus Zabbix

Démarrez les composants Zabbix mis à niveau :

```
systemctl start zabbix-server
systemctl start zabbix-proxy
systemctl start zabbix-agent
systemctl start zabbix-agent2
```

8 Effacer les cookies et le cache du navigateur web

Après la mise à niveau, vous devrez peut-être effacer les cookies et le cache du navigateur web pour que l'interface web de Zabbix fonctionne correctement.

Mise à niveau entre versions mineures

Il est possible d'effectuer une mise à niveau entre les versions mineures 8.0.x de Zabbix (par exemple, de 8.0.1 à 8.0.3).

Tout d'abord, mettez à jour les informations du dépôt :

```
apt update
```

Ensuite, pour mettre à niveau tous les composants Zabbix, exécutez :

```
apt install --only-upgrade 'zabbix*'
```

- Pour mettre à niveau uniquement le serveur Zabbix, remplacez `'zabbix*'` par `'zabbix-server*'` dans la commande.
- Pour mettre à niveau uniquement le proxy Zabbix, remplacez `'zabbix*'` par `'zabbix-proxy*'` dans la commande.
- Pour mettre à niveau uniquement l'agent Zabbix, remplacez `'zabbix*'` par `'zabbix-agent*'` dans la commande.
- Pour mettre à niveau uniquement l'agent Zabbix 2, remplacez `'zabbix*'` par `'zabbix-agent2*'` dans la commande.

Red Hat Enterprise Linux

Aperçu

Cette section fournit des instructions pour effectuer la mise à niveau de Zabbix **7.4.x** vers la dernière version de Zabbix **8.0.x** à l'aide des paquets officiels Zabbix pour Red Hat Enterprise Linux ou ses dérivés - AlmaLinux, CentOS Stream, Oracle Linux et Rocky Linux.

> **Warning:**
> Avant la mise à niveau, veuillez consulter les notes de mise à niveau correspondantes et vous assurer que votre système répond aux exigences de Zabbix 8.0.

> **Note:**
> Envisagez d'exécuter deux sessions SSH parallèles pendant la mise à niveau : l'une pour exécuter les étapes de mise à niveau et l'autre pour surveiller les journaux du serveur/proxy. Par exemple, exécutez `tail -f zabbix_server.log` ou `tail -f zabbix_proxy.log` dans la seconde session afin d'afficher les dernières entrées du journal et les erreurs éventuelles en temps réel. Cela peut être essentiel pour les instances de production.

Pour obtenir des instructions sur la mise à niveau entre les versions mineures de Zabbix 8.0.x (par exemple, de 8.0.1 à 8.0.3), consultez Mise à niveau entre versions mineures.

Procédure de mise à niveau

1 Arrêter les processus Zabbix

Arrêtez le serveur Zabbix pour vous assurer qu'aucune nouvelle donnée n'est insérée dans la base de données :

```
systemctl stop zabbix-server
```

Si vous mettez à niveau Zabbix proxy, l'agent ou agent 2, arrêtez également ces composants :

```
systemctl stop zabbix-proxy
systemctl stop zabbix-agent
systemctl stop zabbix-agent2
```

2 Sauvegarder la base de données Zabbix

Sauvegardez votre base de données Zabbix existante afin de vous prémunir contre les échecs de mise à niveau (par exemple, des problèmes d'espace disque, une panne de courant ou des problèmes inattendus).

3 Sauvegarder les fichiers de configuration de Zabbix, les fichiers PHP et les binaires Zabbix

Sauvegardez les fichiers de configuration Zabbix existants, les fichiers PHP et les binaires Zabbix.

Pour les fichiers de configuration, exécutez :

```
mkdir /opt/zabbix-backup/
cp /etc/zabbix/zabbix_server.conf /opt/zabbix-backup/
cp /etc/httpd/conf.d/zabbix.conf  /opt/zabbix-backup/
```

Pour les fichiers PHP et les binaires Zabbix, exécutez :

```
cp -R /usr/share/zabbix/ /opt/zabbix-backup/
cp -R /usr/share/zabbix-* /opt/zabbix-backup/
```

4 Mettre à jour le paquet de configuration du dépôt

Avant de procéder à la mise à niveau, mettez à jour votre paquet de dépôt actuel vers la dernière version afin de garantir la compatibilité avec les paquets les plus récents et d'inclure les correctifs de sécurité ou corrections de bogues récemment publiés.

Sur **RHEL 10**, exécutez :

```
rpm -Uvh https://repo.zabbix.com/zabbix/8.0/release/rhel/10/noarch/zabbix-release-latest.el10.noarch.rpm
```

Si vous mettez à niveau le proxy Zabbix sur RHEL 10 et souhaitez utiliser APM, installez également le dépôt EPEL et activez le dépôt CRB, qui fournissent tous deux des dépendances pour le paquet APM du proxy Zabbix :

```
dnf -y install epel-release || dnf -y install https://dl.fedoraproject.org/pub/epel/epel-release-latest-$(
crb enable
```

Sur **RHEL 9**, exécutez :

```
rpm -Uvh https://repo.zabbix.com/zabbix/8.0/release/rhel/9/noarch/zabbix-release-latest.el9.noarch.rpm
```

> **Note:**
> Pour les anciennes versions de RHEL ou leurs dérivés, remplacez le lien ci-dessus par le lien approprié du dépôt Zabbix. Notez toutefois que les paquets pour ces versions peuvent ne pas inclure tous les composants Zabbix. Pour mettre à niveau ces composants à partir des paquets, envisagez de mettre à niveau votre système d'exploitation. Pour obtenir la liste des composants inclus, consultez Paquets Zabbix.

Ensuite, nettoyez le cache du gestionnaire de paquets `dnf` (y compris les en-têtes, les métadonnées et les fichiers de paquets téléchargés lors des installations ou mises à jour précédentes) :

```
dnf clean all
```

Lors de la prochaine opération `dnf`, `dnf` téléchargera les métadonnées actualisées depuis les dépôts, puisque les anciennes métadonnées ont été supprimées.

Voir également : Problèmes connus pour mettre à jour le paquet de configuration du dépôt sur RHEL.

5 Mettre à niveau les composants Zabbix

Pour mettre à niveau les composants Zabbix, exécutez :

```
dnf install zabbix-server-mysql zabbix-web-mysql zabbix-selinux-policy zabbix-agent
```

- Si vous utilisez PostgreSQL, remplacez `mysql` par `pgsql` dans la commande.
- Si vous mettez à niveau le proxy, remplacez `server` par `proxy` dans la commande.
- Si vous utilisez RHEL 10 et souhaitez utiliser APM, ajoutez `-apm` au nom du paquet du proxy (par exemple, `zabbix-proxy-mysql-apm`).
- Si vous mettez à niveau l'agent 2, remplacez `zabbix-agent` par `zabbix-agent2 zabbix-agent2-plugin-*` dans la commande.
- Si d'autres paquets Zabbix sont installés, envisagez également de les mettre à niveau en les ajoutant à la commande.

> **Attention:**
> La mise à niveau de l'agent 2 Zabbix avec la commande `dnf install zabbix-agent2` peut entraîner une erreur. Pour plus d'informations, consultez *Problèmes connus*.

Ensuite, pour mettre à niveau l'interface Zabbix avec Apache et redémarrer Apache, exécutez :

```
dnf install zabbix-apache-conf
systemctl restart httpd
```

6 Examiner les paramètres de configuration du composant

Consultez les notes de mise à niveau pertinentes pour vérifier si des modifications des paramètres de configuration sont nécessaires.

Pour les nouveaux paramètres facultatifs, consultez la page Nouveautés.

7 Démarrer les processus Zabbix

Démarrez les composants Zabbix mis à niveau :

```
systemctl start zabbix-server
systemctl start zabbix-proxy
systemctl start zabbix-agent
systemctl start zabbix-agent2
```

8 Effacer les cookies et le cache du navigateur web

Après la mise à niveau, vous devrez peut-être effacer les cookies et le cache du navigateur web pour que l'interface web de Zabbix fonctionne correctement.

Mise à niveau entre versions mineures

Il est possible d'effectuer une mise à niveau entre les versions mineures 8.0.x de Zabbix (par exemple, de 8.0.1 à 8.0.3).

Pour mettre à niveau tous les composants Zabbix, exécutez :

```
dnf upgrade 'zabbix-*'
```

- Pour mettre à niveau uniquement le serveur Zabbix, remplacez `'zabbix-*'` par `'zabbix-server-*'` dans la commande.
- Pour mettre à niveau uniquement le proxy Zabbix, remplacez `'zabbix-*'` par `'zabbix-proxy-*'` dans la commande.
- Pour mettre à niveau uniquement l'agent Zabbix, remplacez `'zabbix-*'` par `'zabbix-agent-*'` dans la commande.
- Pour mettre à niveau uniquement l'agent Zabbix 2, remplacez `'zabbix-*'` par `'zabbix-agent2-*'` dans la commande.

Mise à niveau à partir des sources

Vue d'ensemble

Cette section présente les étapes nécessaires pour effectuer une mise à niveau réussie de Zabbix **7.4**.x vers Zabbix **8.0**.x à l'aide des sources officielles de Zabbix.

> **Warning:**
> Avant la mise à niveau, veillez à lire les **notes de mise à niveau** pertinentes !

Vous pouvez également consulter les prérequis pour la version 8.0.

> **Note:**
> Il peut être utile d'exécuter deux sessions SSH en parallèle pendant la mise à niveau : effectuez les étapes de mise à niveau dans l'une et surveillez les journaux du serveur/proxy dans l'autre. Par exemple, exécutez `tail -f zabbix_server.log` ou `tail -f zabbix_proxy.log` dans la seconde session SSH afin d'afficher les dernières entrées du fichier journal et les éventuelles erreurs en temps réel. Cela peut être essentiel pour les instances de production.

Procédure de mise à jour du serveur

1 Arrêter le serveur

Arrêtez le serveur Zabbix afin de vous assurer qu'aucune nouvelle donnée n'est insérée dans la base de données.

2 Sauvegarder la base de données Zabbix existante

Il s'agit d'une étape très importante.
Assurez-vous de disposer d'une sauvegarde de votre base de données.
Elle vous sera utile si la procédure de mise à niveau échoue (espace disque insuffisant, coupure de courant ou tout autre problème inattendu).

3 Sauvegarder les fichiers de configuration, les fichiers PHP et les binaires Zabbix

Effectuez une copie de sauvegarde des binaires Zabbix, des fichiers de configuration et du répertoire des fichiers PHP.

4 Installer les nouveaux binaires du serveur

Utilisez ces instructions pour compiler le serveur Zabbix à partir des sources.

5 Examiner les paramètres de configuration du serveur

Assurez-vous de consulter les Notes de mise à niveau pour vérifier si des modifications des paramètres de configuration sont nécessaires.

Pour les nouveaux paramètres facultatifs, consultez la page Nouveautés.

6 Démarrer les nouveaux binaires Zabbix

Démarrez les nouveaux binaires.
Consultez les fichiers journaux pour vérifier que les binaires ont démarré correctement.

Le serveur Zabbix met automatiquement à niveau la base de données.
Lors de son démarrage, le serveur Zabbix indique les versions actuelles (obligatoire et facultative) et les versions requises de la base de données.
Si la version obligatoire actuelle est antérieure à la version requise, le serveur Zabbix exécute automatiquement les correctifs de mise à niveau requis pour la base de données.
Le démarrage et le niveau d'avancement (pourcentage) de la mise à niveau de la base de données sont consignés dans le fichier journal du serveur Zabbix.
Lorsque la mise à niveau est terminée, le message « database upgrade fully completed » est écrit dans le fichier journal.
Si l'un des correctifs de mise à niveau échoue, le serveur Zabbix ne démarre pas.
Le serveur Zabbix ne démarre pas non plus si la version obligatoire actuelle de la base de données est plus récente que la version requise.
Le serveur Zabbix démarre uniquement si la version obligatoire actuelle de la base de données correspond à la version obligatoire requise.

```
8673:20161117:104750.259 current database version (mandatory/optional): 03040000/03040000
8673:20161117:104750.259 required mandatory version: 03040000
```

Avant de démarrer le serveur :

- Assurez-vous que l'utilisateur de la base de données dispose de suffisamment d'autorisations (créer une table, supprimer une table, créer un index, supprimer un index).
- Assurez-vous de disposer de suffisamment d'espace disque libre.

7 Installer la nouvelle interface web Zabbix

La version minimale requise de PHP est 8.2.0. Mettez-la à jour si nécessaire et suivez les instructions d'installation.

8 Effacer les cookies et le cache du navigateur web

Après la mise à niveau, vous devrez peut-être effacer les cookies et le cache du navigateur web pour que l'interface web de Zabbix fonctionne correctement.

Procédure de mise à jour du Proxy

1 Arrêt du proxy Zabbix

Arrêtez le proxy Zabbix.

2 Sauvegarde des fichiers de configuration et des binaires du proxy Zabbix

Faites une copie de sauvegarde des binaires du proxy Zabbix et des fichiers de configuration.

3 Installer les nouveaux binaires du proxy

Utilisez ces instructions pour compiler le proxy Zabbix à partir des sources.

4 Examiner les paramètres de configuration du proxy

Veillez à consulter les Notes de mise à niveau pour vérifier si des modifications des paramètres de configuration sont nécessaires.

Pour les nouveaux paramètres facultatifs, consultez la page Nouveautés.

5 Démarrer le nouveau proxy Zabbix

Démarrez le nouveau proxy Zabbix. Consultez les fichiers journaux pour vérifier que le proxy a démarré correctement.

Le proxy Zabbix mettra automatiquement à niveau la base de données. La mise à niveau de la base de données s'effectue de la même manière que lors du démarrage du serveur Zabbix.

Processus de mise à niveau de l'agent

> **Attention:**
> La mise à niveau des agents n'est pas obligatoire. Vous devez mettre à niveau les agents uniquement si cela est nécessaire pour accéder aux nouvelles fonctionnalités.

La procédure de mise à niveau décrite dans cette section peut être utilisée pour mettre à niveau à la fois l'agent Zabbix et l'agent Zabbix 2.

1 Arrêt de l'agent Zabbix

Arrêtez l'agent Zabbix.

2 Sauvegarde des fichiers de configuration et des binaires de l'agent Zabbix

Faites une copie de sauvegarde des binaires de l'agent Zabbix et des fichiers de configuration.

3 Installer les nouveaux binaires de l'agent

Utilisez ces instructions pour compiler l'agent Zabbix à partir des sources.

Vous pouvez également télécharger des agents Zabbix précompilés depuis la page de téléchargement de Zabbix.

4 Examiner les paramètres de configuration de l'agent

Assurez-vous de consulter les Notes de mise à niveau pour vérifier si des modifications des paramètres de configuration sont nécessaires.

Pour les nouveaux paramètres facultatifs, consultez la page Nouveautés.

5 Démarrer le nouvel agent Zabbix

Démarrez le nouvel agent Zabbix. Consultez les fichiers journaux pour vérifier que l'agent a démarré correctement.

Mise à niveau entre versions mineures

Lors d'une mise à niveau entre des versions mineures de 8.0.x (par exemple, de 8.0.1 à 8.0.3), il est nécessaire d'effectuer les mêmes actions pour le serveur/proxy/agent que lors d'une mise à niveau entre des versions majeures. La seule différence est que lors d'une mise à niveau entre des versions mineures, aucune modification n'est apportée à la base de données.

Problèmes de mise à niveau

Échappement et éléments associés Limite de caractères dépassée pour les paramètres de fonction après la mise à niveau

L'échappement correct des barres obliques inverses a été ajouté aux paramètres de chaîne des fonctions d'historique dans Zabbix 7.0.

Comme des barres obliques inverses supplémentaires sont ajoutées lors de la mise à niveau depuis des versions de Zabbix antérieures à 7.0, cela entraîne des paramètres plus longs, ce qui peut provoquer des fonctions de déclencheur défectueuses si la longueur du paramètre dépasse la taille maximale des données de 255 caractères.

Pour illustrer les problèmes pouvant survenir lors de la mise à niveau de la base de données, leur impact, ainsi que la manière de les reconnaître et de les corriger, prenons comme exemple une configuration avec 3 éléments et 2 déclencheurs.

Configuration

1. Version 6.0 de Zabbix.
2. Fichier `/tmp/ONE` contenant 114 barres obliques inverses avec un « a » à la fin :

```
\\\
```

```
find(/Zabbix server/vfs.file.contents[/tmp/ONE],1m:now,"regexp","\\\
```

```
\\\
```

2. Téléchargez la nouvelle version de l'image :

```
docker pull zabbix/zabbix-proxy-sqlite3:alpine-8.0-latest
```

3. Arrêtez le conteneur du composant Zabbix :

```
docker stop zabbix-proxy-sqlite3
```

4. Supprimez le conteneur du composant Zabbix :

Attention:
Assurez-vous d'avoir noté tous les indicateurs de configuration et les montages de volumes utilisés dans la commande `docker run` d'origine avant de continuer, car ils seront nécessaires pour recréer le conteneur.

```
docker rm zabbix-proxy-sqlite3
```

5. Démarrez un nouveau conteneur en utilisant l'image mise à jour :

```
docker run --name zabbix-proxy-sqlite3 \
  -e ZBX_SERVER_HOST=192.0.2.1 \
  -e ZBX_PROXYMODE=0 \
  -e ZBX_HOSTNAME=zabbix-proxy-sqlite3 \
  -v zabbix-proxy-data:/var/lib/zabbix/db_data \
  --init \
  -d \
  zabbix/zabbix-proxy-sqlite3:alpine-8.0-latest
```

Note:
Pour plus d'informations sur l'exécution des conteneurs Zabbix, consultez Docker (déploiement manuel).

6. Vérifiez que la mise à niveau a réussi en consultant les journaux du conteneur :

```
docker logs -f zabbix-proxy-sqlite3
```

Mise à niveau du fichier Docker Compose

Ces instructions permettent de mettre à niveau les fichiers Docker Compose utilisés pour déployer les composants Zabbix. Suivez ces instructions si vous avez installé Zabbix à l'aide de Docker Compose.

1. Vérifiez tous les conteneurs afin d'identifier leurs noms et leurs versions d'image actuelles :

```
docker ps -a

#### CONTAINER ID   IMAGE                                              (...) NAMES
#### d095f486deef   zabbix/zabbix-server-mysql:alpine-7.4-latest       (...) zabbix-docker-zabbix-server-1
#### 70627d833da0   zabbix/zabbix-web-nginx-mysql:alpine-7.4-latest    (...) zabbix-docker-zabbix-web-ngin
#### aaaa664cc185   zabbix/zabbix-server-mysql:alpine-7.4-latest       (...) zabbix-docker-server-db-init-1
#### 9ab3e77492ff   mysql:8.0-oracle                                   (...) zabbix-docker-mysql-server-1
```

2. Accédez au dépôt Docker Zabbix que vous avez cloné précédemment, récupérez les dernières mises à jour et basculez vers la branche 8.0 :

Attention:
Si vous avez modifié `.env`, `compose.yaml` ou d'autres fichiers dans votre dépôt local, un `git pull` peut les écraser. Sauvegardez toute modification locale avant d'exécuter cette étape.

```
cd zabbix-docker
git pull
git checkout 8.0
```

3. Déployez les composants Zabbix à l'aide du fichier Docker Compose mis à jour :

```
#### Avec MySQL comme base de données :
docker compose -f ./compose.yaml up -d

#### Avec PostgreSQL comme base de données :
docker compose -f ./compose_pgsql.yaml up -d
```

> **Note:**
> Une fois que tous les conteneurs ont été recréés et démarrés, la reprise de la collecte de données peut prendre 1 à 3 minutes. Pour plus d'informations sur l'exécution des conteneurs Zabbix, consultez Docker Compose.

4. Vérifiez que la mise à niveau a réussi en consultant les journaux du conteneur (remplacez `zabbix-docker-zabbix-server-1` par le nom du conteneur de votre composant) :

```
docker logs -f zabbix-docker-zabbix-server-1
```

Problèmes connus

Voir aussi : Problèmes de compilation.

Mise à niveau

Paramètre SQL mode pour une mise à niveau réussie

Le paramètre `sql_mode` dans MySQL/MariaDB doit avoir le mode « STRICT_TRANS_TABLES » défini. S'il est absent, la mise à niveau de la base de données Zabbix échouera (voir aussi ZBX-19435).

Mise à niveau avec MariaDB 10.2.1 et versions antérieures

La mise à niveau de Zabbix peut échouer si les tables de la base de données ont été créées avec MariaDB 10.2.1 ou une version antérieure, car dans ces versions, le format de ligne par défaut est compact. Ce problème peut être corrigé en modifiant le format de ligne en dynamic (voir aussi ZBX-17690).

Jeu de caractères et collation de la base de données MySQL

Les installations de Zabbix utilisant une base de données MySQL créée avec le jeu de caractères utf8mb3 et la collation utf8mb3_bin peuvent rencontrer des problèmes. Par exemple, un problème MySQL connu peut entraîner des erreurs `Row size too large (> 8126)` lors des mises à niveau du schéma de la base de données ou d'autres opérations DDL.
 Il est fortement recommandé de convertir les bases de données utilisant le jeu de caractères utf8mb3 et la collation utf8mb3_bin vers le jeu de caractères utf8mb4 et la collation utf8mb4_bin, comme décrit dans Réparer le jeu de caractères et la collation de la base de données Zabbix.

Fuite de mémoire et de descripteurs de fichiers avec libcurl 8.21.0

Un problème connu dans libcurl 8.21.0 peut provoquer des fuites de mémoire et de descripteurs de fichiers dans les fonctionnalités de Zabbix utilisant libcurl lors du traitement de connexions sécurisées par TLS.
 Une solution de contournement consiste à rétrograder libcurl vers la version 8.20.0 jusqu'à ce qu'une version contenant la correction soit disponible.

Modèles

Compatibilité des modèles dans les environnements double pile (IPv4/IPv6)

Dans les environnements double pile (systèmes configurés pour prendre en charge à la fois IPv4 et IPv6), le nom d'hôte `localhost` est généralement résolu à la fois en adresses IPv4 et IPv6. En raison de la priorisation courante d'IPv6 par rapport à IPv4 par de nombreux systèmes d'exploitation et résolveurs DNS, les modèles Zabbix peuvent ne pas fonctionner correctement si le service surveillé est configuré pour écouter uniquement sur IPv4.

Les services qui ne sont pas configurés pour écouter sur des adresses IPv6 peuvent devenir inaccessibles, entraînant des échecs de supervision. Les utilisateurs peuvent configurer l'accès correctement pour IPv4, mais rencontrer malgré tout des problèmes de connectivité en raison du comportement par défaut qui privilégie IPv6.

Une solution de contournement consiste à s'assurer que les services (Nginx, Apache, PostgreSQL, etc.) sont configurés pour écouter à la fois sur des adresses IPv4 et IPv6, et que l'accès au serveur/agent Zabbix via IPv6 est autorisé. De plus, dans les modèles et configurations Zabbix, utilisez explicitement `localhost` au lieu de `127.0.0.1` afin d'assurer la compatibilité avec IPv4 et IPv6.

Par exemple, lors de la supervision de PostgreSQL avec le modèle PostgreSQL by Zabbix agent 2, vous devrez peut-être modifier le fichier `pg_hba.conf` pour autoriser les connexions de l'utilisateur `zbx_monitor`. Si l'environnement double pile privilégie IPv6 (le système résout localhost en `::1`) et que vous configurez `localhost` mais n'ajoutez qu'une entrée IPv4 (`127.0.0.1/32`), la connexion échouera car aucune entrée IPv6 correspondante n'existe.

L'exemple suivant de fichier `pg_hba.conf` garantit que l'utilisateur `zbx_monitor` peut se connecter à n'importe quelle base de données depuis la machine locale en utilisant à la fois des adresses IPv4 et IPv6 avec différentes méthodes d'authentification :

```
### TYPE     DATABASE     USER            ADDRESS          METHOD
  host     all          zbx_monitor     localhost        trust
  host     all          zbx_monitor     127.0.0.1/32     md5
  host     all          zbx_monitor     ::1/128          scram-sha-256
```

Si nécessaire, vous pouvez également utiliser directement l'adresse IPv4 (`127.0.0.1`) lors de la configuration de la macro du modèle PostgreSQL by Zabbix agent 2 pour la chaîne de connexion.

Installation accidentelle des paquets Zabbix d'EPEL

Si le dépôt EPEL est installé et activé, l'installation des paquets Zabbix peut récupérer les versions EPEL au lieu des paquets Zabbix officiels. Pour résoudre ce problème :

1. Supprimez tous les paquets Zabbix installés depuis EPEL :

```
dnf remove zabbix-server-mysql
```

2. Excluez les paquets Zabbix d'EPEL en ajoutant la ligne suivante au fichier `/etc/yum.repos.d/epel.repo` :

```
[epel]
...
excludepkgs=zabbix*
```

3. Réinstallez le paquet officiel du serveur Zabbix :

```
dnf install zabbix-server-mysql
```

Pendant l'installation, les paquets Zabbix officiels incluent le mot `release` dans leur chaîne de version (par exemple, `7.0.0-release1.el8`), ce qui permet de les distinguer des paquets EPEL.

Paquets Zabbix pour RHEL dans des environnements Red Hat UBI

Lors de l'installation de Zabbix à partir de paquets Red Hat Enterprise Linux dans des environnements Red Hat Universal Base Image, assurez-vous d'avoir accès aux dépôts et dépendances requis. Les paquets Zabbix dépendent des bibliothèques `libOpenIPMI.so` et `libOpenIPMIposix.so`, qui ne sont fournies par aucun paquet dans les dépôts par défaut du gestionnaire de paquets activés sur les systèmes UBI, ce qui entraînera des échecs d'installation.

Les bibliothèques `libOpenIPMI.so` et `libOpenIPMIposix.so` sont disponibles dans le paquet `OpenIPMI-libs`, fourni par le dépôt `redhat-#-for-<arch>-appstream-rpms`. L'accès à ce dépôt est géré par les abonnements qui, dans le cas des environnements UBI, sont propagés en montant les répertoires de configuration des dépôts et des secrets de l'hôte RHEL dans l'espace de noms du système de fichiers du conteneur.

Pour plus d'informations, consultez ZBX-24291.

Clé de signature expirée pour les paquets RHEL

Lors de la mise à niveau de Zabbix sur Red Hat Enterprise Linux ou ses dérivés, vous pouvez rencontrer un problème de clé de signature expirée pour les paquets du dépôt Zabbix. Lorsqu'une clé de signature expire, les tentatives de vérification des signatures des paquets entraînent une erreur indiquant que le certificat ou la clé n'est plus valide. Par exemple :

```
error: Verifying a signature using certificate D9AA84C2B617479C6E4FCF4D19F2475308EFA7DD (Zabbix LLC (Jul 2
  1. Certificate 19F2475308EFA7DD invalid: certificate is not alive
      because: The primary key is not live
      because: Expired on 2024-07-04T11:41:23Z
  2. Key 19F2475308EFA7DD invalid: key is not alive
      because: The primary key is not live
      because: Expired on 2024-07-04T11:41:23Z
```

Pour résoudre ce type de problème, réinstallez manuellement le dernier paquet `zabbix-release` correspondant à votre variante spécifique de RHEL (remplacez le lien ci-dessous par le bon lien depuis le dépôt Zabbix).

Par exemple, sur **RHEL 10**, exécutez :

```
rpm -Uvh https://repo.zabbix.com/zabbix/8.0/release/rhel/10/noarch/zabbix-release-latest.el10.noarch.rpm
```

Ensuite, mettez à jour les informations du dépôt :

```
dnf update
```

Pour plus d'informations, consultez ZBX-24761.

Connexion TLS à la base de données avec MariaDB

La connexion TLS à la base de données n'est pas prise en charge avec l'option `verify_ca` pour le paramètre DBTLSConnect si MariaDB est utilisé.

Interblocages possibles avec MySQL/MariaDB

Lors d'une exécution sous forte charge, et avec plus d'un worker LLD impliqué, il est possible de rencontrer un interblocage causé par une erreur InnoDB liée à la stratégie de verrouillage des lignes (voir le bug en amont). L'erreur a été corrigée dans MySQL à partir de la version 8.0.29, mais pas dans MariaDB. Pour plus de détails, voir ZBX-21506.

Corrélation globale des événements

Les événements peuvent ne pas être corrélés correctement si l'intervalle de temps entre le premier et le deuxième événement est très court, c'est-à-dire d'une demi-seconde ou moins.

NetBSD 8.0 et versions ultérieures

Divers processus Zabbix peuvent se fermer inopinément au démarrage sur les versions 8.X et 9.X de NetBSD. Cela est dû à une taille de pile par défaut trop faible (4MB), qui doit être augmentée en exécutant :

```
ulimit -s 10240
```

Pour plus d'informations, veuillez consulter le rapport de problème associé : ZBX-18275.

Limitations des expressions régulières dans Zabbix agent 2

Zabbix agent 2 ne prend pas en charge les assertions anticipées (lookaheads) ni les assertions rétroactives (lookbehinds) dans les expressions régulières en raison des limitations de la bibliothèque standard regexp de Go.

Vérifications IPMI

Les vérifications IPMI ne fonctionneront pas avec le paquet standard de la bibliothèque OpenIPMI sur Debian antérieur à 9 (stretch) et Ubuntu antérieur à 16.04 (xenial). Pour corriger cela, recompilez la bibliothèque OpenIPMI avec OpenSSL activé, comme indiqué dans ZBX-6139.

IPMI — des hôtes non fiables peuvent provoquer le plantage d'OpenIPMI

Une vulnérabilité dans la bibliothèque OpenIPMI utilisée par Zabbix pour interroger les données IPMI peut être déclenchée par des réponses spécialement conçues provenant d'un périphérique non fiable. Un périphérique IPMI non fiable peut envoyer des données spécialement conçues qui provoquent le plantage de la bibliothèque OpenIPMI, ce qui peut à son tour entraîner l'arrêt du processus du serveur Zabbix chargé de l'interrogation IPMI.

Vérifications SSH

- Certaines distributions Linux comme Debian et Ubuntu ne prennent pas en charge les clés privées chiffrées (avec phrase de passe) si la bibliothèque libssh2 est installée à partir de paquets. Veuillez consulter ZBX-4850 pour plus de détails.
- Lors de l'utilisation de libssh 0.9.x sur certaines distributions Linux avec OpenSSH 8, les vérifications SSH peuvent occasionnellement signaler « Cannot read data from SSH server ». Cela est dû à un problème de libssh (rapport plus détaillé). L'erreur est censée avoir été corrigée dans la version stable libssh 0.9.5. Voir également ZBX-17756 pour plus de détails.
- L'utilisation du caractère de tube "|" dans le script SSH peut entraîner une erreur « Cannot read data from SSH server ». Dans ce cas, il est recommandé de mettre à niveau la version de la bibliothèque libssh. Voir également ZBX-21337 pour plus de détails.

Vérifications ODBC

- Le pilote MySQL unixODBC ne doit pas être utilisé avec le serveur Zabbix ou le proxy Zabbix compilé avec la bibliothèque de connecteur MariaDB, et inversement ; si possible, il est également préférable d'éviter d'utiliser le même connecteur que le pilote en raison d'un bug amont. Configuration recommandée :

  Connecteur PostgreSQL, SQLite ou Oracle → pilote MariaDB ou MySQL unixODBC
Connecteur MariaDB → pilote MariaDB unixODBC
Connecteur MySQL → pilote MySQL unixODBC

  Voir ZBX-7665 pour plus d'informations et les solutions de contournement disponibles.

- Les données XML interrogées depuis Microsoft SQL Server peuvent être tronquées de différentes manières sur les systèmes Linux et UNIX.

- Il a été observé que l'utilisation des vérifications ODBC pour superviser des bases de données Oracle avec différentes versions d'Oracle Instant Client pour Linux provoque le plantage du serveur Zabbix.
 Voir aussi : ZBX-18402, ZBX-20803.

- Si vous utilisez le pilote FreeTDS UnixODBC, vous devez ajouter au début d'une requête SQL l'instruction 'SET NOCOUNT ON' (par exemple, `SET NOCOUNT ON DECLARE @strsql NVARCHAR(max) SET @strsql = ....`). Sinon, l'élément de supervision de base de données dans Zabbix ne parviendra pas à récupérer les informations et renverra l'erreur "SQL query returned empty result".
 Voir ZBX-19917 pour plus d'informations.

Paramètre de méthode de requête incorrect dans les éléments

Le paramètre de méthode de requête, utilisé uniquement dans les vérifications HTTP, peut être incorrectement défini sur « 1 », une valeur non par défaut pour tous les éléments à la suite d'une mise à niveau depuis une version de Zabbix antérieure à 4.0. Pour plus de détails sur la façon de corriger cette situation, consultez ZBX-19308.

Surveillance Web et agent HTTP

Le serveur Zabbix présente une fuite de mémoire sur certaines distributions Linux en raison d'un bogue amont lorsque « SSL verify peer » est activé dans les scénarios Web ou l'agent HTTP. Veuillez consulter ZBX-10486 pour plus d'informations et les solutions de contournement disponibles.

Vérifications simples

Il existe un bogue dans les versions de **fping** antérieures à la v3.10 qui gère incorrectement les paquets de réponse d'écho dupliqués. Cela peut entraîner des résultats inattendus pour les éléments `icmpping`, `icmppingloss`, `icmppingsec`. Il est recommandé d'utiliser la dernière version de **fping**. Veuillez consulter ZBX-11726 pour plus de détails.

Erreurs lors de l'exécution de fping dans des conteneurs rootless

Lorsque des conteneurs s'exécutent en mode rootless ou dans un environnement avec des restrictions spécifiques, vous pouvez rencontrer des erreurs liées à l'exécution de fping lors des vérifications ICMP, telles que `fping: Operation not permitted` ou la perte de tous les paquets vers toutes les ressources.

Pour corriger ce problème, ajoutez `--cap-add=net_raw` aux commandes "docker run" ou "podman run".

De plus, l'exécution de fping dans des environnements non root peut nécessiter une modification de sysctl, par exemple :

```
sudo sysctl -w "net.ipv4.ping_group_range=0 1995"
```

où "1995" correspond au GID de zabbix. Pour plus de détails, consultez ZBX-22833.

Vérifications SNMP

Si le système d'exploitation OpenBSD est utilisé, un bogue de type use-after-free dans la bibliothèque Net-SNMP jusqu'à la version 5.7.3 peut provoquer un plantage du serveur Zabbix si le paramètre SourceIP est défini dans le fichier de configuration du serveur Zabbix. Comme solution de contournement, veuillez ne pas définir le paramètre SourceIP. Le même problème s'applique également à Linux, mais il n'entraîne pas l'arrêt du serveur Zabbix. Un correctif local pour le paquet net-snmp sur OpenBSD a été appliqué et sera publié avec OpenBSD 6.3.

Pics de données SNMP

Des pics dans les données SNMP ont été observés et peuvent être liés à certains facteurs physiques, comme des surtensions sur le réseau électrique. Voir ZBX-14318 pour plus de détails.

Traps SNMP

Le paquet « net-snmp-perl », nécessaire pour les traps SNMP, a été supprimé dans RHEL 8.0-8.2 ; il a été réintroduit dans RHEL 8.3.

Ainsi, si vous utilisez RHEL 8.0-8.2, la meilleure solution est de passer à RHEL 8.3.

Veuillez également consulter ZBX-17192 pour plus d'informations.

Plantage du processus d'alerte dans RHEL 7

Des cas de plantage d'un processus d'alerte du serveur Zabbix ont été constatés dans RHEL 7. Veuillez consulter ZBX-10461 pour plus de détails.

Mise à niveau de Zabbix agent 2 (6.0.5 ou version antérieure)

Lors de la mise à niveau de Zabbix agent 2 (version 6.0.5 ou antérieure) à partir de paquets, une erreur de conflit de fichiers liée à un plugin peut se produire. Pour corriger cette erreur, sauvegardez la configuration de votre agent 2 (si nécessaire), désinstallez agent 2, puis réinstallez-le.

Sur les systèmes basés sur RHEL, exécutez :

```
dnf remove zabbix-agent2
dnf install zabbix-agent2
```

Sur les systèmes basés sur Debian, exécutez :

```
apt remove zabbix-agent2
apt install zabbix-agent2
```

Pour plus d'informations, consultez ZBX-23250.

Changement des paramètres régionaux de l'interface

Il a été observé que les paramètres régionaux de l'interface peuvent changer sans logique apparente : certaines pages (ou parties de pages) sont affichées dans une langue, tandis que d'autres pages (ou parties de pages) le sont dans une autre langue. En

général, le problème peut apparaître lorsqu'il y a plusieurs utilisateurs, dont certains utilisent un paramètre régional et d'autres un autre.

Une solution connue consiste à désactiver le multithreading dans PHP et Apache.

Le problème est lié au fonctionnement de la définition des paramètres régionaux dans PHP : les informations de paramètres régionaux sont conservées par processus, et non par thread. Ainsi, dans un environnement multithread, lorsque plusieurs projets sont exécutés par le même processus Apache, il est possible que les paramètres régionaux soient modifiés dans un autre thread, ce qui change la manière dont les données peuvent être traitées dans le thread Zabbix.

Pour plus d'informations, consultez les rapports de problèmes associés :

- ZBX-10911 (Problème de changement des paramètres régionaux de l'interface)
- ZBX-16297 (Problème de traitement des nombres dans les graphiques utilisant la fonction `bcdiv` des fonctions BC Math)

Graphiques

Problèmes des graphiques (classiques)

Si vous rencontrez des problèmes avec les graphiques classiques, il est recommandé de mettre à jour la bibliothèque GD (libgd) vers la version 2.3.3-13 ou ultérieure, ainsi que PHP vers la version 8.0.19, 8.1.33, 8.2.29, 8.3.25, 8.4.12 ou ultérieure.

Heure d'été

Les changements liés à l'heure d'été (DST) entraînent des irrégularités lors de l'affichage des libellés de l'axe X (duplication de dates, dates manquantes, etc.).

Agrégation par somme

Lors de l'utilisation de l'agrégation par somme dans un graphique pour une période inférieure à une heure, les graphiques affichent des valeurs incorrectes (multipliées) lorsque les données proviennent des tendances.

Chevauchement du texte

Pour certaines langues de l'interface (par exemple, le japonais), les polices locales peuvent provoquer un chevauchement du texte dans la légende du graphique. Pour éviter cela, utilisez la version 2.3.0 (ou ultérieure) de l'extension PHP GD.

Surveillance des fichiers journaux

Les éléments `log[]` et `logrt[]` relisent de manière répétée le fichier journal depuis le début si le système de fichiers est plein à 100 % et que des données sont ajoutées au fichier journal (voir ZBX-10884 pour plus d'informations).

Requêtes MySQL lentes

Le serveur Zabbix génère des requêtes `SELECT` lentes en cas d'absence de valeurs pour les éléments. Ce problème est connu pour se produire dans les versions MySQL 5.6/5.7 (pour une discussion plus approfondie, voir ZBX-10652) et, dans certains cas spécifiques, peut également se produire dans des versions ultérieures de MySQL. Une solution de contournement consiste à désactiver l'optimiseur `index_condition_pushdown` ou `prefer_ordering_index` dans MySQL. Notez toutefois que cette solution de contournement peut ne pas résoudre tous les problèmes liés aux requêtes lentes.

Paramètres de filtre persistants à partir des liens

Lors de l'ouverture d'un lien vers une page du frontend Zabbix contenant des paramètres de filtre, y compris le sélecteur de temps, le filtre est automatiquement enregistré dans la base de données pour l'utilisateur, en remplaçant le filtre précédemment enregistré et/ou les paramètres du sélecteur de temps pour cette page. Ces paramètres restent actifs jusqu'à ce que l'utilisateur les mette à jour ou les réinitialise manuellement.

Problème d'adresse IPv6 dans les traps SNMPv3

En raison d'un bug de net-snmp, l'adresse IPv6 peut ne pas s'afficher correctement lors de l'utilisation de SNMPv3 dans les traps SNMP. Pour plus de détails et une solution de contournement possible, consultez ZBX-14541.

Adresse IP IPv6 longue tronquée dans les informations de connexion échouée

Un message de tentative de connexion échouée n'affichera que les 39 premiers caractères d'une adresse IP stockée, car il s'agit de la limite de caractères du champ dans la base de données. Cela signifie que les adresses IP IPv6 de plus de 39 caractères seront affichées de manière incomplète.

Vérifications de l'agent Zabbix sous Windows

Des entrées DNS inexistantes dans un paramètre `Server` du fichier de configuration de l'agent Zabbix (`zabbix_agentd.conf`) peuvent augmenter le temps de réponse de l'agent Zabbix sous Windows. Cela se produit parce que le démon de mise en cache DNS de Windows ne met pas en cache les réponses négatives pour les adresses IPv4. En revanche, pour les adresses IPv6, les réponses négatives sont mises en cache, de sorte qu'une solution de contournement possible consiste à désactiver IPv4 sur l'hôte.

Export/import YAML

Il existe quelques problèmes connus avec l'export/import YAML :

- Les messages d'erreur ne sont pas traduisibles ;
- Un JSON valide avec une extension de fichier .yaml ne peut parfois pas être importé ;
- Les dates lisibles par l'humain non entre guillemets sont automatiquement converties en horodatages Unix.

Assistant de configuration sur SUSE avec NGINX et php-fpm

L'assistant de configuration de l'interface ne peut pas enregistrer le fichier de configuration sur SUSE avec NGINX + php-fpm. Cela est dû à un paramètre dans l'unité /usr/lib/systemd/system/php-fpm.service, qui empêche Zabbix d'écrire dans /etc. (introduit dans PHP 7.4).

Deux solutions de contournement sont disponibles :

- Définir l'option ProtectSystem sur 'true' au lieu de 'full' dans l'unité systemd php-fpm.
- Enregistrer manuellement le fichier `/etc/zabbix/web/zabbix.conf.php`.

Transfert de l'en-tête Authorization

Dans certains cas, Apache ou NGINX peuvent empêcher l'en-tête Authorization des requêtes API d'atteindre Zabbix. Cela peut entraîner des problèmes d'authentification lors de l'utilisation de l'API Zabbix ou de services d'authentification unique (SSO), tels que SAML avec Okta.

Pour résoudre ce problème, mettez à jour la configuration de votre serveur web.

Pour **Apache**, si vous l'utilisez comme proxy inverse (configuration non CGI), ajoutez la directive suivante à `/etc/httpd/conf/httpd.conf` (sur les systèmes basés sur RHEL) ou à `/etc/apache2/apache2.conf` (sur Debian/Ubuntu) :

```
SetEnvIfNoCase ^Authorization$ "(.+)" HTTP_AUTHORIZATION=$1
```

Si Apache exécute directement des scripts pour traiter les requêtes (par exemple, en utilisant mod_cgi), ajoutez plutôt la directive suivante :

```
CGIPassAuth On
```

En revanche, **NGINX** gère automatiquement l'en-tête Authorization. Cependant, si NGINX agit comme proxy inverse, vous pouvez transférer explicitement l'en-tête Authorization en ajoutant les directives suivantes à `/etc/nginx/nginx.conf` (pour l'emplacement de votre interface Zabbix) :

```
...
location / {
...
    proxy_set_header Authorization $http_authorization;
    proxy_pass http://backend_server;
...
}
```

Après avoir mis à jour la configuration, redémarrez votre serveur web.

Pour plus de détails, voir :

- ZBX-22952
- Apache 2.4 + PHP-FPM and Authorization headers
- Directives SetEnvIfNoCase et CGIPassAuth
- NGINX Reverse Proxy

Chromium pour le service web Zabbix sur Ubuntu 20

Bien que, dans la plupart des cas, le service web Zabbix puisse fonctionner avec Chromium, sur Ubuntu 20.04, l'utilisation de Chromium provoque l'erreur suivante :

```
Cannot fetch data: chrome failed to start:cmd_run.go:994:
WARNING: cannot create user data directory: cannot create
"/var/lib/zabbix/snap/chromium/1564": mkdir /var/lib/zabbix: permission denied
Sorry, home directories outside of /home are not currently supported. See https://forum.snapcraft.io/t/112
```

Cette erreur se produit parce que `/var/lib/zabbix` est utilisé comme répertoire personnel de l'utilisateur « zabbix ».

Codes d'erreur MySQL personnalisés

Lorsque Zabbix détecte que la base de données backend est inaccessible, il envoie une notification et continue à tenter de se connecter. Pour certains moteurs de base de données, des codes d'erreur spécifiques sont reconnus. Dans MySQL, ces codes d'erreur reconnus incluent :

- CR_CONN_HOST_ERROR

- CR_SERVER_GONE_ERROR
- CR_CONNECTION_ERROR
- CR_SERVER_LOST
- CR_UNKNOWN_HOST
- ER_SERVER_SHUTDOWN
- ER_ACCESS_DENIED_ERROR
- ER_ILLEGAL_GRANT_FOR_TABLE
- ER_TABLEACCESS_DENIED_ERROR
- ER_UNKNOWN_ERROR

De plus, lors de l'utilisation de Zabbix avec une installation MySQL sur Azure, le message d'erreur générique *[9002] Some errors occurred* peut apparaître dans les journaux Zabbix. Ce message est envoyé par la base de données au serveur ou au proxy Zabbix. Pour déterminer la cause de l'erreur, veuillez consulter les journaux Azure.

Expressions régulières invalides après le passage à PCRE2

Dans Zabbix 6.0, la prise en charge de PCRE2 a été ajoutée. Bien que PCRE soit toujours pris en charge, les paquets d'installation Zabbix pour RHEL 7 et versions ultérieures, SLES (toutes les versions), Debian 9 et versions ultérieures, Ubuntu 16.04 et versions ultérieures ont été mis à jour pour utiliser PCRE2. Tout en offrant de nombreux avantages, le passage à PCRE2 peut rendre certains modèles d'expressions régulières PCRE existants invalides ou modifier leur comportement. En particulier, cela affecte le modèle *[\w-\.]*. Afin de rendre cette expression régulière à nouveau valide sans en modifier la sémantique, remplacez l'expression par *[-\w\.]* . Cela se produit parce que PCRE2 traite le tiret comme un délimiteur, créant une plage à l'intérieur d'une classe de caractères.

Erreur du widget Geomap

Les cartes du widget Geomap peuvent ne pas se charger correctement si vous avez effectué une mise à niveau depuis une ancienne version de Zabbix avec NGINX et que vous n'avez pas basculé vers le nouveau fichier de configuration NGINX pendant la mise à niveau.

Pour corriger le problème, vous pouvez supprimer l'ancien fichier de configuration, utiliser le fichier de configuration du package de la version actuelle, puis le reconfigurer comme décrit dans les instructions de téléchargement dans la section *e. Configure PHP for Zabbix frontend*.

Sinon, vous pouvez modifier manuellement un fichier de configuration NGINX existant (généralement, `/etc/zabbix/nginx.conf`). Pour ce faire, ouvrez le fichier et localisez le bloc suivant :

```
location ~ /(api\/|conf[^\.]|include|locale|vendor) {
        deny            all;
        return          404;
}
```

Remplacez ensuite ce bloc par :

```
location ~ /(api\/|conf[^\.]|include|locale) {
        deny            all;
        return          404;
}

location /vendor {
        deny            all;
        return          404;
}
```

Prétraitement — les variables globales sont non sécurisées

Le JavaScript de prétraitement s'exécute à chaque requête, mais les affectations à des identifiants non déclarés (par exemple `secret = value`) créent des variables globales implicites qui peuvent persister au-delà de l'exécution en cours. Le stockage de données sensibles (jetons, mots de passe, etc.) dans des variables globales implicites augmente le risque d'exposition accidentelle ou de réutilisation lors d'exécutions de prétraitement ultérieures ou par d'autres intégrations s'exécutant dans le même environnement.

Ne vous appuyez pas sur des variables globales implicites. Déclarez toujours les variables avec `var` ou `const`, et évitez d'attacher des secrets à des objets globaux (par exemple `globalThis` ou `window`). Il n'existe aucun moyen pris en charge de remplacer les objets globaux intégrés depuis le prétraitement.

Exemple sécurisé :

```
var apiToken = payload.token;
var count = 1;
return JSON.stringify({ token: apiToken, calls: count });
```

Groupes de processeurs sous Windows

La documentation de Microsoft indique que les systèmes disposant de moins de 64 processeurs logiques n'ont toujours qu'un seul groupe de processeurs, le groupe 0. Cependant, des utilisateurs de Zabbix ont signalé un bogue rare ZBX-20260, lorsque deux groupes de processeurs sont présents sur des systèmes ayant 64 processeurs logiques ou moins. Cela a eu pour conséquence que les compteurs de performance "\Processor(n)" n'étaient disponibles que pour un seul des deux groupes de processeurs. La cause racine exacte de ce bogue n'est pas connue. Cependant, un cas similaire a été décrit sur stackoverflow.com, où la cause racine était liée à l'interopérabilité entre le BIOS et Windows.

Limites du filtrage avec les interclassements utf8mb4

Les filtres (par exemple, dans *Collecte de données* > *Maintenance*) peuvent ne pas fonctionner correctement lorsqu'ils sont appliqués à des entités contenant certains caractères Unicode (par exemple, 𝓍, 𝔁). Ce problème est dû à la manière dont l'interclassement utf8mb4_bin par défaut des bases de données MySQL ou MariaDB gère le tri et la comparaison des caractères Unicode.

Pour remédier à cette limitation, les utilisateurs peuvent modifier l'interclassement des colonnes de la base de données vers des alternatives telles que utf8mb4_0900_bin, utf8mb4_0900_ai_ci ou utf8mb4_unicode_520_ci. Notez toutefois que la modification de l'interclassement peut entraîner un comportement inattendu dans la gestion des espaces vides, ainsi que dans le tri et le filtrage d'autres caractères.

Pour plus d'informations sur la modification des interclassements, consultez la documentation MySQL ou la documentation MariaDB. Pour plus de détails sur les différences entre interclassements, consultez Jeux de caractères Unicode dans la documentation MySQL.

Profilage d'une utilisation excessive de la mémoire avec tcmalloc

Si vous pensez que votre installation Zabbix utilise trop de mémoire, vous pouvez utiliser la fonctionnalité de profilage mémoire de tcmalloc pour analyser la consommation mémoire du serveur/proxy Zabbix.

1. Lors de l'installation de Zabbix à partir des sources, configurez des indicateurs supplémentaires :

```
export CFLAGS="-std=gnu99 -g -O0"
```

L'indicateur `-std=gnu99` est requis pour compiler le serveur Zabbix, le proxy Zabbix ou l'agent Zabbix. L'indicateur `-g` ajoute des informations de débogage supplémentaires, tandis que `-O0` désactive les optimisations, qui peuvent interférer avec le profilage de tcmalloc.

2. Définissez les variables d'environnement suivantes avant de démarrer le serveur Zabbix. Ces variables indiquent à tcmalloc comment suivre et signaler l'utilisation de la mémoire :

```
LD_PRELOAD="/usr/lib/aarch64-linux-gnu/libtcmalloc.so" \
HEAPPROFILE=./heap_profile \
HEAP_PROFILE_ALLOCATION_INTERVAL=0 \
HEAP_PROFILE_INUSE_INTERVAL=4294967296 \
HEAPPROFILESIGNAL=5 \
MALLOCSTATS=1 \
./sbin/zabbix_server -f -c /etc/zabbix/zabbix_server.conf
```

3. Déclenchez un vidage du profil en envoyant le signal 5 au processus cible. Remplacez 1234 par l'identifiant de processus réel (PID) :

```
kill -5 1234
```

4. Affichez le profil généré :

```
pprof-symbolize -text ./sbin/zabbix_server ./heap_profile.0001.heap

Using local file ./sbin/zabbix_server.
Using local file ./heap_profile.0001.heap.
Total: 1078.1 MB
  1076.8  99.9%  99.9%   1076.8  99.9% zbx_malloc2
     1.0   0.1% 100.0%      1.0   0.1% __GI___strdup
     0.2   0.0% 100.0%      0.2   0.0% CRYPTO_zalloc@@OPENSSL_3.0.0
     0.1   0.0% 100.0%      0.1   0.0% OPENSSL_LH_insert@@OPENSSL_3.0.0
     0.0   0.0% 100.0%      0.0   0.0% zbx_realloc2
     0.0   0.0% 100.0%      0.1   0.0% PKCS7_decrypt@@OPENSSL_3.0.0
     0.0   0.0% 100.0%      0.0   0.0% find_best_tree_node
     0.0   0.0% 100.0%      0.0   0.0% CRYPTO_strndup@@OPENSSL_3.0.0
     ...
```

```
     0.0    0.0% 100.0%      0.0    0.0% preprocessing_flush_value
     0.0    0.0% 100.0%   1074.0   99.6% preprocessor_add_request
```

Dans cet exemple, zbx_malloc2 est responsable de presque toutes les allocations mémoire.

Voir aussi :

- ZBX-25050 et ZBX-25584 pour les rapports de problème associés.
- GCC Option Summary sur les options de compilation (`-std=gnu99`, `-g`, `-O0`, etc.).
- La documentation Gperftools Heap Profiler sur les variables d'environnement pour le profilage tcmalloc.

Réplication de groupe MySQL 8.0 en mode multi-primaire

Lors de l'utilisation de la réplication de groupe MySQL 8.0 en mode multi-primaire, vous pouvez rencontrer une erreur lors de la validation des transactions, similaire à la suivante :

```
1531697:20250128:064734.697 query [txnlev:1] [update alerts set status=1,retries=0,error='' where alertid=
1531697:20250128:064734.713 query [txnlev:1] [commit;]
1531697:20250128:064734.753 [Z3005] query failed: [3101] Plugin instructed the server to rollback the curr
```

Cette erreur semble être déclenchée par des problèmes liés aux opérations de rollback impliquant des contraintes de clé étrangère.

Voir aussi :

- ZBX-26060 pour le rapport de problème associé.
- MySQL Bug #96758 "Rollbacks with Foreign Keys on single node" pour le problème en amont.

Connexion et systemd

Nous recommandons de créer un utilisateur *zabbix* en tant qu'utilisateur système, c'est-à-dire sans possibilité de se connecter. Certains utilisateurs ignorent cette recommandation et utilisent le même compte pour se connecter (par exemple via SSH) à l'hôte exécutant Zabbix. Cela peut provoquer l'arrêt brutal du démon Zabbix lors de la déconnexion. Dans ce cas, vous verrez quelque chose comme ce qui suit dans le journal du serveur Zabbix :

```
zabbix_server [27730]: [file:'selfmon.c',line:375] lock failed: [22] Invalid argument
zabbix_server [27716]: [file:'dbconfig.c',line:5266] lock failed: [22] Invalid argument
zabbix_server [27706]: [file:'log.c',line:238] lock failed: [22] Invalid argument
```

et dans le journal de l'agent Zabbix :

```
zabbix_agentd [27796]: [file:'log.c',line:238] lock failed: [22] Invalid argument
```

Cela se produit en raison du paramètre systemd par défaut `RemoveIPC=yes`, configuré dans `/etc/systemd/logind.conf`. Lorsque vous vous déconnectez du système, les sémaphores précédemment créés par Zabbix sont supprimés, ce qui provoque l'arrêt brutal.

Extrait de la documentation systemd :

```
RemoveIPC=

Controls whether System V and POSIX IPC objects belonging to the user shall be removed when the
user fully logs out. Takes a boolean argument. If enabled, the user may not consume IPC resources
after the last of the user's sessions terminated. This covers System V semaphores, shared memory
and message queues, as well as POSIX shared memory and message queues. Note that IPC objects of the
root user and other system users are excluded from the effect of this setting. Defaults to "yes".
```

Il existe 2 solutions à ce problème :

1. (recommandée) N'utilisez plus le compte *zabbix* pour autre chose que les processus Zabbix et créez un compte dédié pour les autres activités.
2. (non recommandée) Définissez `RemoveIPC=no` dans `/etc/systemd/logind.conf` et redémarrez le système. Notez que `RemoveIPC` est un paramètre applicable à l'ensemble du système ; sa modification affectera tout le système.

Prérequis

Matériel

Mémoire

Zabbix nécessite à la fois de la mémoire physique et de l'espace disque. La quantité d'espace disque nécessaire dépend évidemment du nombre d'hôtes et de paramètres surveillés. Si vous prévoyez de conserver un long historique des paramètres surveillés, vous devriez envisager au moins quelques gigaoctets afin de disposer de suffisamment d'espace pour stocker l'historique dans la base de données. Chaque processus démon Zabbix nécessite plusieurs connexions à un serveur de base de données. La quantité de mémoire allouée à la connexion dépend de la configuration du moteur de base de données.

Note:
Plus vous disposez de mémoire physique, plus la base de données (et donc Zabbix) fonctionne rapidement.

CPU

Zabbix, et en particulier la base de données Zabbix, peuvent nécessiter d'importantes ressources CPU en fonction du nombre de paramètres surveillés et du moteur de base de données choisi.

Autre matériel

Un port de communication série et un modem GSM série sont nécessaires pour utiliser la prise en charge des notifications par SMS dans Zabbix. Un convertisseur USB-série fonctionnera également.

Exemples de configuration matérielle

Le tableau fournit des exemples de configuration matérielle, en supposant une plateforme **Linux/BSD/Unix**.

Il s'agit d'exemples de dimensionnement et de configuration matérielle pour commencer. Chaque installation Zabbix est unique. Veillez à évaluer les performances de votre système Zabbix dans un environnement de préproduction ou de développement, afin de bien comprendre vos besoins avant de déployer l'installation Zabbix dans son environnement de production.

Taille de l'installation	Métriques surveillées[1]	Cœurs CPU/vCPU	Mémoire (Gio)	Base de données	Amazon EC2[2]
Petite	1 000	2	8	MySQL Server, Percona Server, MariaDB Server, PostgreSQL	m6i.large/m6g.large
Moyenne	10 000	4	16	MySQL Server, Percona Server, MariaDB Server, PostgreSQL	m6i.xlarge/m6g.xlarge
Grande	100 000	16	64	MySQL Server, Percona Server, MariaDB Server, PostgreSQL	m6i.4xlarge/m6g.4xlarge
Très grande	1 000 000	32	96	MySQL Server, Percona Server, MariaDB Server, PostgreSQL	m6i.8xlarge/m6g.8xlarge

[1] À des fins de dimensionnement, 1 métrique surveillée est approximativement représentée par la formule 1 métrique = 1 élément + 1 déclencheur + 1 graphique. Il s'agit uniquement d'une règle d'estimation et non d'une formule permettant de calculer le nombre de métriques surveillées.
 [2] Exemple utilisant des instances EC2 Amazon à usage général, avec une architecture ARM64 ou x86_64. Un type d'instance approprié, optimisé pour le calcul, la mémoire ou le stockage, doit être sélectionné lors de l'évaluation et des tests de l'installation Zabbix avant son installation dans l'environnement de production.

Note:
La configuration réelle dépend fortement du nombre d'éléments actifs et des fréquences d'actualisation (voir la section Taille de la base de données de cette page pour plus de détails). Il est fortement recommandé d'exécuter la base de données sur un serveur distinct pour les grandes installations.

Plateformes prises en charge

En raison des exigences de sécurité et de la nature critique du serveur de supervision, UNIX est le seul système d'exploitation capable de fournir de manière constante les performances, la tolérance aux pannes et la résilience nécessaires. Zabbix fonctionne sur les versions leaders du marché.

Les composants Zabbix sont disponibles et testés pour les plateformes suivantes :

Plateforme	Serveur	Agent	Agent 2	Commentaires
Linux	x	x	x	
Windows	-	x	x	L'agent Zabbix est pris en charge sur toutes les versions de bureau et serveur depuis Windows XP (64 bits)/Server 2003. Zabbix agent 2 est pris en charge sur toutes les versions de bureau et serveur depuis Windows 10 (32 bits)/Server 2016, car il est compilé uniquement avec une version de Go prise en charge afin d'éviter des vulnérabilités de sécurité critiques. Depuis Go 1.21, les versions minimales requises de Windows ont été relevées, ce qui fait de Windows 10/Server 2016 la version minimale pour Zabbix agent 2.
macOS	x	x	-	
IBM AIX	x	x	-	L'agent Zabbix ne fonctionne pas sur les plateformes AIX antérieures aux versions 6.1 TL07 / 7.1 TL01.
FreeBSD	x	x	-	
OpenBSD	x	x	-	
Solaris	x	x	-	
NetBSD	x	x	-	
HP-UX	x	x	-	

Note:
Le serveur/l'agent Zabbix peut fonctionner sur d'autres systèmes d'exploitation de type Unix.

Attention:
Zabbix désactive les core dumps s'il est compilé avec le chiffrement et ne démarre pas si le système n'autorise pas la désactivation des core dumps.

Logiciels requis

Zabbix repose sur des serveurs web modernes, des moteurs de base de données de premier plan et le langage de script PHP.

Logiciels externes tiers environnants

Lorsqu'il est indiqué comme obligatoire, le logiciel/la bibliothèque requis est strictement nécessaire.
Les éléments facultatifs sont nécessaires pour prendre en charge certaines fonctions spécifiques.

Logiciel	Statut obligatoire	Versions prises en charge	Commentaires
MySQL/Percona	L'un des deux	8.4.0-9.7.X	Requis si MySQL (ou Percona) est utilisé comme base de données principale de Zabbix. Le moteur InnoDB est requis. Nous recommandons d'utiliser la bibliothèque C API (libmysqlclient) pour compiler le serveur/proxy. Nous recommandons vivement d'utiliser uniquement des versions LTS dans les environnements de production. Les versions indiquées comme « rolling release », « early access » ou similaires ne sont pas recommandées.

Logiciel	Statut obligatoire	Versions prises en charge	Commentaires
MariaDB		10.11.00-12.3.X	Le moteur InnoDB est requis. Nous recommandons d'utiliser la bibliothèque MariaDB Connector/C pour compiler le serveur/proxy. Voir également : Interblocages possibles avec MariaDB et Accès aux éléments de l'interface avec MariaDB 10.5.1-10.5.9. Nous recommandons vivement d'utiliser uniquement des versions LTS dans les environnements de production. Les versions indiquées comme « rolling release », « early access » ou similaires ne sont pas recommandées.
PostgreSQL		15.0-18.X	Requis si PostgreSQL est utilisé comme base de données principale de Zabbix. Selon la taille de l'installation, il peut être nécessaire d'augmenter la propriété de configuration *work_mem* de PostgreSQL (4 Mo par défaut), afin que la quantité de mémoire utilisée par la base de données pour une opération donnée soit suffisante et que l'exécution des requêtes ne prenne pas trop de temps.
TimescaleDB pour PostgreSQL		2.20.X-2.29.X	Requis si TimescaleDB est utilisé comme extension de base de données PostgreSQL. Veillez à installer TimescaleDB Community Edition, qui prend en charge la compression. Consultez la documentation de TimescaleDB pour plus d'informations sur la compatibilité des versions de PostgreSQL et de TimescaleDB.
Elasticsearch		7.X	Requis si Elasticsearch est utilisé comme base de données principale de Zabbix. Elasticsearch peut être utilisé uniquement pour stocker les données d'historique et est pris en charge uniquement avec les serveurs Zabbix. La prise en charge d'Elasticsearch est actuellement expérimentale. Voir également les logiciels requis pour le serveur/proxy.
ClickHouse		26.3.0.0-26.9.X.X	Requis si ClickHouse est utilisé comme base de données principale de Zabbix pour stocker les données d'historique (pris en charge uniquement avec les serveurs Zabbix) ou comme source de données de télémétrie. Pour utiliser ClickHouse comme source de données de télémétrie, ClickHouse 26.8.17.4 ou une version ultérieure est requis. L'heure système du serveur Zabbix et du serveur ClickHouse doit également être synchronisée ; sinon, les requêtes de télémétrie peuvent omettre des données ou renvoyer deux fois les mêmes données.
SQLite	Facultatif	3.3.5-3.53.X	SQLite est pris en charge uniquement avec les proxies Zabbix. Requis si SQLite est utilisé comme base de données du proxy Zabbix.
smartmontools		7.1 ou version ultérieure	Requis pour l'agent Zabbix 2.
who			Requis pour le plugin de comptage des utilisateurs.
dpkg			Requis pour le plugin system.sw.packages.
pkgtool			Requis pour le plugin system.sw.packages.
rpm			Requis pour le plugin system.sw.packages.
pacman			Requis pour le plugin system.sw.packages.
q applets			`qlist` et `qsize`, qui font partie des q applets, sont requis pour le plugin system.sw.packages sur Gentoo Linux.

Note:
Bien que Zabbix puisse fonctionner avec les bases de données disponibles dans les systèmes d'exploitation, nous recommandons, pour une expérience optimale, d'utiliser des bases de données installées depuis les dépôts officiels des développeurs de bases de données.

Interface

La largeur minimale d'écran prise en charge par l'interface Zabbix est de 1200px.

Lorsqu'il est indiqué comme obligatoire, le logiciel ou la bibliothèque requis est strictement nécessaire.
Les éléments facultatifs sont nécessaires pour prendre en charge certaines fonctions spécifiques.

Logiciel	Statut obligatoire	Versions prises en charge	Commentaires
PHP	Oui	8.2.0 - 8.5.X	
Apache	L'un des deux	2.4 ou version ultérieure	
Nginx		1.20 ou version ultérieure	
MySQL	L'un des deux	Voir Logiciels tiers externes environnants	
PostgreSQL			
Extensions PHP			
mysqli	Oui		Requis si MySQL est utilisé comme base de données principale du serveur Zabbix.
pgsql			Requis si PostgreSQL est utilisé comme base de données principale du serveur Zabbix.
bcmath			php-bcmath (*--enable-bcmath*)
mbstring			php-mbstring (*--enable-mbstring*)
sockets			php-net-socket (*--enable-sockets*) ; requis pour la prise en charge des scripts utilisateur.
gd		2.0.28 ou version ultérieure	php-gd (s'il est fourni sous forme de paquet distinct par le distributeur) ; l'extension PHP GD doit prendre en charge les images PNG (*--with-png-dir*), les images JPEG (*--with-jpeg-dir*) et FreeType 2 (*--with-freetype-dir*). La version 2.3.0 ou ultérieure peut être requise pour éviter un éventuel chevauchement de texte dans les graphiques pour certaines langues de l'interface.
libxml		2.6.15 ou version ultérieure	php-xml (s'il est fourni sous forme de paquet distinct par le distributeur)
xmlwriter			php-xmlwriter (s'il est fourni sous forme de paquet distinct par le distributeur)
xmlreader			php-xmlreader (s'il est fourni sous forme de paquet distinct par le distributeur)
ctype			php-ctype (*--enable-ctype*)
session			php-session (s'il est fourni sous forme de paquet distinct par le distributeur)
ldap	Non		php-ldap ; requis pour l'authentification LDAP.
openssl			php-openssl ; requis pour l'authentification SAML.
gettext			php-gettext (*--with-gettext*) ; requis pour les traductions.
cURL		7.19.4 ou version ultérieure	php-curl ; requis pour Duo Universal Prompt MFA, l'authentification SMTP et le test de la source de données de télémétrie.

Bibliothèques d'interface tierces fournies avec Zabbix :

Bibliothèque	Statut obligatoire	Version fournie	Commentaires
Bibliothèque JavaScript jQuery	Oui	3.6.0	Bibliothèque JavaScript qui simplifie le développement compatible avec plusieurs navigateurs.
jQuery UI		1.12.1	Ensemble d'interactions, d'effets, de widgets et de thèmes d'interface utilisateur basé sur jQuery.
SAML PHP Toolkit		4.3.1	Boîte à outils PHP qui ajoute la prise en charge de l'authentification SAML 2.0 afin de permettre la connexion à Zabbix.
Duo Universal PHP library		1.1.2	Bibliothèque PHP qui ajoute la prise en charge de l'authentification multifacteur Duo Universal Prompt pour Zabbix.
Symfony Yaml Component		5.1.0	Ajoute la prise en charge de l'exportation et de l'importation des éléments de configuration Zabbix au format YAML.

Note:
Zabbix peut également fonctionner avec des versions antérieures d'Apache, de MySQL et de PostgreSQL.

Attention:
Pour les polices autres que DejaVu, la fonction PHP imagerotate peut être requise. Si elle est absente, ces polices peuvent être rendues incorrectement lors de l'affichage d'un graphique. Cette fonction est disponible uniquement si PHP est compilé avec GD intégré, ce qui n'est pas le cas dans Debian et d'autres distributions.

Bibliothèques tierces utilisées pour écrire et déboguer le code de l'interface Zabbix :

Bibliothèque	Statut obligatoire	Version minimale	Description
Composer	Non	2.4.1	Gestionnaire de paquets au niveau applicatif pour PHP, qui fournit un format standard pour gérer les dépendances des logiciels PHP et les bibliothèques requises.
PHPUnit		8.5.29	Framework de tests unitaires PHP destiné à tester l'interface Zabbix.
SASS		3.4.22	Langage de script préprocesseur interprété et compilé en feuilles de style en cascade (CSS).

Navigateur web côté client

Les cookies et JavaScript doivent être activés.

Les dernières versions stables de Google Chrome, Mozilla Firefox, Microsoft Edge, Apple Safari et Opera sont prises en charge.

Warning:
La politique de même origine pour les IFrames est appliquée, ce qui signifie que Zabbix ne peut pas être placé dans des cadres sur un domaine différent.

 Néanmoins, les pages placées dans un cadre Zabbix auront accès à l'interface web de Zabbix (via JavaScript) si la page placée dans le cadre et l'interface web de Zabbix se trouvent sur le même domaine. Une page telle que `http://secure-zabbix.com/cms/page.html`, si elle est placée dans des tableaux de bord sur `http://secure-zabbix.com/zabbix/`, aura un accès JS complet à Zabbix.

Serveur/proxy

Lorsqu'il est indiqué comme obligatoire, le logiciel ou la bibliothèque requis est strictement nécessaire. Les éléments facultatifs sont nécessaires pour prendre en charge certaines fonctions spécifiques.

Exigence	Statut obligatoire	Description
libpcre2	Oui	La bibliothèque PCRE2 est requise pour la prise en charge des expressions régulières compatibles avec Perl (PCRE). PCRE2 v10.x est pris en charge.
libevent		Requis pour la communication interprocessus. Version 2.0.10 ou ultérieure.
libevent-pthreads		Requis pour la communication interprocessus.
libpthread		Requis pour la prise en charge des mutex et des verrous en lecture-écriture (peut faire partie de libc).
libresolv		Requis pour la résolution DNS (peut faire partie de libc).

Exigence	Statut obligatoire	Description
libiconv		Requis pour la conversion de l'encodage et du format du texte (peut faire partie de libc). Obligatoire pour le serveur Zabbix sous Linux.
libz		Requis pour la prise en charge de la compression.
libm		Bibliothèque mathématique. Requise uniquement par le serveur Zabbix.
libmysqlclient	L'un des deux	Requis si MySQL est utilisé.
libmariadb		Requis si MariaDB est utilisé.
libpq5		Requis si PostgreSQL est utilisé ; la version de *libpq5* doit être identique ou supérieure à celle de la base de données PostgreSQL utilisée.
libsqlite3		Requis si SQLite est utilisé. Requis uniquement pour le proxy Zabbix.
libOpenIPMI	Non	Requis pour la prise en charge d'IPMI. Requis uniquement pour le serveur Zabbix.
libssh2 ou *libssh*		Requis pour les vérifications SSH. Version 1.8.0 ou ultérieure (libssh2) ; 0.9.0 ou ultérieure (libssh).
libcurl		Requis pour les fonctionnalités suivantes : - éléments de supervision Web, de supervision VMware et de type agent HTTP (pour tous : version 7.19.1 ou ultérieure) ; - éléments web.page.* de l'agent Zabbix (version 7.19.1 ou ultérieure ; voir également les exigences de l'agent) ; - authentification SMTP (Basic : version 7.20.0 ou ultérieure ; OAuth : version 7.33 ou ultérieure ; voir également les exigences de l'interface) ; - Elasticsearch (version 7.28.0 ou ultérieure) ; - éléments de type requête de télémétrie (version 7.28.0 ou ultérieure). La version 7.28.0 ou ultérieure est recommandée pour toutes les fonctionnalités. Pour utiliser les fonctionnalités cURL mises à niveau avec les éléments `web.page.*`, redémarrez le serveur/proxy Zabbix. Pour l'authentification SMTP, utilisez le paquet `libcurl-full` lors de l'exécution.
libxml2		Requis pour la supervision VMware et le prétraitement XML XPath.
net-snmp		Requis pour la prise en charge de SNMP. Version 5.3.0 ou ultérieure. La prise en charge des protocoles de chiffrement fort (AES192/AES192C, AES256/AES256C) est disponible à partir de la bibliothèque net-snmp 5.8 ; sur les systèmes basés sur RHEL 8 ou ultérieur, il est recommandé d'utiliser net-snmp 5.8.15 ou ultérieur.
libunixodbc		Requis pour la supervision de bases de données.
libgnutls ou *libopenssl*		Requis lors de l'utilisation du chiffrement. Versions minimales : *libgnutls* - 3.1.18, *libopenssl* - 1.0.1
libldap		Requis pour la prise en charge de LDAP.
fping		Requis pour les éléments ping ICMP.
c-ares		Requis pour la résolution DNS asynchrone si Zabbix est configuré avec l'option `--with-ares`. Sinon, *libevent* sera utilisé. Version minimale : 1.16.0. La fonctionnalité de mise en cache des requêtes DNS et de basculement (cache des requêtes activé par défaut) nécessite c-ares 1.26.0 ou ultérieur. Sur les systèmes où c-ares est fourni par le paquet du système d'exploitation, assurez-vous que la bibliothèque d'exécution installée expose les fonctionnalités du résolveur requises pour la mise en cache DNS et le basculement.
g++		Compilateur C++. Requis pour compiler le proxy Zabbix avec l'option `--with-apm`.
gRPC		Requis pour la prise en charge de l'APM sur le proxy Zabbix. Bibliothèques requises : *grpc++*, *grpc*, *gpr*. L'utilitaire `grpc_cpp_plugin` est également requis pour compiler le proxy Zabbix. La version requise est disponible dans les dépôts de paquets standard des systèmes d'exploitation suivants : - Ubuntu 23.04 ou ultérieur ; - Debian 12 (Bookworm) ou ultérieur ; - Alpine Linux 3.15 ou ultérieur ; - RHEL, Rocky Linux, CentOS Stream 9 ou ultérieur (uniquement depuis le dépôt EPEL) ; - Fedora X ou ultérieur. Sur les versions plus anciennes des systèmes d'exploitation, *gRPC* doit être compilé à partir des sources.

Exigence	Statut obligatoire	Description
protobuf		Requis pour la prise en charge de l'APM sur le proxy Zabbix. Le compilateur `protoc` est également requis pour compiler le proxy Zabbix.
Abseil		Requis pour la prise en charge de l'APM sur le proxy Zabbix. Bibliothèques requises : *absl_synchronization*, *absl_base*. Généralement installé avec *gRPC*.

Agent

Requirement	Mandatory status	Description
libpcre2	Yes	La bibliothèque PCRE2 est requise pour la prise en charge des expressions régulières compatibles Perl (PCRE). La version PCRE2 v10.x est prise en charge. Requise pour la surveillance des journaux. Également requise sous Windows.
libpthread		Requise pour la prise en charge des mutex et des verrous en lecture/écriture (peut faire partie de libc). Non requise sous Windows.
libresolv		Requise pour la résolution DNS (peut faire partie de libc). Non requise sous Windows.
libiconv		Requise pour la conversion de l'encodage/du format du texte en UTF-8 dans les éléments de journal, le contenu de fichier, les expressions régulières de fichier et les éléments regmatch (peut faire partie de libc). Non requise sous Windows.
libgnutls or *libopenssl*	No	Requise si le chiffrement est utilisé. Versions minimales: *libgnutls* - 3.1.18, *libopenssl* - 1.0.1 Sous Microsoft Windows, OpenSSL 1.1.1 ou version ultérieure est requis.
libldap		Requise si LDAP est utilisé. Non prise en charge sous Windows.
libcurl		Requise pour la prise en charge étendue des éléments Zabbix agent web.page.*. Sans libcurl, les fonctionnalités de base sont disponibles (par exemple, web.page.get[http://example.com]). Avec libcurl, l'agent prend en charge des fonctionnalités supplémentaires, telles que les URL HTTP avec identifiants (par exemple, http://user:password@example.com) et les URL HTTPS. La version 7.19.1 ou supérieure est requise (7.28.0 ou supérieure est recommandée). Pour utiliser les fonctionnalités cURL mises à niveau, redémarrez l'agent Zabbix.
libmodbus		Requise uniquement si la surveillance Modbus est utilisée. Version 3.0 ou supérieure.

Agent 2

Requirement	Mandatory status	Description
Go	Yes	Requis pour compiler Zabbix agent 2 et ses plugins à partir des sources. Go 1.25 ou version ultérieure est pris en charge. Voir go.dev pour les instructions d'installation. Les bibliothèques Go utilisées par Zabbix agent 2 et ses plugins sont □□□□ées dans le dépôt Git de Zabbix (les bibliothèques marquées comme `indirect` dans le dépôt sont des dépendances d'autres bibliothèques requises) : - Zabbix agent 2 - Plugin support - PostgreSQL - MongoDB - MSSQL - Ember+ - NVIDIA GPU - Example plugin
libpcre2	Yes	La bibliothèque PCRE2 est requise pour la prise en charge des expressions régulières compatibles Perl (PCRE). PCRE2 v10.x est pris en charge. Requis pour la surveillance des journaux. Également requis sous Windows.

Requirement	Mandatory status	Description
libopenssl	No	Requis lors de l'utilisation du chiffrement. OpenSSL 1.0.1 ou version ultérieure est requis sur les plateformes UNIX. La bibliothèque OpenSSL doit avoir la prise en charge PSK activée. LibreSSL n'est pas pris en charge. Sur les systèmes Microsoft Windows, OpenSSL 1.1.1 ou version ultérieure est requis.

Service web

La dernière version stable de Google Chrome est prise en charge pour générer des rapports planifiés à l'aide du service web Zabbix.

La version de Go requise pour compiler le service web correspond à celle utilisée pour Zabbix agent 2.

Java gateway

Si vous avez obtenu Zabbix depuis le dépôt de sources ou une archive, les dépendances nécessaires sont déjà incluses dans l'arborescence des sources.

Si vous avez obtenu Zabbix depuis le paquet de votre distribution, les dépendances nécessaires sont déjà fournies par le système de gestion des paquets.

Dans les deux cas ci-dessus, le logiciel est prêt à être utilisé et aucun téléchargement supplémentaire n'est nécessaire.

Si, toutefois, vous souhaitez fournir vos propres versions de ces dépendances (par exemple, si vous préparez un paquet pour une distribution Linux), vous trouverez ci-dessous la liste des versions de bibliothèques avec lesquelles Java gateway est connu pour fonctionner. Zabbix peut également fonctionner avec d'autres versions de ces bibliothèques.

Le tableau suivant répertorie les fichiers JAR actuellement inclus avec Java gateway dans le code d'origine :

Bibliothèque	Statut obligatoire	Version incluse	Commentaires
android-json	Oui	4.3r1	JSON (JavaScript Object Notation) est un format léger d'échange de données. Il s'agit de l'implémentation Android compatible avec `org.json`, extraite du SDK Android.
logback-classic		1.5.25	
logback-core		1.5.25	
slf4j-api		2.0.16	

Java gateway peut être compilé avec Oracle Java ou avec OpenJDK open source (version 8 ou ultérieure). Les paquets fournis par Zabbix sont compilés avec OpenJDK. Le tableau suivant répertorie les paquets OpenJDK utilisés pour compiler les paquets Zabbix par distribution :

Distribution	Paquet OpenJDK
AlmaLinux 10	`java-21-openjdk-devel` (amd64, arm64: 21.0.12.0.8-1.2)
AlmaLinux 9	`java-17-openjdk-devel` (amd64, arm64: 17.0.15.0.6-2)
AlmaLinux 8	`java-1.8.0-openjdk-devel` (amd64: 1.8.0.332.b09-2; arm64: 1.8.0.382.b05-2)
Amazon Linux 2023	`java-22-amazon-corretto-devel` (amd64, arm64: 22.0.2+9-1)
CentOS Stream 10	`java-21-openjdk-devel` (amd64, arm64: 21.0.11.0.10-2)
CentOS Stream 9	`java-17-openjdk-devel` (amd64, arm64: 17.0.15.0.6-2)
CentOS Stream 8	`java-1.8.0-openjdk-devel` (amd64, arm64: 1.8.0.362.b08-3)
CentOS 7	`java-1.8.0-openjdk-devel` (amd64: 1.8.0.282.b08-1)
Debian 13	`default-jdk-headless` (amd64, arm64: 2:1.21-76)
Debian 12	`default-jdk-headless` (amd64, arm64: 2:1.17-74)
Debian 11	`default-jdk-headless` (amd64: 2:1.11-72)
OpenSUSE Leap 16	`java-21-openjdk-devel` (amd64, arm64: 21.0.11.0-160000.2.1)
OpenSUSE Leap 15	`java-17-openjdk-devel` (amd64: 17.0.5.0-150400.3.9.3; arm64: 17.0.8.0-150400.3.27.1)
Oracle Linux 10	`java-21-openjdk-devel` (amd64, arm64: 21.0.11.0.10-2.0.1)
Oracle Linux 9	`java-17-openjdk-devel` (amd64, arm64: 17.0.15.0.6-2.0.1)
Oracle Linux 8	`java-1.8.0-openjdk-devel` (amd64: 1.8.0.372.b07-4.0.1); `java-11-openjdk-devel` (arm64: 11.0.20.0.8-3.0.1)
Oracle Linux 7	`java-1.8.0-openjdk-devel` (amd64: 1.8.0.282.b08-1)

Distribution	Paquet OpenJDK
Raspberry Pi OS 13	`default-jdk-headless` (arm64, armhf: 2:1.21-76)
Raspberry Pi OS 12	`default-jdk-headless` (arm64, armhf: 2:1.17-74)
Raspberry Pi OS 11	`default-jdk-headless` (arm64: 2:1.11-72; armhf: 2:1.11-72+b4)
RHEL 10	`java-21-openjdk-devel` (amd64, arm64: 21.0.12.0.8-1.2)
RHEL 9	`java-17-openjdk-devel` (amd64, arm64: 17.0.15.0.6-2)
RHEL 8	`java-1.8.0-openjdk-devel` (amd64: 1.8.0.372.b07-4; arm64: 1.8.0.382.b05-2)
RHEL 7	`java-1.8.0-openjdk-devel` (amd64: 1.8.0.282.b08-1)
Rocky Linux 10	`java-21-openjdk-devel` (amd64, arm64: 21.0.12.0.8-1.2)
Rocky Linux 9	`java-17-openjdk-devel` (amd64, arm64: 17.0.15.0.6-2)
Rocky Linux 8	`java-1.8.0-openjdk-devel` (amd64: 1.8.0.372.b07-4; arm64: 1.8.0.382.b05-2)
SLES 16	`java-21-openjdk-devel` (amd64, arm64: 21.0.11.0-160000.1.1)
SLES 15	`java-17-openjdk-devel` (amd64: 17.0.5.0-150400.3.9.3; arm64: 17.0.8.0-150400.3.27.1)
Ubuntu 26.04	`default-jdk-headless` (amd64, arm64: 2:1.25-77)
Ubuntu 24.04	`default-jdk-headless` (amd64, arm64: 2:1.21-75+exp1)
Ubuntu 22.04	`default-jdk-headless` (amd64, arm64: 2:1.11-72build2)
Ubuntu 20.04	`default-jdk-headless` (amd64, arm64: 2:1.11-72)

Adaptateur bridge

L'adaptateur bridge de Zabbix n'a aucune exigence particulière.

Ports par défaut

La liste suivante contient les ports que les composants Zabbix peuvent utiliser dans une configuration par défaut :

Composant Zabbix	Protocole et port	Connexion initiée par
Zabbix agent	TCP 10050	serveur Zabbix ou proxy Zabbix (vérifications passives uniquement)
Zabbix agent 2	TCP 10050	serveur Zabbix ou proxy Zabbix (vérifications passives uniquement)
serveur Zabbix	TCP 10051	agents Zabbix actifs, proxy Zabbix actif, Zabbix sender, interface Zabbix
proxy Zabbix	TCP 10051	agents Zabbix actifs, Zabbix sender, serveur Zabbix (proxy passif uniquement)
Zabbix Java gateway	TCP 10052	serveur Zabbix ou proxy
service web Zabbix	TCP 10053	serveur Zabbix
serveur MCP Zabbix	TCP 8443	clients MCP (agents IA)
interface Zabbix	TCP 80	navigateurs web, service web Zabbix (rapports planifiés), clients API
	TCP 443	
Traps SNMP (snmptrapd)	UDP 162	équipements SNMP
adaptateur de pont Zabbix	TCP 10054	serveur Zabbix

Warning:
Ouvrez uniquement les ports requis par les composants que vous exécutez réellement et limitez chaque règle de pare-feu aux hôtes spécifiques qui doivent se connecter (par exemple, autorisez le port 10050 uniquement depuis le serveur Zabbix ou le proxy, et non depuis n'importe quelle adresse). N'exposez pas directement les ports Zabbix sur Internet.
 Avant d'ouvrir les ports, renforcez la sécurité de l'installation : activez le chiffrement entre les composants, limitez les hôtes autorisés dans les fichiers de configuration des composants et consultez les bonnes pratiques de sécurité.

Taille de la base de données

Les données de configuration de Zabbix nécessitent une quantité fixe d'espace disque et n'augmentent que très peu.

La taille de la base de données Zabbix dépend principalement de ces variables, qui définissent la quantité de données historiques stockées :

- Nombre de valeurs traitées par seconde

Il s'agit du nombre moyen de nouvelles valeurs que le serveur Zabbix reçoit chaque seconde. Par exemple, si nous avons 3000 éléments à surveiller avec un intervalle de rafraîchissement de 60 secondes, le nombre de valeurs par seconde est calculé comme suit : 3000/60 = **50**.

Cela signifie que 50 nouvelles valeurs sont ajoutées à la base de données Zabbix chaque seconde.

- Paramètres du housekeeper pour l'historique

Zabbix conserve les valeurs pendant une période fixe, généralement plusieurs semaines ou plusieurs mois. Chaque nouvelle valeur nécessite une certaine quantité d'espace disque pour les données et l'index.

Ainsi, si nous souhaitons conserver 30 jours d'historique et que nous recevons 50 valeurs par seconde, le nombre total de valeurs sera d'environ (**30***24*3600)* **50** = 129.600.000, soit environ 130M de valeurs.

Selon le moteur de base de données utilisé, le type de valeurs reçues (floats, entiers, chaînes, fichiers journaux, etc.), l'espace disque nécessaire pour conserver une seule valeur peut varier de 40 octets à plusieurs centaines d'octets. En général, il est d'environ 90 octets par valeur pour les éléments numériques[2]. Dans notre cas, cela signifie que 130M de valeurs nécessiteront 130M * 90 octets = **10.9GB** d'espace disque.

Note:
La taille des valeurs des éléments texte/journal est impossible à prévoir exactement, mais vous pouvez compter environ 500 octets par valeur.

- Paramètre du housekeeper pour les tendances

Zabbix conserve un ensemble de valeurs max/min/moyenne/compte sur 1 heure pour chaque élément dans la table **trends**. Ces données sont utilisées pour les tendances et les graphiques sur de longues périodes. La période d'une heure ne peut pas être personnalisée.

La base de données Zabbix, selon le type de base de données, nécessite environ 90 octets par total. Supposons que nous souhaitions conserver les données de tendances pendant 5 ans. Les valeurs pour 3000 éléments nécessiteront 3000*24*365* **90** = **2.2GB** par an, soit **11GB** pour 5 ans.

- Paramètres du housekeeper pour les événements

Chaque événement Zabbix nécessite environ 250 octets d'espace disque[1]. Il est difficile d'estimer le nombre d'événements générés quotidiennement par Zabbix. Dans le pire des cas, nous pouvons supposer que Zabbix génère un événement par seconde.

Pour chaque événement récupéré, un enregistrement event_recovery est créé. En général, la plupart des événements seront récupérés, nous pouvons donc supposer un enregistrement event_recovery par événement. Cela représente 80 octets supplémentaires par événement.

En option, les événements peuvent avoir des tags, chaque enregistrement de tag nécessitant environ 100 octets d'espace disque[1]. Le nombre de tags par événement (#tags) dépend de la configuration. Ainsi, chacun nécessitera #tags * 100 octets d'espace disque supplémentaires.

Cela signifie que si nous voulons conserver 3 ans d'événements, cela nécessitera 3*365*24*3600* (250+80+#tags*100) = **~30GB**+#tags*100B d'espace disque[2].

Note:
[1] Davantage en présence de noms d'événements, de tags et de valeurs non ASCII.
 [2] Les estimations de taille sont basées sur MySQL et peuvent être différentes pour d'autres bases de données.

Le tableau contient des formules qui peuvent être utilisées pour calculer l'espace disque requis pour le système Zabbix :

Parameter	Formula for required disk space (in bytes)
Zabbix configuration	Fixed size. Normally 10MB or less.
History	days*(items/refresh rate)*24*3600*bytes items : number of items days : number of days to keep history refresh rate : average refresh rate of items bytes : number of bytes required to keep single value, depends on database engine, normally ~90 bytes.

Parameter	Formula for required disk space (in bytes)
Trends	days*(items/3600)*24*3600*bytes items : number of items days : number of days to keep history bytes : number of bytes required to keep single trend, depends on the database engine, normally ~90 bytes.
Events	days*events*24*3600*bytes events : number of event per second. One (1) event per second in worst-case scenario. days : number of days to keep history bytes : number of bytes required to keep single event, depends on the database engine, normally ~330 + average number of tags per event * 100 bytes.

Ainsi, l'espace disque total requis peut être calculé comme suit :

Configuration + History + Trends + Events

L'espace disque ne sera PAS utilisé immédiatement après l'installation de Zabbix. La taille de la base de données augmentera, puis cessera d'augmenter à un certain moment, ce qui dépend des paramètres du housekeeper.

Synchronisation de l'heure

Il est très important d'avoir une heure système précise sur le serveur exécutant Zabbix. ntpd est le démon le plus populaire pour synchroniser l'heure de l'hôte avec celle d'autres machines. Il est fortement recommandé de maintenir une heure système synchronisée sur tous les systèmes sur lesquels les composants Zabbix sont exécutés.

Nouveautés et notes de mise à niveau : 8.0

Zabbix 8.0.0 fait suite à Zabbix 7.4.0 et apporte de nouvelles fonctionnalités et améliorations.

Également sur cette page :

- Nouveautés dans les modèles, notamment les nouveaux modèles et les modifications apportées aux modèles existants.
- Notes de mise à niveau, notamment les changements incompatibles.

Voir également : Modifications de l'API.

Notes de mise à niveau

Ces notes concernent la mise à niveau de Zabbix 7.4.x vers Zabbix 8.0.0.

Toutes les notes sont regroupées en :

- **Modifications incompatibles** - modifications susceptibles de rompre les installations existantes et autres informations critiques liées au processus de mise à niveau
- **Autres** - toutes les autres informations décrivant les changements dans les fonctionnalités de Zabbix

Voir aussi :

- Procédure de mise à niveau pour toutes les informations pertinentes sur la mise à niveau depuis les versions antérieures à Zabbix 8.0.0 ;
- Mise à niveau du cluster HA pour des instructions sur la mise à niveau des serveurs dans un cluster à **haute disponibilité** (HA).

Modifications incompatibles

Autres

Nouveautés et notes de mise à niveau : 8.0.x

Cette page présente toutes les nouvelles fonctionnalités incluses dans les versions mineures de la version majeure de Zabbix. Les modifications des modèles et les notes de mise à niveau sont également incluses.

Voir également les nouveautés de la version majeure.

Quoi de neuf dans Zabbix 8.0.1 Cette version n'est pas encore publiée.

Consultez les notes de mise à niveau pour cette version.

Notes de mise à niveau

Appliance Zabbix

Présentation L'appliance Zabbix permet de déployer instantanément le serveur Zabbix et l'interface, au lieu de les configurer manuellement ou de réutiliser un serveur existant pour Zabbix.

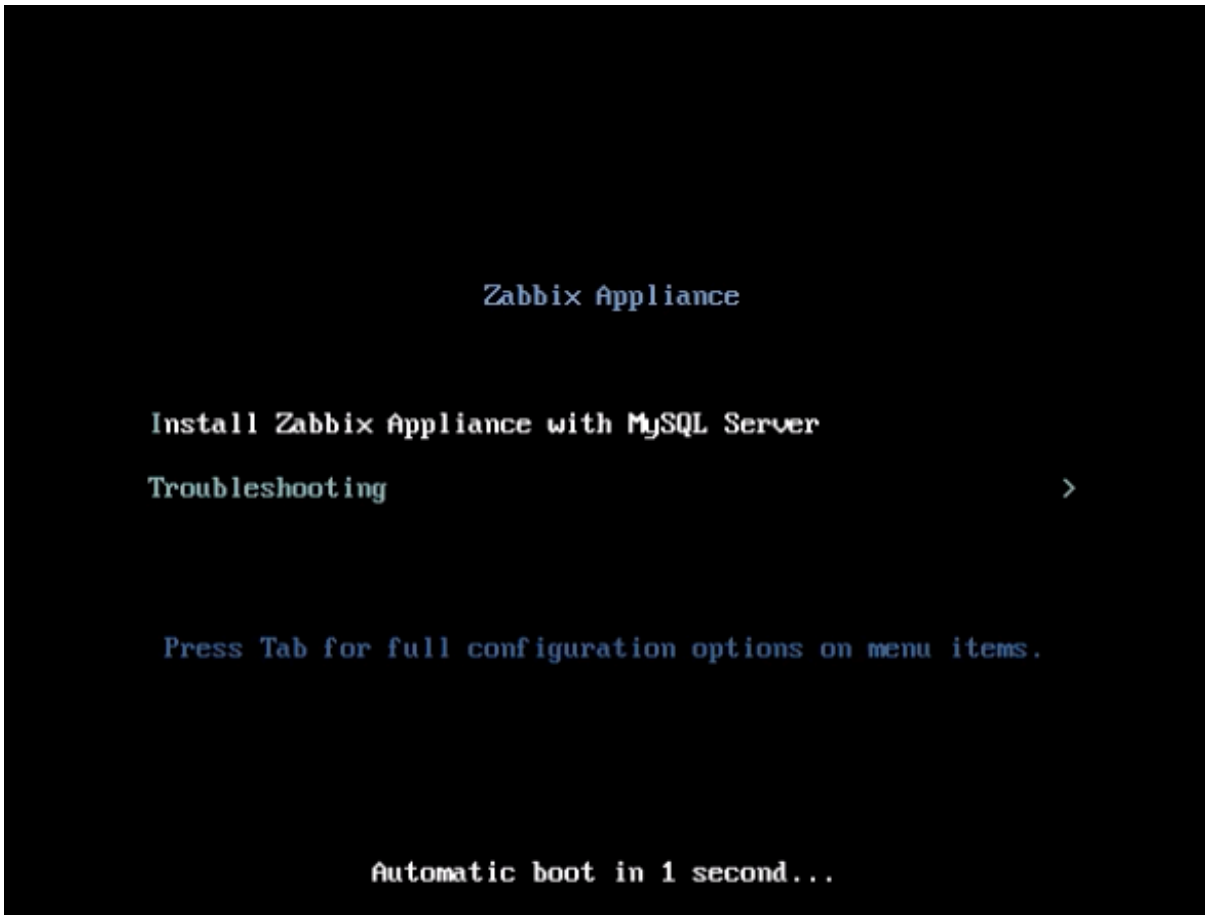
L'appliance pour Zabbix 8.0 est basée sur des dérivés de Red Hat Enterprise Linux 10, tels qu'AlmaLinux ou Rocky Linux, et contient un serveur Zabbix préconfiguré fonctionnant avec MySQL ainsi que l'interface exécutée sur un serveur web NGINX.

> **Attention:**
> Cette appliance est conçue pour évaluer Zabbix. Son utilisation dans des environnements de production sérieux est déconseillée.

Les images de l'appliance sont disponibles au téléchargement dans les formats suivants :

- CD/DVD d'installation (.iso)
- VMware (.vmx) - voir les remarques
- Format de virtualisation ouvert (.ovf)
- Microsoft Hyper-V (.vhd/.vhdx) - voir les remarques
- KVM, Parallels, QEMU, clé USB, VirtualBox, Xen (.raw) - voir les remarques
- KVM, QEMU (.qcow2)

Menu de démarrage du CD/DVD d'installation de Zabbix :



Démarrage rapide Prérequis

1. Assurez-vous que la machine hôte dispose de ressources suffisantes pour répondre aux exigences système de la machine virtuelle :

- *RAM* : 4 Go
- *Espace disque* : au moins 8 Go doivent être alloués à la machine virtuelle
- *CPU* : 2 cœurs minimum

2. Si ce n'est pas encore fait, installez le logiciel de virtualisation permettant de démarrer l'image de l'appliance (par exemple, VirtualBox).
3. Téléchargez l'appliance dans le format pris en charge par votre logiciel de virtualisation.
4. Vérifiez les paramètres réseau afin de vous assurer que DHCP est activé sur la machine hôte.

Installation

1. Démarrez la machine virtuelle de l'appliance à partir de l'image téléchargée. Pour obtenir des instructions, veuillez consulter la documentation de votre logiciel de virtualisation, par exemple, la documentation VirtualBox.
2. Configurez les paramètres réseau de la machine virtuelle afin d'autoriser l'accès depuis un navigateur sur la machine hôte. Cela peut être réalisé en activant le *mode pont*.
3. Connectez-vous à la machine virtuelle à l'aide des identifiants système par défaut.
4. Pour récupérer l'adresse IP, exécutez la commande suivante sur la machine virtuelle :

```
ip addr show
```

5. Ouvrez un navigateur sur la machine hôte et pointez-le vers l'adresse IP que l'appliance a reçue via DHCP.
6. Connectez-vous à Zabbix à l'aide des identifiants par défaut et commencez la supervision.

Configuration Cette section décrit les paramètres de configuration par défaut fréquemment requis ainsi que les options de personnalisation disponibles.

Identifiants

Système

- Nom d'utilisateur : `root`
- Mot de passe : `zabbix`

Interface Zabbix

- Nom d'utilisateur : `Admin`
- Mot de passe : `zabbix`

Après vous être connecté, vous pouvez modifier le mot de passe par défaut dans les paramètres du profil utilisateur ou créer de nouveaux utilisateurs et supprimer l'utilisateur par défaut.

Base de données

Les mots de passe de tous les utilisateurs de la base de données sont générés aléatoirement pendant le processus d'installation. Les utilisateurs suivants sont définis pour la base de données :

Root :

- Nom d'utilisateur : `root`
- Mot de passe : le mot de passe est stocké dans le fichier `/root/.my.cnf` Il n'est pas nécessaire de saisir un mot de passe avec le compte root.

Serveur Zabbix :

- Nom d'utilisateur : `zabbix_srv`
- Mot de passe : le mot de passe est stocké dans le fichier de configuration du serveur

Interface Zabbix :

- Nom d'utilisateur : `zabbix_web`
- Mot de passe : le mot de passe est stocké dans le fichier de configuration de l'interface

Pour modifier le mot de passe d'un utilisateur de la base de données, modifiez-le dans MySQL ainsi que dans le fichier de configuration correspondant.

Accès à l'interface

L'interface Zabbix est accessible à l'adresse *http://<virtual machine's IP>*.

Par défaut, l'accès est autorisé depuis n'importe où. Pour limiter l'accès, modifiez `/etc/nginx/conf.d/zabbix.conf`. Après avoir enregistré le fichier modifié, redémarrez NGINX en vous connectant via SSH en tant qu'utilisateur root et en exécutant :

```
systemctl restart nginx
```

Adresse IP statique

Par défaut, l'appliance utilise DHCP pour obtenir l'adresse IP. Pour définir une adresse IP statique :

1. Connectez-vous en tant que utilisateur root.
2. Exécutez les commandes suivantes, en remplaçant les valeurs par votre adresse IP personnalisée :

```
nmcli con mod eth0 ipv4.addresses 192.168.1.10/24    # Adresse IP/CIDR de l'appliance
nmcli con mod eth0 ipv4.gateway 192.168.1.1     # Adresse IP de la passerelle
nmcli con mod eth0 ipv4.dns 8.8.8.8    # Adresse IP du serveur DNS
nmcli con mod eth0 ipv4.method manual
nmcli con up eth0
```

Vous pouvez également ouvrir le fichier `/etc/NetworkManager/system-connections/eth0.nmconnection` et ajouter les lignes suivantes :

```
[ipv4]
address1=192.168.1.10/24
dns=8.8.8.8;
gateway=192.168.1.1
method=manual
```

Après avoir enregistré le fichier modifié, exécutez les commandes suivantes :

```
nmcli con reload
nmcli con up eth0
```

Pour vérifier que les modifications de configuration réseau ont été appliquées, exécutez la commande suivante :

```
nmcli dev show eth0
```

Configuration du pare-feu

Pour gérer les paramètres du pare-feu, l'appliance utilise iptables avec des règles prédéfinies :

- Ouvrir le port SSH (22 TCP)
- Ouvrir les ports de l'agent Zabbix (10050 TCP) et du Zabbix trapper (10051 TCP)
- Ouvrir les ports HTTP (80 TCP) et HTTPS (443 TCP)
- Ouvrir le port de trap SNMP (162 UDP)
- Ouvrir les connexions sortantes vers le port NTP (123 UDP)
- Limiter les paquets ICMP à 5 paquets par seconde
- Rejeter toutes les autres connexions entrantes

Pour ouvrir des ports supplémentaires, modifiez le fichier `/etc/sysconfig/iptables` et rechargez les règles du pare-feu :

```
systemctl reload iptables
```

Dépôts

L'appliance Zabbix utilise le paquet *zabbix-release* du dépôt Zabbix. Les dépôts sont configurés dans le répertoire `/etc/yum.repos.d/*`.

Fuseau horaire

Par défaut, l'appliance utilise UTC pour l'horloge système. Pour modifier le fuseau horaire, copiez le fichier approprié depuis `/usr/share/zoneinfo` vers `/etc/localtime`, par exemple :

```
cp /usr/share/zoneinfo/Europe/Riga /etc/localtime
```

> **Note:**
> Le fuseau horaire du frontend de Zabbix est défini séparément et peut être modifié dans les paramètres du frontend. Le fuseau horaire par défaut du frontend Zabbix est Europe/Riga.

Emplacements des fichiers

- Les fichiers de configuration du serveur Zabbix et de l'agent se trouvent dans `/etc/zabbix`.
- Les fichiers journaux du serveur Zabbix et de l'agent se trouvent dans `/var/log/zabbix`.
- Le fichier de configuration de l'interface Zabbix se trouve dans `/etc/zabbix/web`.
- L'interface Zabbix se trouve dans `/usr/share/zabbix`.
- Le répertoire personnel de l'utilisateur *zabbix* est `/var/lib/zabbix`.

Services système

Les services systemd sont disponibles. Pour voir la liste des services Zabbix, exécutez la commande suivante sur la machine virtuelle :

```
systemctl list-units zabbix*
```

Remarques spécifiques au format d'image VMware

Les images au format *vmdk* sont directement utilisables dans les produits VMware Player, Server et Workstation. Pour les utiliser avec ESX, ESXi et vSphere, elles doivent être converties à l'aide de VMware vCenter Converter (authentification requise pour le téléchargement). Si vous utilisez VMware vCenter Converter, vous pouvez rencontrer des problèmes avec l'adaptateur réseau hybride. Dans ce cas, vous pouvez essayer de spécifier l'adaptateur E1000 pendant le processus de conversion. Vous pouvez également, une fois la conversion terminée, supprimer l'adaptateur existant et ajouter un adaptateur E1000.

Image HDD/flash (raw)

Pour démarrer l'image, exécutez :

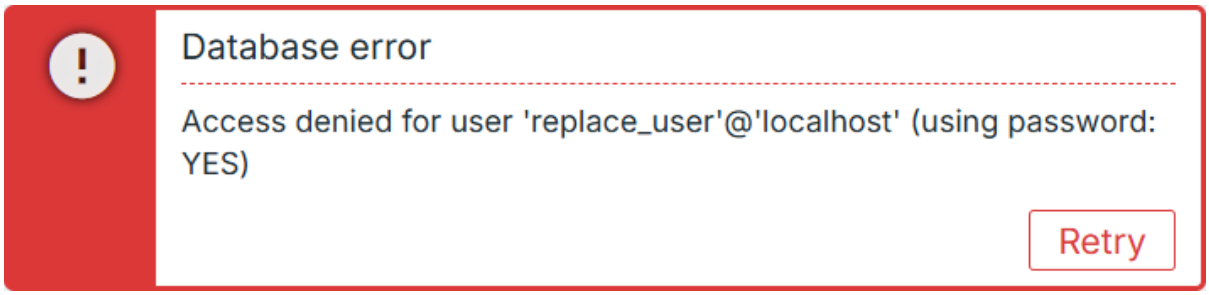
```
dd if=./zabbix_appliance_8.0.0.raw of=/dev/sdc bs=4k conv=fdatasync
```

Remplacez `/dev/sdc` par le chemin du périphérique de votre disque Flash/HDD.

Hyper-V

Si l'appliance ne parvient pas à démarrer dans Hyper-V, essayez d'appuyer sur **Ctrl+Alt+F2** pour basculer vers une session TTY.

Dépannage Si vous rencontrez le message d'erreur *Access denied for user 'replace_user'@'localhost' (using password: YES)* lors de la tentative de connexion à l'interface, cela peut indiquer que l'installation est toujours en cours.



Si l'erreur persiste après quelques minutes d'attente, ou si vous observez un autre comportement inattendu, cela signifie probablement que le processus d'installation ne s'est pas terminé correctement. Dans ce cas, nous vous recommandons de supprimer l'appliance actuelle et de la redéployer en suivant les mêmes instructions d'installation. Cette étape résout généralement le problème.

Notez qu'il n'est pas recommandé de tenter de corriger manuellement une installation défectueuse, car cela peut entraîner d'autres complications.

Processus

Vue d'ensemble Zabbix se compose de plusieurs processus qui travaillent ensemble pour surveiller les systèmes et envoyer des notifications. Chaque processus a un rôle spécifique dans l'infrastructure de surveillance.

Processus principaux

- Serveur - le processus central qui effectue la collecte et la réception des données, calcule les déclencheurs et envoie des notifications aux utilisateurs.
- Agent - collecte les données des hôtes surveillés.
- Agent 2 - un agent de nouvelle génération doté d'une architecture à plugins et de fonctionnalités améliorées.
- Proxy - collecte les données de surveillance des appareils surveillés pour le compte du serveur Zabbix et envoie les informations à un serveur Zabbix centralisé dans une configuration distribuée.
- Java gateway - permet la surveillance des applications Java.
- Serveur MCP - permet aux agents d'IA de travailler avec Zabbix à l'aide du protocole Model Context Protocol.

Pour accéder facilement à Zabbix depuis n'importe où et depuis n'importe quelle plateforme, une interface web est fournie. L'interface fait partie du serveur Zabbix et s'exécute généralement (mais pas nécessairement) sur la même machine physique que celle qui exécute le serveur.

Toutes les informations de configuration ainsi que les données collectées par Zabbix sont stockées dans une base de données.

En-tête L'en-tête est présent dans tous les messages de requête et de réponse entre les composants Zabbix. Il est nécessaire pour déterminer la longueur du message, s'il est compressé ou non, et s'il s'agit ou non d'un paquet volumineux.

Le protocole de communication Zabbix impose une limite de taille de paquet de 1 Go par connexion. La limite de 1 Go s'applique à la fois à la longueur des données du paquet reçu et à la longueur des données non compressées.

Lors de l'envoi de la configuration à un proxy Zabbix, la limite de taille de paquet est portée à 4 Go afin de permettre la synchronisation de configurations volumineuses. Lorsque la longueur des données avant compression dépasse 4 Go, le serveur Zabbix commence automatiquement à utiliser le format de paquet volumineux (indicateur 0x04), ce qui porte la limite de taille de paquet à 16 Go.

Notez que, bien qu'un format de paquet volumineux puisse être utilisé pour l'envoi de n'importe quelles données, à l'heure actuelle seul le synchroniseur de configuration du proxy Zabbix peut traiter des paquets supérieurs à 1 Go.

Structure de l'en-tête L'en-tête se compose de quatre champs. Tous les nombres dans l'en-tête sont formatés en little-endian.

Field	Size	Size (large packet)	Description
<PROTOCOL>	4	4	"ZBXD" ou 5A 42 58 44
<FLAGS>	1	1	Indicateurs du protocole : 0x01 - protocole de communication Zabbix 0x02 - compression 0x04 - paquet volumineux
<DATALEN>	4	8	Longueur des données.
<RESERVED>	4	8	Lorsque la compression est utilisée (indicateur 0x02) - longueur des données non compressées Lorsque la compression n'est pas utilisée - 00 00 00 00

Exemples d'en-tête Voici quelques extraits de code montrant comment ajouter l'en-tête du protocole Zabbix aux données que vous souhaitez envoyer afin d'obtenir le paquet à envoyer à Zabbix pour qu'il soit interprété correctement. Ces extraits de code supposent que les données ne dépassent pas 1 Go, donc le format de grand paquet n'est pas utilisé.

Python

```
packet = b"ZBXD\1" + struct.pack("<II", len(data), 0) + data
```

ou

```
def zbx_create_header(plain_data_size, compressed_data_size=None):
    protocol = b"ZBXD"
    flags = 0x01
    if compressed_data_size is None:
        datalen = plain_data_size
        reserved = 0
    else:
        flags |= 0x02
        datalen = compressed_data_size
        reserved = plain_data_size
    return protocol + struct.pack("<BII", flags, datalen, reserved)

packet = zbx_create_header(len(data)) + data
```

Perl

```
my $packet = "ZBXD\1" . pack("(II)<", length($data), 0) . $data;
```

ou

```
sub zbx_create_header($;$)
{
    my $plain_data_size = shift;
    my $compressed_data_size = shift;

    my $protocol = "ZBXD";
    my $flags = 0x01;
    my $datalen;
    my $reserved;

    if (!defined($compressed_data_size))
    {
        $datalen = $plain_data_size;
        $reserved = 0;
    }
    else
    {
        $flags |= 0x02;
        $datalen = $compressed_data_size;
        $reserved = $plain_data_size;
    }

    return $protocol . chr($flags) . pack("(II)<", $datalen, $reserved);
}

my $packet = zbx_create_header(length($data)) . $data;
```

PHP

```
$packet = "ZBXD\1" . pack("VV", strlen($data), 0) . $data;
```

ou

```
function zbx_create_header($plain_data_size, $compressed_data_size = null)
{
    $protocol = "ZBXD";
    $flags = 0x01;
    if (is_null($compressed_data_size))
    {
        $datalen = $plain_data_size;
        $reserved = 0;
    }
    else
    {
        $flags |= 0x02;
        $datalen = $compressed_data_size;
        $reserved = $plain_data_size;
    }
    return $protocol . chr($flags) . pack("VV", $datalen, $reserved);
}

$packet = zbx_create_header(strlen($data)) . $data;
```

Bash

```
datalen=$(printf "%08x" ${#data})
datalen="\\x${datalen:6:2}\\x${datalen:4:2}\\x${datalen:2:2}\\x${datalen:0:2}"
printf "ZBXD\1${datalen}\0\0\0\0%s" "$data"
```

Serveur

Vue d'ensemble Le serveur Zabbix est le processus central du logiciel Zabbix.

Le serveur effectue le sondage et la réception des données, il calcule les déclencheurs, envoie des notifications aux utilisateurs. C'est le composant central auquel les agents Zabbix et les proxies transmettent des données sur la disponibilité et l'intégrité des systèmes. Le serveur peut également vérifier à distance des services en réseau (tels que des serveurs web et des serveurs de messagerie) à l'aide de simples contrôles de service.

Le serveur est le référentiel central dans lequel sont stockées toutes les données de configuration, statistiques et opérationnelles, et c'est l'entité de Zabbix qui alertera activement les administrateurs lorsque des problèmes surviennent sur l'un des systèmes surveillés.

Le fonctionnement d'un serveur Zabbix de base se décompose en trois composants distincts ; il s'agit de : serveur Zabbix, interface web et stockage de la base de données.

Toutes les informations de configuration de Zabbix sont stockées dans la base de données, avec laquelle interagissent à la fois le serveur et l'interface web. Par exemple, lorsque vous créez un nouvel élément à l'aide de l'interface web (ou de l'API), il est ajouté à la table items dans la base de données. Ensuite, environ une fois par minute, le serveur Zabbix interrogera la table items pour obtenir une liste des éléments actifs, qui est ensuite stockée dans un cache au sein du serveur Zabbix. C'est pourquoi il peut s'écouler jusqu'à deux minutes avant que toute modification effectuée dans l'interface Zabbix n'apparaisse dans la section des dernières données.

Voir aussi :

- Configuration de la haute disponibilité pour les serveurs Zabbix
- Paramètres de configuration: serveur Zabbix

Exécution du serveur Si installé en tant que paquet

Le serveur Zabbix s'exécute en tant que processus démon. Le serveur peut être démarré en exécutant :

```
systemctl start zabbix-server
```

Cela fonctionnera sur la plupart des systèmes GNU/Linux. Sur d'autres systèmes, vous devrez peut-être exécuter :

```
/etc/init.d/zabbix-server start
```

De même, pour arrêter/redémarrer/afficher l'état, utilisez les commandes suivantes :

```
systemctl stop zabbix-server
systemctl restart zabbix-server
systemctl status zabbix-server
```

Démarrer manuellement

Si ce qui précède ne fonctionne pas, vous devez le démarrer manuellement. Trouvez le chemin vers le binaire `zabbix_server` et exécutez :

```
zabbix_server
```

Vous pouvez utiliser les paramètres de ligne de commande suivants avec le serveur Zabbix :

Parameter	Description
`-c --config <config-file>`	Chemin vers le fichier de configuration.
`-f --foreground`	Exécuter le serveur Zabbix au premier plan.
`-R --runtime-control <option>`	Effectuer des fonctions administratives.
`-T --test-config`	Valider le fichier de configuration et quitter.
`-h --help`	Afficher les informations d'aide et quitter.
`-V --version`	Afficher le numéro de version.

Exemples d'exécution du serveur Zabbix avec des paramètres de ligne de commande :

```
zabbix_server -c /usr/local/etc/zabbix_server.conf
zabbix_server --help
zabbix_server -V
```

Contrôle à l'exécution

Options de contrôle à l'exécution :

Option	Description	Cible
`config_cache_reload`	Recharger le cache de configuration. Ignoré si le cache est actuellement en cours de chargement.	
`history_cache_clear[=<target>]`	Effacer le cache de l'historique pour l'élément spécifié par son ID. Affecte toutes les valeurs de l'élément, à l'exception de la première et de la dernière valeur.	**target** - ID de l'élément
`diaginfo[=<section>]`	Collecter les informations de diagnostic dans le fichier journal du serveur.	`historycache` - statistiques du cache de l'historique ; `valuecache` - statistiques du cache des valeurs ; `preprocessing` - statistiques du gestionnaire du prétraitement ; `alerting` - statistiques du gestionnaire des alertes ; `lld` - statistiques du gestionnaire LLD ; `locks` - liste des mutex (vide sur les systèmes *BSD) ; `connector` - statistiques des connecteurs ayant la file d'attente la plus importante.
`ha_status`	Consigner l'état du cluster de haute disponibilité (HA).	
`ha_remove_node=target`	Supprimer le nœud de haute disponibilité (HA) spécifié par son nom ou son ID. Notez que les nœuds actifs/en attente ne peuvent pas être supprimés.	**target** - nom ou ID du nœud (peut être obtenu en exécutant `ha_status`).
`ha_set_failover_delay=delay`	Définir le délai de basculement de la haute disponibilité (HA). Les suffixes de temps sont pris en charge, par exemple 10s, 1m.	
`proxy_config_cache_reload[=<target>]`	Recharger le cache de configuration du proxy.	**target** - liste de noms de proxy séparés par des virgules. Si aucune cible n'est spécifiée, recharger la configuration de tous les proxy.
`secrets_reload`	Recharger les secrets depuis Vault.	
`service_cache_reload`	Recharger le cache du gestionnaire des services.	
`snmp_cache_reload`	Recharger le cache SNMP - effacer les propriétés du moteur SNMP (temps du moteur, démarrages du moteur, ID du moteur, identifiants) pour tous les hôtes. À utiliser pour forcer un effacement global du cache lors du dépannage de problèmes SNMP.	
`housekeeper_execute`	Démarrer la procédure de maintenance. Ignoré si la procédure de maintenance est actuellement en cours.	
`trigger_housekeeper_execute`	Démarrer la procédure de maintenance des déclencheurs. Ignoré si la procédure de maintenance des déclencheurs est actuellement en cours.	
`log_level_increase[=<target>]`	Augmenter le niveau de journalisation ; affecte tous les processus si aucune cible n'est spécifiée. Non pris en charge sur les systèmes *BSD*.	**process type** - tous les processus du type spécifié (par exemple, `poller`). Voir tous les types de processus du serveur. **process type,N** - type et numéro du processus (par exemple, `poller,3`). **pid** - identifiant du processus (1 à 65535). Pour les valeurs supérieures, spécifiez la cible sous la forme « process type,N ».
`log_level_decrease[=<target>]`	Diminuer le niveau de journalisation ; affecte tous les processus si aucune cible n'est spécifiée. Non pris en charge sur les systèmes *BSD*.	

Option	Description	Cible
`prof_enable[=<target>]`	Activer le profilage. Affecte tous les processus si aucune cible n'est spécifiée. Le profilage activé fournit des informations détaillées sur tous les rwlocks/mutex par nom de fonction.	**process type** - tous les processus du type spécifié (par exemple, `history syncer`) Types de processus pris en charge comme cibles du profilage : `alerter`, `alert manager`, `availability manager`, `configuration syncer`, `discovery manager`, `escalator`, `history poller`, `history syncer`, `housekeeper`, `http poller`, `icmp pinger`, `ipmi manager`, `ipmi poller`, `java poller`, `lld manager`, `lld worker`, `odbc poller`, `poller`, `preprocessing manager`, `proxy poller`, `self-monitoring`, `service manager`, `snmp trapper`, `task manager`, `timer`, `trapper`, `unreachable poller`, `vmware collector`. **process type,N** - type et numéro du processus (par exemple, `history syncer,1`). **pid** - identifiant du processus (1 à 65535). Pour les valeurs supérieures, spécifiez la cible sous la forme « process type,N ». **scope** - `rwlock`, `mutex`, `processing` peuvent être utilisés avec le type et le numéro du processus (par exemple, `history syncer,1,processing`) ou avec tous les processus du type (par exemple, `history syncer,rwlock`).
`prof_disable[=<target>]`	Désactiver le profilage. Affecte tous les processus si aucune cible n'est spécifiée.	**process type** - tous les processus du type spécifié (par exemple, `history syncer`). Types de processus pris en charge comme cibles du profilage : voir `prof_enable`. **process type,N** - type et numéro du processus (par exemple, `history syncer,1`). **pid** - identifiant du processus (1 à 65535). Pour les valeurs supérieures, spécifiez la cible sous la forme « process type,N ».

Lorsqu'un type de processus est spécifié comme cible pour le contrôle à l'exécution, la commande s'applique uniquement aux processus en cours d'exécution de ce type. Si aucun processus du type spécifié n'est actuellement en cours d'exécution, la commande renvoie une erreur :

```
zabbix_server -R log_level_increase='discovery manager'
Cannot redirect signal: "discovery manager" process does not exist
```

Exemple d'utilisation du contrôle à l'exécution pour recharger le cache de configuration du serveur :

```
zabbix_server -c /usr/local/etc/zabbix_server.conf -R config_cache_reload
```

Exemples d'utilisation du contrôle à l'exécution pour recharger la configuration du proxy :

```
### Recharger la configuration de tous les proxy :
zabbix_server -R proxy_config_cache_reload

### Recharger la configuration de Proxy1 et Proxy2 :
zabbix_server -R proxy_config_cache_reload=Proxy1,Proxy2
```

Exemple d'utilisation du contrôle à l'exécution pour vider le cache d'historique d'un élément :

```
zabbix_server -c /usr/local/etc/zabbix_server.conf -R history_cache_clear=42243
```

Exemples d'utilisation du contrôle à l'exécution pour collecter des informations de diagnostic :

```
### Collecter toutes les informations de diagnostic disponibles dans le fichier journal du serveur :
zabbix_server -R diaginfo
```

```
### Collecter les statistiques du cache d'historique dans le fichier journal du serveur :
zabbix_server -R diaginfo=historycache
```

Exemple d'utilisation du contrôle à l'exécution pour recharger le cache SNMP :

```
zabbix_server -R snmp_cache_reload
```

Attention:
Lorsqu'une interface SNMPv3 est mise à jour via l'interface Zabbix, Zabbix recharge automatiquement les nouveaux identifiants SNMPv3 pour cette interface dans la plupart des cas ; utilisez `-R snmp_cache_reload` uniquement si la collecte échoue encore après une modification des identifiants (par exemple, en raison d'incohérences engineBoots/engineID ou de périphériques non conformes RFC), ou lorsque vous devez forcer un vidage global du cache SNMP à des fins de dépannage.

Exemple d'utilisation du contrôle à l'exécution pour déclencher l'exécution du housekeeper :

```
zabbix_server -c /usr/local/etc/zabbix_server.conf -R housekeeper_execute
```

Exemples d'utilisation du contrôle à l'exécution pour modifier le niveau de journalisation :

```
### Augmenter le niveau de journalisation de tous les processus :
zabbix_server -c /usr/local/etc/zabbix_server.conf -R log_level_increase

### Augmenter le niveau de journalisation du deuxième processus poller :
zabbix_server -c /usr/local/etc/zabbix_server.conf -R log_level_increase=poller,2

### Augmenter le niveau de journalisation du processus avec le PID 1234 :
zabbix_server -c /usr/local/etc/zabbix_server.conf -R log_level_increase=1234

### Diminuer le niveau de journalisation de tous les processus http poller :
zabbix_server -c /usr/local/etc/zabbix_server.conf -R log_level_decrease="http poller"
```

Exemple de définition du délai de basculement HA au minimum de 10 secondes :

```
zabbix_server -R ha_set_failover_delay=10s
```

Utilisateur du processus

Le serveur Zabbix est conçu pour s'exécuter sous un utilisateur non-root. Il s'exécutera sous n'importe quel utilisateur non-root avec lequel il est démarré. Vous pouvez donc exécuter le serveur sous n'importe quel utilisateur non-root sans aucun problème.

Si vous essayez de l'exécuter en tant que 'root', il basculera vers un utilisateur 'zabbix' codé en dur, qui doit être présent sur votre système. Vous ne pouvez exécuter le serveur en tant que 'root' que si vous modifiez en conséquence le paramètre 'AllowRoot' dans le fichier de configuration du serveur.

Si le serveur Zabbix et l'agent s'exécutent sur la même machine, il est recommandé d'utiliser un utilisateur différent pour l'exécution du serveur et de l'agent. Sinon, si les deux s'exécutent sous le même utilisateur, l'agent peut accéder au fichier de configuration du serveur et tout utilisateur de niveau Admin dans Zabbix peut assez facilement récupérer, par exemple, le mot de passe de la base de données.

Fichier de configuration

Le chemin par défaut du fichier de configuration du serveur dépend du système d'exploitation et de la méthode d'installation de Zabbix :

- Pour une installation à partir des sources : `/usr/local/etc/zabbix_server.conf`, ou selon les variables définies lors de la compilation `--sysconfdir` ou `--prefix`.
- Pour une installation à partir de paquets ou d'une appliance : `/etc/zabbix/zabbix_server.conf`.

Consultez les options du fichier de configuration pour plus d'informations sur la configuration de `zabbix_server`.

Scripts de démarrage

Les scripts sont utilisés pour démarrer/arrêter automatiquement les processus Zabbix lors du démarrage/de l'arrêt du système. Les scripts se trouvent dans le répertoire misc/init.d.

Types de processus et threads du serveur

- `agent poller` - processus de collecte asynchrone pour les vérifications passives avec un thread de travail ;
- `alert manager` - gestionnaire de la file d'attente des alertes ;
- `alert syncer` - processus d'écriture des alertes dans la base de données ;

- alerter - processus d'envoi des notifications ;
- availability manager - processus de mise à jour de la disponibilité des hôtes ;
- browser poller - processus de collecte pour les vérifications des éléments de navigateur ;
- configuration syncer - processus de gestion du cache en mémoire des données de configuration ;
- configuration syncer worker - processus de résolution et de synchronisation des valeurs des macros utilisateur dans les noms des éléments ;
- connector manager - processus de gestion des connecteurs ;
- connector worker - processus de traitement des requêtes du gestionnaire de connecteurs ;
- discovery manager - processus de gestion de la découverte des périphériques ;
- discovery worker - processus de traitement des tâches de découverte du gestionnaire de découverte ;
- escalator - processus d'escalade des actions ;
- ha manager - processus de gestion de la haute disponibilité ;
- history poller - processus de traitement des vérifications calculées nécessitant une connexion à la base de données ;
- history syncer - processus d'écriture de l'historique dans la base de données ;
- housekeeper - processus de suppression des données obsolètes (historique et tendances des éléments, sessions utilisateur, événements, etc.), ainsi que des données laissées par les objets supprimés ;
- http agent poller - processus de collecte asynchrone pour les vérifications HTTP avec un thread de travail ;
- http poller - processus de collecte de la supervision Web ;
- icmp pinger - processus de collecte pour les vérifications icmpping ;
- internal poller - processus de collecte pour les vérifications internes ;
- ipmi manager - gestionnaire du processus de collecte IPMI ;
- ipmi poller - processus de collecte pour les vérifications IPMI ;
- java poller - processus de collecte pour les vérifications Java ;
- lld manager - processus de gestion des tâches de découverte de bas niveau ;
- lld worker - processus de travail des tâches de découverte de bas niveau ;
- odbc poller - processus de collecte pour les vérifications ODBC ;
- poller - processus de collecte normal pour les vérifications passives ;
- preprocessing manager - gestionnaire des tâches de prétraitement avec des threads de travail de prétraitement ;
- preprocessing worker - thread de prétraitement des données ;
- proxy poller - processus de collecte pour les proxy passifs ;
- proxy group manager - gestionnaire de l'équilibrage de charge et de la haute disponibilité des proxy ;
- report manager- gestionnaire des tâches planifiées de génération de rapports ;
- report writer - processus de génération des rapports planifiés ;
- self-monitoring - processus de collecte des statistiques internes du serveur ;
- service manager - processus de gestion des services en recevant les informations sur les problèmes, les balises de problème et la résolution des problèmes depuis le processus de synchronisation de l'historique, le gestionnaire de tâches et le gestionnaire des alertes ;
- snmp poller - processus de collecte asynchrone pour les vérifications SNMP avec un thread de travail (uniquement les éléments walk[OID] et get[OID]) ;
- snmp trapper - processus de réception des traps SNMP ;
- supervisor - coordonne le démarrage et l'arrêt des processus et threads du serveur, et surveille leur état de préparation ;
- task manager - processus d'exécution à distance des tâches demandées par d'autres composants (par exemple, fermer un problème, acquitter un problème, vérifier immédiatement la valeur d'un élément, fonctionnalité de commande à distance) ;
- telemetry query poller - processus de collecte pour les vérifications des éléments de requête de télémétrie ;
- timer - processus de traitement des périodes de maintenance ;
- trapper - processus de réception pour les vérifications actives, les traps et la communication avec les proxy ;
- trigger housekeeper - processus de suppression des problèmes et des événements générés par des déclencheurs qui ont depuis été supprimés ;
- unreachable poller - processus de collecte pour les périphériques inaccessibles ;
- vmware collector - collecteur de données VMware chargé de recueillir les données des services VMware.

Le fichier journal du serveur peut être utilisé pour observer ces types de processus.

Le fichier journal du serveur est créé avec des autorisations de lecture-écriture uniquement pour le propriétaire du fichier. De plus, le fichier est lisible par le groupe propriétaire. Toutes les autres autorisations sont refusées.

Différents types de processus du serveur Zabbix peuvent être surveillés à l'aide de l'élément interne zabbix[process,<type>,<mode>,<st

Statistiques des transactions du synchroniseur d'historique

Le titre du processus du synchroniseur d'historique affiche des statistiques détaillées sur les transactions du synchroniseur d'historique :

```
205182 ?        S      0:00  zabbix_server: history syncer #2 [processed 0 values, 0+0 triggers in 0.00002
205183 ?        S      0:00  zabbix_server: history syncer #3 [processed 18 values, 7+0 triggers in 0.0026
205184 ?        S      0:00  zabbix_server: history syncer #4 [processed 0 values, 0+0 triggers in 0.00002
```

Dans "A+B triggers" :

- A - déclencheurs traités en raison des valeurs d'historique ;
- B - déclencheurs traités en raison des minuteries.

Les durées, dans `processed...in N (<timings>) sec`, sont :

- Temps passé à écrire les valeurs des éléments dans la base de données ;
- Temps passé à mettre à jour les données des éléments (état, erreurs, inventaire de l'hôte, etc.) ;
- Temps passé à vider les tendances vers la base de données ;
- Temps passé à calculer les déclencheurs ;
- Temps passé à traiter les événements et les actions.

Procédure de housekeeping

Le processus `housekeeper` supprime périodiquement les données obsolètes (historique et tendances des éléments, sessions utilisateur, événements, etc.), ainsi que les données laissées par des objets supprimés. Il s'exécute par cycles, avec une fréquence et une limite de suppression par cycle déterminées par `HousekeepingFrequency` et `MaxHousekeeperDelete`. Toute donnée non supprimée lors d'un cycle est reportée au cycle suivant. Le housekeeping automatique peut être activé et configuré dans *Administration* > *Housekeeping*.

Lorsque le processus `housekeeper` supprime les données laissées par des objets supprimés, il s'appuie sur des tâches de la table `housekeeper`, qui est alimentée chaque fois qu'un objet est supprimé. Par exemple, lorsque vous supprimez un hôte, Zabbix supprime ses éléments, mais pas leur historique, leurs tendances ni leurs problèmes. À la place, des déclencheurs de base de données alimentent la table `housekeeper` avec des tâches composées des champs suivants :

- `housekeeperid` - ID de la tâche
- `object` - type d'objet (0 - élément; 1 - déclencheur; 2 - service; 3 - hôte découvert; 4 - service découvert)
- `objectid` - ID de l'objet (aide le housekeeper à trouver les données liées à l'objet)

Par exemple, la suppression d'un hôte avec deux éléments et un déclencheur remplit la table `housekeeper` comme suit :

```
+---------------+--------+----------+
| housekeeperid | object | objectid |
+---------------+--------+----------+
|             1 |      1 |    28724 |
|             2 |      0 |    59396 |
|             3 |      0 |    59397 |
+---------------+--------+----------+
```

Les déclencheurs de base de données alimentent la table `housekeeper` avec des tâches, sans vérifier les données liées à l'objet ; cette vérification est effectuée par le processus `housekeeper` lorsqu'il supprime les données :

- Pour les éléments supprimés (y compris les règles LLD) - supprime leurs données de toutes les tables d'historique et de tendances (`history`, `history_str`, `history_log`, `history_uint`, `history_text`, `history_bin`, `history_json`, `trends`, `trends_uint`). Supprime également leurs événements internes de la table `problem`.
- Pour les déclencheurs supprimés - supprime leurs événements des tables liées aux événements (`problem`, `event_symptom`, `events` ; la base de données supprime également les événements de la table `event_recovery`). Notifie également le processus `service manager` des événements supprimés.

Note:
Un processus distinct `trigger housekeeper` gère une tâche plus restreinte : la suppression des problèmes et des événements qui n'ont pas de déclencheur source connu. Sa fréquence d'exécution est contrôlée par `ProblemHousekeepingFrequency`.
Tant que la procédure de housekeeping des déclencheurs n'a pas démarré, les problèmes créés par des déclencheurs qui ont depuis été supprimés peuvent encore générer des problèmes de service et les attribuer à des services. Si votre configuration implique de nombreuses règles de calcul d'état des services basées sur des déclencheurs fréquemment découverts/non découverts, envisagez d'augmenter la fréquence de la procédure de housekeeping en ajustant le paramètre de configuration du serveur `ProblemHousekeepingFrequency`.

- Pour les services supprimés - supprime leurs problèmes de la table `problem` ; leurs événements de la table `events` ne sont supprimés qu'après la période de conservation des données configurée.
- Pour les hôtes/services découverts supprimés (découverte réseau) - supprime leurs événements de la table `events`.

Le processus `housekeeper` supprime uniquement les événements qui ne sont pas associés à des problèmes. Par exemple, un événement de problème/rétablissement obsolète ne sera pas supprimé s'il est associé à un problème ouvert. Lorsque `housekeeper` supprime des entités obsolètes, il supprime d'abord les problèmes, puis les événements.

Note:
Les tables utilisant le mode `partition` (tables partitionnées TimescaleDB) sont ignorées ; seules les tables utilisant le mode `regular` sont traitées.

Plates-formes prises en charge En raison des exigences de sécurité et du caractère critique de l'exploitation du serveur, UNIX est le seul système d'exploitation capable d'offrir de manière constante les performances, la tolérance aux pannes et la résilience nécessaires. Zabbix fonctionne sur les versions leaders du marché.

Le serveur Zabbix est testé sur les plates-formes suivantes :

- Linux
- Solaris
- AIX
- HP-UX
- Mac OS X
- FreeBSD
- OpenBSD
- NetBSD
- SCO Open Server

Note:
Zabbix peut également fonctionner sur d'autres systèmes d'exploitation de type Unix.

Paramètres régionaux Notez que le serveur nécessite des paramètres régionaux UTF-8 afin que certains éléments textuels puissent être interprétés correctement. La plupart des systèmes Unix modernes ont des paramètres régionaux UTF-8 par défaut, cependant, sur certains systèmes, il peut être nécessaire de les définir explicitement.

Configuration de haute disponibilité pour les serveurs Zabbix

Vue d'ensemble

La haute disponibilité (HA) est généralement requise dans les infrastructures critiques qui ne peuvent pratiquement se permettre aucune interruption de service. Ainsi, pour tout service susceptible de tomber en panne, il doit exister une option de basculement permettant de prendre le relais si le service actuel échoue.

Zabbix propose une solution de haute disponibilité **native**, facile à mettre en place et ne nécessitant aucune expertise préalable en HA. La HA native de Zabbix peut être utile pour ajouter une couche de protection supplémentaire contre les défaillances logicielles/matérielles du serveur Zabbix ou pour réduire les temps d'arrêt liés à la maintenance.

En mode haute disponibilité de Zabbix, plusieurs serveurs Zabbix sont exécutés comme des nœuds dans un cluster. Tandis qu'un serveur Zabbix du cluster est actif, les autres sont en veille, prêts à prendre le relais si nécessaire.



Le passage à la HA de Zabbix n'est pas contraignant. Vous pouvez revenir à un fonctionnement autonome à tout moment.

Voir aussi: Détails d'implémentation

Activation du cluster HA

Démarrage du serveur Zabbix en tant que nœud de cluster

Deux paramètres sont requis dans la configuration du serveur pour démarrer un serveur Zabbix en tant que nœud de cluster :

- Le paramètre `HANodeName` doit être spécifié pour chaque serveur Zabbix qui sera un nœud de cluster HA.

Il s'agit d'un identifiant de nœud unique (par exemple, `zabbix-node-01`) sous lequel le serveur sera référencé dans les configurations des agents et des proxys. Si vous ne spécifiez pas `HANodeName`, le serveur démarrera en mode autonome.

- Le paramètre `NodeAddress` doit être spécifié pour chaque nœud.

Le paramètre `NodeAddress` (`address:port`) sera utilisé par l'interface Zabbix pour se connecter au nœud de serveur actif. `NodeAddress` doit correspondre à l'adresse IP ou au nom FQDN du serveur Zabbix concerné.

Redémarrez tous les serveurs Zabbix après avoir modifié les fichiers de configuration. Ils démarreront alors en tant que nœuds de cluster. Le nouvel état des serveurs peut être consulté dans *Rapports* > *Informations système*, ainsi qu'en exécutant :

```
zabbix_server -R ha_status
```

Cette commande d'exécution consignera l'état actuel du cluster HA dans le journal du serveur Zabbix (ainsi que sur la sortie standard) :

```
Failover delay: 60 seconds
Cluster status:
   #  ID                        Name                 Address                Status      Last Access
   1. ckzxxqg7u0001lsropenyzh3m zabbix-node-01       64.227.66.193:10051    standby     0s
   2. ckzxyqo1k00013frpq539e1jp zabbix-node-02       64.227.74.25:10051     active      3s
```

Préparation de l'interface

Assurez-vous que `address:port` du serveur Zabbix n'est pas défini dans le fichier de configuration de l'interface Zabbix (`zabbix.conf.php`).

```
// Uncomment and set to desired values to override Zabbix hostname/IP and port.
// $ZBX_SERVER                  = '';
// $ZBX_SERVER_PORT             = '';
```

L'interface Zabbix détectera automatiquement le nœud actif en lisant les paramètres de la table des nœuds dans la base de données Zabbix. L'adresse du nœud actif sera utilisée comme adresse du serveur Zabbix.

Configuration du proxy

Les nœuds du cluster HA (serveurs) doivent être répertoriés dans la configuration du proxy Zabbix passif ou actif.

Pour un proxy passif, les noms des nœuds doivent être répertoriés dans le paramètre Server du proxy, séparés par une **virgule**.

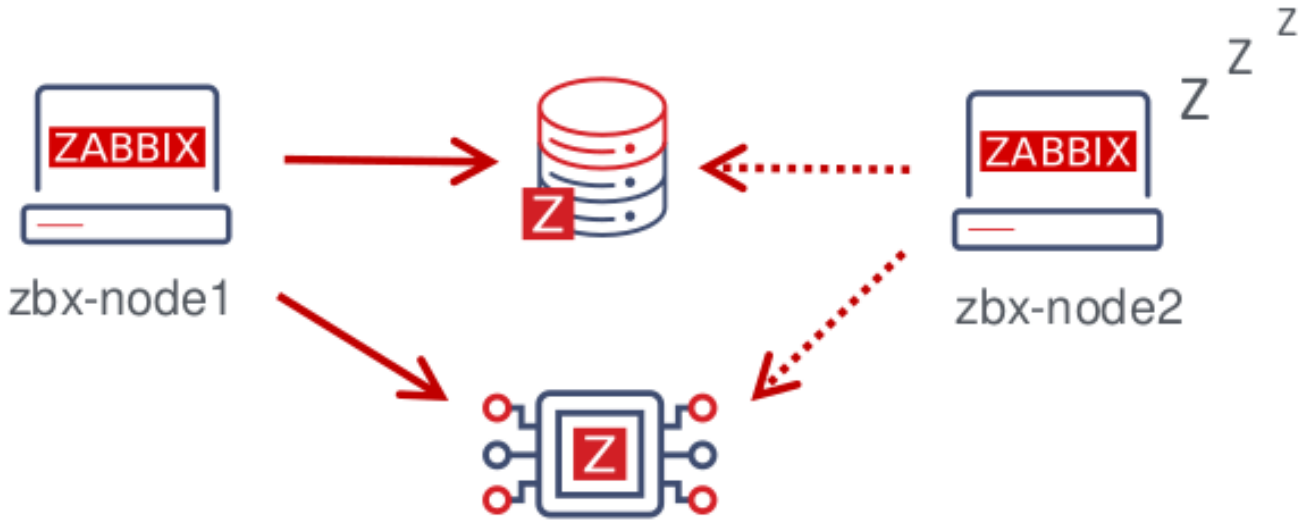
```
Server=zabbix-node-01,zabbix-node-02
```

Pour un proxy actif, les noms des nœuds doivent être répertoriés dans le paramètre Server du proxy, séparés par un **point-virgule**.

```
Server=zabbix-node-01;zabbix-node-02
```

Configuration de l'agent

Les nœuds du cluster HA (serveurs) doivent être listés dans la configuration de Zabbix agent ou Zabbix agent 2.



Pour activer les vérifications passives, les noms des nœuds doivent être listés dans le paramètre Server parameter, séparés par une **virgule**.

```
Server=zabbix-node-01,zabbix-node-02
```

Pour activer les vérifications actives, les noms des nœuds doivent être listés dans le paramètre ServerActive parameter. Notez que pour les vérifications actives, les nœuds doivent être séparés des autres serveurs par une virgule, tandis que les nœuds eux-mêmes doivent être séparés par un **point-virgule**, par exemple :

```
ServerActive=zabbix-node-01;zabbix-node-02
```

Basculement vers un noeud de secours

Zabbix effectuera automatiquement un basculement vers un autre noeud si le noeud actif s'arrête. Il doit y avoir au moins un noeud en statut de veille pour que le basculement puisse se produire.

Quelle est la rapidité du basculement ? Tous les noeuds mettent à jour leur dernière heure d'accès (et leur statut, s'il a changé) toutes les 5 secondes. Donc :

- Si le noeud actif s'arrête et parvient à signaler son statut comme "stopped", un autre noeud prendra le relais en moins de **5 secondes**.
- Si le noeud actif s'arrête/devient indisponible sans pouvoir mettre à jour son statut, les noeuds de veille attendront le **délai de basculement** + 5 secondes avant de prendre le relais.

Le délai de basculement est configurable, avec une plage prise en charge comprise entre 10 secondes et 15 minutes (une minute par défaut). Pour modifier le délai de basculement, vous pouvez exécuter :

```
zabbix_server -R ha_set_failover_delay=5m
```

Gestion du cluster HA

L'état actuel du cluster HA peut être géré à l'aide des options dédiées de contrôle à l'exécution :

- `ha_status` - consigne l'état du cluster HA dans le journal du serveur Zabbix (ainsi que dans la sortie standard) ;
- `ha_remove_node=target` - supprime un nœud HA identifié par son `<target>` - nom ou ID du nœud (le nom/ID peut être obtenu à partir de la sortie de l'exécution de `ha_status`), par exemple :

```
zabbix_server -R ha_remove_node=zabbix-node-02
```

Notez que les nœuds actifs/en veille ne peuvent pas être supprimés.

- `ha_set_failover_delay=delay` - définit le délai de basculement HA (entre 10 secondes et 15 minutes ; les suffixes de temps sont pris en charge (par exemple, `30s`, `1m`)).

L'état des nœuds peut être surveillé :

- Dans *Rapports* > *Informations système*.
- Dans le widget de tableau de bord *Informations système*.
- À l'aide de l'option de contrôle à l'exécution `ha_status` du serveur (voir ci-dessus).

L'élément interne `zabbix[cluster,discovery,nodes]` peut être utilisé pour la découverte des nœuds, car il renvoie un JSON contenant les informations sur les nœuds à haute disponibilité.

Désactivation du cluster HA

Pour désactiver un cluster à haute disponibilité :

1. Faites des copies de sauvegarde des fichiers de configuration.
2. Arrêtez les nœuds de secours.
3. Supprimez le paramètre HANodeName du serveur principal actif.
4. Redémarrez le serveur principal (il démarrera en mode autonome).

Mise à niveau du cluster HA

Pour effectuer une mise à niveau majeure de version pour les nœuds HA :

1. Arrêtez tous les nœuds.
2. Créez une sauvegarde complète de la base de données.
3. Si la base de données utilise la réplication, assurez-vous que tous les nœuds sont synchronisés et ne présentent aucun problème. N'effectuez pas la mise à niveau si la réplication est rompue.
4. Sélectionnez un seul nœud qui effectuera la mise à niveau de la base de données, passez sa configuration en mode autonome en commentant `HANodeName` et mettez-le à niveau.
5. Assurez-vous que la mise à niveau de la base de données est entièrement terminée (*Informations système* doit afficher que le serveur Zabbix est en cours d'exécution).
6. Redémarrez le nœud en mode HA.
7. Mettez à niveau et démarrez le reste des nœuds (il n'est pas nécessaire de les passer en mode autonome, car la base de données est déjà mise à niveau à ce stade).

Lors d'une mise à niveau mineure de version, il suffit de mettre à niveau le premier nœud, de vérifier qu'il a bien été mis à niveau et qu'il est en cours d'exécution, puis de lancer la mise à niveau du nœud suivant.

Détails d'implémentation

Le cluster de haute disponibilité (HA) est une solution optionnelle et elle est prise en charge pour le serveur Zabbix. La solution HA native est conçue pour être simple d'utilisation, elle fonctionne entre plusieurs sites et n'impose pas d'exigences particulières pour les bases de données reconnues par Zabbix. Les utilisateurs sont libres d'utiliser la solution HA native de Zabbix ou une solution HA tierce, selon ce qui répond le mieux aux exigences de haute disponibilité de leur environnement.

La solution se compose de plusieurs instances ou nœuds `zabbix_server`. Chaque nœud :

- Est configuré séparément.
- Utilise la même base de données.
- Peut avoir plusieurs modes : actif, veille, indisponible, arrêté.

Un seul nœud peut être actif (en fonctionnement) à la fois. Un nœud en veille n'exécute qu'un seul processus - le gestionnaire HA. Un nœud en veille n'effectue aucune collecte de données, aucun traitement ni aucune autre activité régulière du serveur ; il n'écoute sur aucun port ; il a un nombre minimal de connexions à la base de données.

Les nœuds actifs et en veille mettent à jour leur heure de dernier accès toutes les 5 secondes. Chaque nœud en veille surveille l'heure de dernier accès du nœud actif. Si l'heure de dernier accès du nœud actif dépasse de plus de 'failover delay' secondes, le nœud en veille devient lui-même le nœud actif et attribue le statut `unavailable` au nœud précédemment actif.

Le nœud actif surveille sa propre connectivité à la base de données - si elle est perdue pendant plus de `failover delay`−5 secondes, il doit arrêter tout traitement et passer en mode veille. Le nœud actif surveille également l'état des nœuds en veille - si l'heure de dernier accès d'un nœud en veille dépasse de plus de 'failover delay' secondes, le nœud en veille se voit attribuer le statut `unavailable`.

Les nœuds sont conçus pour être compatibles entre les versions mineures de Zabbix.

Paramètres de configuration: serveur Zabbix

Vue d'ensemble

Les paramètres de configuration permettent de personnaliser le comportement du serveur Zabbix.

Indiquez les valeurs des paramètres dans le fichier de configuration du serveur Zabbix (`zabbix_server.conf`). Tous les paramètres sont **facultatifs**, sauf indication explicite contraire.

Cette page est structurée comme suit :

- Vue d'ensemble des paramètres (pour afficher tous les détails d'un paramètre, cliquez sur son nom)
- Détails des paramètres

Paramètre	Description
AlertScriptsPath	Indiquer l'emplacement des scripts d'alerte.
AllowRoot	Autoriser le serveur à s'exécuter en tant que `root`.
AllowSoftwareUpdateCheck	Autoriser la réception d'informations sur les mises à jour logicielles depuis `zabbix.com`.
AllowUnsupportedDBVersions	Autoriser le serveur à fonctionner avec des versions de base de données non prises en charge.
BridgeAdapterURL	Définir l'URL HTTP(S) de l'adaptateur de pont Zabbix.
CacheSize	Définir la taille du cache de configuration (en octets).
CacheUpdateFrequency	Définir la fréquence de mise à jour du cache de configuration (en secondes).
DBHost	Indiquer l'hôte de la base de données (ou le répertoire du socket).
DBName	Indiquer le nom de la base de données. **Obligatoire**.
DBPassword	Indiquer le mot de passe de la base de données.
DBPort	Indiquer le port de l'hôte de la base de données.
DBSchema	Indiquer le nom du schéma de la base de données. Utilisé pour PostgreSQL.
DBSocket	Indiquer le chemin du fichier socket MySQL.
DBUser	Indiquer l'utilisateur de la base de données.
DBTLSConnect	Indiquer une valeur pour imposer une connexion TLS à la base de données.
DBTLSCAFile	Indiquer le chemin complet du fichier contenant les certificats de l'autorité de certification (CA) de niveau supérieur pour la vérification du certificat de la base de données.
DBTLSCertFile	Indiquer le chemin complet du fichier contenant le certificat du serveur Zabbix pour l'authentification auprès de la base de données.
DBTLSKeyFile	Indiquer le chemin complet du fichier contenant la clé privée pour l'authentification auprès de la base de données.
DBTLSCipher	Indiquer la liste des chiffrements de chiffrement que le serveur Zabbix autorise pour les protocoles TLS jusqu'à TLS v1.2 inclus. Pris en charge uniquement pour MySQL.
DBTLSCipher13	Indiquer la liste des suites de chiffrement que le serveur Zabbix autorise pour le protocole TLS v1.3. Prise en charge uniquement pour MySQL, à partir de la version 8.0.16.
DebugLevel	Définir le niveau de débogage.
EnableGlobalScripts	Activer les scripts globaux sur le serveur Zabbix.
EnableMobileDevices	Activer la prise en charge des appareils mobiles sur le serveur Zabbix.
ExportDir	Indiquer un répertoire pour activer l'exportation en temps réel des événements, de l'historique et des tendances.
ExportFileSize	Définir la taille maximale du fichier d'exportation (en octets).
ExportType	Indiquer les types d'entités à exporter en temps réel (événements, historique, tendances).
ExternalScripts	Indiquer l'emplacement des scripts externes.
Fping6Location	Indiquer l'emplacement de fping6.
FpingLocation	Indiquer l'emplacement de fping.
FrontendAllowedIP	Indiquer les adresses IP ou plages CIDR autorisées à se connecter depuis l'interface.
HANodeName	Indiquer le nom du nœud dans un cluster à haute disponibilité.
HistoryCacheSize	Définir la taille du cache de l'historique.
HistoryIndexCacheSize	Définir la taille du cache des index de l'historique.
HistoryProvider	Définir le fournisseur de stockage de l'historique (Elasticsearch, ClickHouse ou SQL).
HistoryStorageDateIndex	Ce paramètre est **obsolète** (utiliser HistoryProvider à la place). Activer le stockage des valeurs d'historique dans différents index selon la date.
HistoryStorageURL	Ce paramètre est **obsolète** (utiliser HistoryProvider à la place). Indiquer l'URL du stockage de l'historique.
HistoryStorageTypes	Ce paramètre est **obsolète** (utiliser HistoryProvider à la place). Indiquer les types de valeurs à envoyer au stockage de l'historique.
HousekeepingFrequency	Définir la fréquence de la procédure de maintenance (en heures).
Include	Indiquer un répertoire ou des fichiers individuels à inclure dans le fichier de configuration.
JavaGateway	Indiquer l'adresse IP (ou le nom d'hôte) de Java gateway Zabbix.
JavaGatewayPort	Indiquer le port de Java gateway Zabbix.
ListenBacklog	Définir le nombre maximal de connexions en attente dans la file TCP.
ListenIP	Indiquer l'adresse IP d'écoute du trapper.
ListenPort	Indiquer le port d'écoute du trapper.
LoadModule	Indiquer le module à charger au démarrage du serveur.
LoadModulePath	Indiquer le chemin complet de l'emplacement des modules du serveur.
LogFile	Indiquer le fichier journal du serveur.
LogFileSize	Définir la taille maximale du fichier journal.
LogSlowQueries	Définir la durée maximale d'exécution d'une requête de base de données avant sa consignation (en millisecondes).
LogType	Indiquer le type de sortie du journal.
MaxConcurrentChecksPerPoller	Définir le nombre maximal de vérifications asynchrones pouvant être exécutées simultanément par chaque poller HTTP agent, poller agent ou poller SNMP.

Paramètre	Description
MaxHousekeeperDelete	Définir le nombre maximal de lignes à supprimer par tâche au cours d'un cycle de maintenance.
NodeAddress	Indiquer l'adresse IP ou le nom d'hôte, avec un port facultatif, pour remplacer la manière dont l'interface se connecte au serveur.
PidFile	Indiquer le fichier PID.
ProblemHousekeepingFrequency	Définir la fréquence de suppression des problèmes associés aux déclencheurs supprimés.
ProxyConfigFrequency	Définir la fréquence à laquelle le serveur envoie les données de configuration au proxy Zabbix.
ProxyDataFrequency	Définir la fréquence à laquelle le serveur demande les données d'historique au proxy Zabbix.
ServiceManagerSyncFrequency	Définir la fréquence de synchronisation de la configuration d'un gestionnaire de services.
SMSDevices	Indiquer les fichiers de modem autorisés à utiliser le serveur Zabbix.
SNMPTrapperFile	Indiquer le fichier temporaire utilisé pour transmettre les données du démon de traps SNMP au serveur.
SocketDir	Indiquer le répertoire de stockage des sockets IPC utilisés par les services internes de Zabbix.
SourceIP	Indiquer l'adresse IP source.
SSHKeyLocation	Indiquer l'emplacement des clés publiques et privées pour les vérifications et actions SSH.
SSLCertLocation	Indiquer l'emplacement des fichiers de certificat du client SSL pour l'authentification du client.
SSLKeyLocation	Indiquer l'emplacement des fichiers de clé privée SSL pour l'authentification du client.
SSLCALocation	Indiquer l'emplacement des fichiers de l'autorité de certification (CA) pour la vérification du certificat du serveur SSL.
StartAgentPollers	Définir le nombre d'instances pré-forkées de pollers agent Zabbix asynchrones.
StartAlerters	Définir le nombre d'instances pré-forkées d'alerteurs.
StartBrowserPollers	Définir le nombre d'instances pré-forkées de pollers d'éléments navigateur.
StartConnectors	Définir le nombre d'instances pré-forkées de workers de connecteurs.
StartDBSyncers	Définir le nombre d'instances pré-forkées de synchroniseurs d'historique.
StartDiscoverers	Définir le nombre d'instances pré-forkées de workers de découverte.
StartEscalators	Définir le nombre d'instances pré-forkées d'escaladeurs.
StartHistoryPollers	Définir le nombre d'instances pré-forkées de pollers d'historique.
StartHTTPAgentPollers	Définir le nombre d'instances pré-forkées de pollers HTTP agent asynchrones.
StartHTTPPollers	Définir le nombre d'instances pré-forkées de pollers HTTP.
StartIPMIPollers	Définir le nombre d'instances pré-forkées de pollers IPMI.
StartJavaPollers	Définir le nombre d'instances pré-forkées de pollers Java.
StartLLDProcessors	Définir le nombre d'instances pré-forkées de workers de découverte de bas niveau (LLD).
StartODBCPollers	Définir le nombre d'instances pré-forkées de pollers ODBC.
StartPingers	Définir le nombre d'instances pré-forkées de pingers ICMP.
StartPollersUnreachable	Définir le nombre d'instances pré-forkées de pollers pour les hôtes inaccessibles (y compris IPMI et Java).
StartPollers	Définir le nombre d'instances pré-forkées de pollers.
StartPreprocessors	Définir le nombre d'instances pré-démarrées de workers de prétraitement.
StartProxyPollers	Définir le nombre d'instances pré-forkées de pollers pour les proxy passifs.
StartReportWriters	Définir le nombre d'instances pré-forkées de rédacteurs de rapports.
StartSNMPPollers	Définir le nombre d'instances pré-forkées de pollers SNMP asynchrones.
StartSNMPTrapper	Définir sur « 1 » pour démarrer un processus trapper SNMP.
StartTelemetryQueryPollers	Définir le nombre d'instances pré-forkées de pollers de requêtes de télémétrie asynchrones.
StartTimers	Définir le nombre d'instances pré-forkées de temporisateurs.
StartTrappers	Définir le nombre d'instances pré-forkées de trappers.
StartVMwareCollectors	Définir le nombre d'instances pré-forkées de collecteurs VMware.
StatsAllowedIP	Indiquer les adresses IP ou noms DNS des instances Zabbix externes autorisées à demander des statistiques.
TelemetryProvider	Définir la source des données de télémétrie.
Timeout	Indiquer le délai d'attente (en secondes) pour établir une connexion et échanger des données avec le proxy Zabbix, l'agent, le service web et pour les vérifications SNMP (à l'exception des éléments SNMP `walk[OID]` et `get[OID]`).
TLSCAFile	Indiquer le chemin complet du fichier contenant les certificats de l'autorité de certification (CA) de niveau supérieur pour la vérification des certificats des pairs, utilisé pour les communications chiffrées entre les composants Zabbix.
TLSCertFile	Indiquer le chemin complet du fichier contenant le certificat du serveur ou la chaîne de certificats, utilisé pour les communications chiffrées entre les composants Zabbix.
TLSCipherAll	Indiquer la chaîne de priorité GnuTLS ou la chaîne de chiffrement OpenSSL (TLS 1.2). Remplacer les critères de sélection par défaut des suites de chiffrement pour le chiffrement basé sur un certificat ou une PSK.
TLSCipherAll13	Indiquer la chaîne de chiffrement pour OpenSSL 1.1.1 ou version ultérieure avec TLS 1.3. Remplacer les critères de sélection par défaut des suites de chiffrement pour le chiffrement basé sur un certificat ou une PSK.

Paramètre	Description
TLSCipherCert	Indiquer la chaîne de priorité GnuTLS ou la chaîne de chiffrement OpenSSL (TLS 1.2). Remplacer les critères de sélection par défaut des suites de chiffrement pour le chiffrement basé sur un certificat.
TLSCipherCert13	Indiquer la chaîne de chiffrement pour OpenSSL 1.1.1 ou version ultérieure avec TLS 1.3. Remplacer les critères de sélection par défaut des suites de chiffrement pour le chiffrement basé sur un certificat.
TLSCipherPSK	Indiquer la chaîne de priorité GnuTLS ou la chaîne de chiffrement OpenSSL (TLS 1.2). Remplacer les critères de sélection par défaut des suites de chiffrement pour le chiffrement basé sur une PSK.
TLSCipherPSK13	Indiquer la chaîne de chiffrement pour OpenSSL 1.1.1 ou version ultérieure avec TLS 1.3. Remplacer les critères de sélection par défaut des suites de chiffrement pour le chiffrement basé sur une PSK.
TLSCRLFile	Indiquer le chemin complet du fichier contenant les certificats révoqués. Ce paramètre est utilisé pour les communications chiffrées entre les composants Zabbix.
TLSFrontendAccept	Indiquer le niveau de chiffrement des connexions entrantes depuis l'interface.
TLSFrontendCertIssuer	Indiquer l'émetteur de certificat autorisé pour l'interface.
TLSFrontendCertSubject	Indiquer le sujet de certificat autorisé pour l'interface.
TLSKeyFile	Indiquer le chemin complet du fichier contenant la clé privée du serveur, utilisée pour les communications chiffrées entre les composants Zabbix.
TLSListen	Exiger des connexions exclusivement TLS sur le socket du trapper.
TmpDir	Indiquer le répertoire temporaire.
TrapperTimeout	Indiquer le délai d'attente du trapper, en secondes, pour : - la récupération des données historiques depuis le proxy Zabbix ; - l'envoi des données de configuration au proxy Zabbix ; - l'exécution de scripts globaux ou de commandes à distance sur le serveur Zabbix.
TrendCacheSize	Définir la taille du cache des tendances.
TrendFunctionCacheSize	Définir la taille du cache des fonctions de tendance.
UnavailableDelay	Définir la fréquence de vérification de la disponibilité de l'hôte pendant la période d'indisponibilité.
UnreachableDelay	Définir la fréquence de vérification de la disponibilité de l'hôte pendant la période d'inaccessibilité.
UnreachablePeriod	Définir après combien de secondes d'inaccessibilité l'hôte doit être considéré comme indisponible.
User	Abandonner les privilèges pour utiliser un utilisateur spécifique existant sur le système.
ValueCacheSize	Définir la taille du cache des valeurs d'historique.
Vault	Indiquer le fournisseur de coffre-fort.
VaultAppRoleID	Indiquer l'ID AppRole du coffre-fort HashiCorp.
VaultAppSecretID	Indiquer l'ID secret AppRole du coffre-fort HashiCorp.
VaultDBPath	Indiquer l'emplacement de récupération des identifiants de la base de données par clés.
VaultPrefix	Indiquer un préfixe personnalisé pour le chemin ou la requête du coffre-fort.
VaultTLSCertFile	Indiquer le nom du fichier de certificat SSL utilisé pour l'authentification du client.
VaultTLSKeyFile	Indiquer le nom du fichier de clé privée SSL utilisé pour l'authentification du client.
VaultToken	Indiquer le jeton d'authentification du coffre-fort HashiCorp.
VaultURL	Indiquer l'URL du serveur de coffre-fort.
VMwareCacheSize	Définir la taille de la mémoire partagée utilisée pour stocker les données VMware.
VMwareFrequency	Définir la fréquence de collecte des données auprès d'un service VMware donné, en secondes.
VMwarePerfFrequency	Définir la fréquence de récupération des statistiques des compteurs de performance auprès d'un service VMware donné, en secondes.
VMwareTimeout	Définir le nombre maximal de secondes pendant lesquelles un `vmware collector` attend une réponse du service VMware.
WebDriverURL	Indiquer l'URL de l'interface WebDriver.
WebServiceURL	Indiquer l'URL du service web au format `http[s]://host:port/report`.

Notez que :

- Les valeurs par défaut correspondent aux valeurs par défaut des démons, et non aux valeurs des fichiers de configuration fournis.
- Les valeurs prennent en charge les variables d'environnement.
- Zabbix prend uniquement en charge les fichiers de configuration encodés en UTF-8 sans BOM.
- Les commentaires commençant par # sont pris en charge uniquement au début de la ligne.

Warning:
Un trop grand nombre de processus peut épuiser la limite de descripteurs de fichiers par processus du gestionnaire de prétraitement.

Ces processus assurent la collecte des données par les pollers, les pollers inaccessibles, les pollers ODBC, les pollers HTTP, les pollers Java, les pingers, les trappers et les pollers de proxy, ainsi que par le gestionnaire IPMI, le trapper SNMP, les workers de prétraitement et les workers de découverte. L'épuisement de la limite de descripteurs de fichiers entraînera l'**arrêt** du serveur Zabbix, généralement peu après le démarrage, mais parfois après un délai plus long.

Pour éviter ces problèmes, consultez le fichier de configuration du serveur Zabbix afin d'optimiser le nombre de vérifications et de processus simultanés. En outre, si nécessaire, vérifiez et ajustez les limites du système afin de vous assurer que la limite de descripteurs de fichiers est suffisamment élevée.

Détails du paramètre

AlertScriptsPath

Spécifie l'emplacement du script d'alerte. Dépend de la variable d'installation définie à la compilation `datadir`.

Par défaut: `/usr/local/share/zabbix/alertscripts`

AllowRoot

Autoriser le serveur à s'exécuter en tant que `root`. Si cette option est désactivée et que le serveur est démarré par `root`, le serveur tentera plutôt de basculer vers l'utilisateur `zabbix`. N'a aucun effet s'il est démarré sous un utilisateur standard.

Par défaut: 0
 Valeurs: 0 - ne pas autoriser; 1 - autoriser

AllowSoftwareUpdateCheck

Autoriser l'interface Zabbix à recevoir des informations sur les mises à jour logicielles depuis `zabbix.com`.

Par défaut: 1
 Valeurs: 0 - ne pas autoriser; 1 - autoriser

AllowUnsupportedDBVersions

Autoriser le serveur à fonctionner avec des versions de base de données non prises en charge.

Par défaut: 0
 Valeurs: 0 - ne pas autoriser; 1 - autoriser

BridgeAdapterURL

Définit l'URL HTTP(S) de l'adaptateur bridge Zabbix.

Exemple :

```
BridgeAdapterURL=https://bridge-adapter.example.com:10054/rpc
```

CacheSize

Définit la taille du cache de configuration (en octets). Taille de la mémoire partagée utilisée pour stocker les données de l'hôte, de l'élément et du déclencheur.

Par défaut: 32M
 Plage: 128K-64G

CacheUpdateFrequency

Définissez la fréquence de mise à jour du cache de configuration (en secondes). Voir aussi les options de contrôle à l'exécution.

Par défaut: 10
 Plage: 1-3600

DBHost

Spécifiez l'hôte de la base de données (ou le répertoire du socket).

Pour MySQL :

- `localhost` ou une chaîne vide utilise le socket UNIX par défaut.

Pour PostgreSQL :

- `localhost` est résolu via DNS (généralement vers 127.0.0.1).
- Une chaîne vide utilise le socket UNIX par défaut.
- Un chemin (par exemple, `/var/run/pgbouncer`) entraîne l'utilisation du socket UNIX à ce chemin.
- Une liste séparée par des virgules peut contenir plusieurs valeurs `host:port`. Chaque hôte est essayé dans l'ordre jusqu'à ce qu'une connexion en lecture-écriture soit établie.

Exemple : `DBHost=localhost:5431,127.0.0.1:20051,zabbix.domain,[::1]:30051,[12fc::1]`

Par défaut : `localhost`

DBName

Spécifiez le nom de la base de données.

Obligatoire: **Oui**

DBPassword

Spécifiez le mot de passe de la base de données. Commentez cette ligne si aucun mot de passe n'est utilisé.

DBPort

Spécifiez le port de l'hôte de la base de données Zabbix (voir `DBHost`).
 `DBSocket` et `DBPort` s'excluent mutuellement dans la configuration du serveur. Spécifiez-en un seul, ou laissez les deux non définis.

Valeur par défaut pour MySQL: 3306

Valeur par défaut pour PostgreSQL: 5432

Plage: 1024-65535

DBSchema

Spécifiez le nom du schéma de base de données. Utilisé pour PostgreSQL.

DBSocket

Spécifiez le chemin d'accès au fichier socket MySQL.
 `DBSocket` et `DBPort` s'excluent mutuellement dans la configuration du serveur. Spécifiez-en un seul, ou laissez les deux non définis.

DBUser

Spécifiez l'utilisateur de la base de données.

DBTLSConnect

Spécifiez une valeur pour imposer une connexion TLS à la base de données : - *required* - se connecter en utilisant TLS ; - *verify_ca* - se connecter en utilisant TLS et vérifier le certificat ; - *verify_full* - se connecter en utilisant TLS, vérifier le certificat et vérifier que l'identité de la base de données spécifiée par `DBHost` correspond à son certificat.

Avec MySQL, à partir de la version 5.7.11, et PostgreSQL, les valeurs suivantes sont prises en charge : `required`, `verify_ca`, `verify_full`.
 Avec MariaDB, à partir de la version 10.2.6, les valeurs `required` et `verify_full` sont prises en charge.
Par défaut, aucune option n'est définie et le comportement dépend de la configuration de la base de données.

DBTLSCAFile

Spécifiez le chemin d'accès complet du fichier contenant le ou les certificats de l'autorité de certification racine pour la vérification du certificat de la base de données.

Obligatoire: non (oui, si DBTLSConnect est défini sur `verify_ca` ou `verify_full`)

DBTLSCertFile

Spécifiez le chemin d'accès complet du fichier contenant le certificat du serveur Zabbix pour l'authentification à la base de données.

DBTLSKeyFile

Spécifiez le chemin d'accès complet du fichier contenant la clé privée utilisée pour l'authentification à la base de données.

DBTLSCipher

Spécifiez la liste des chiffrements de chiffrement que le serveur Zabbix autorise pour les protocoles TLS jusqu'à TLS v1.2. Pris en charge uniquement pour MySQL.

DBTLSCipher13

Spécifiez la liste des suites de chiffrement que le serveur Zabbix autorise pour le protocole TLS v1.3. Pris en charge uniquement pour MySQL, à partir de la version 8.0.16.

DebugLevel

Définissez le niveau de débogage : - *0* - informations de base sur le démarrage et l'arrêt des processus Zabbix ; - *1* - informations critiques ; - *2* - informations d'erreur ; - *3* - avertissements ; - *4* - pour le débogage (génère beaucoup d'informations) ; - *5* - débogage étendu (génère encore plus d'informations).
 Voir aussi les options de contrôle à l'exécution.

Valeur par défaut : 3
 Plage : 0-5

EnableGlobalScripts

Active les scripts globaux sur le serveur Zabbix.
 Remarque: l'exécution des scripts globaux est activée par défaut. Cependant, pour les nouvelles installations à partir de Zabbix 7.0, EnableGlobalScripts est explicitement défini sur 0 (désactivé).

Par défaut: 1
 Valeurs: 0 - désactiver; 1 - activer

EnableMobileDevices

Active la prise en charge des appareils mobiles sur le serveur Zabbix.

Lorsque cette option est activée, le serveur Zabbix traite les demandes d'ajout ou de suppression d'appareils mobiles et envoie des notifications push à ces appareils.

Valeur par défaut : 1
 Valeurs : 0 - désactiver ; 1 - activer

ExportDir

Spécifiez un répertoire pour activer l'export en temps réel des événements, de l'historique et des tendances.

ExportFileSize

Définissez la taille maximale du fichier d'exportation (en octets). Utilisé pour la rotation si `ExportDir` est défini.

Par défaut: 1G
 Plage: 1M-1G

ExportType

Spécifiez les types d'entités pour l'export en temps réel (events, history, trends). Liste séparée par des virgules. Tous les types sont exportés par défaut. Valide uniquement si `ExportDir` est défini.
 Note que si `ExportType` est spécifié, mais que `ExportDir` ne l'est pas, il s'agit alors d'une erreur de configuration et le serveur ne démarrera pas.

Exemple pour l'export de history et trends :

```
ExportType=history,trends
```

Exemple pour l'export des events uniquement :

```
ExportType=events
```

ExternalScripts

Spécifiez l'emplacement du script externe. Dépend de la variable d'installation définie à la compilation `datadir`.

Par défaut : `/usr/local/share/zabbix/externalscripts`

Fping6Location

Spécifiez l'emplacement de fping6. Assurez-vous que le binaire fping6 appartient à root et que l'indicateur SUID est défini. Laissez vide (`Fping6Location=`) si votre utilitaire fping est capable de traiter les adresses IPv6.

Par défaut: `/usr/sbin/fping6`

FpingLocation

Spécifiez l'emplacement de fping. Assurez-vous que le binaire fping appartient à root et que l'attribut SUID est défini.

Par défaut : `/usr/sbin/fping`

FrontendAllowedIP

Spécifiez les adresses IP ou les plages CIDR autorisées à se connecter depuis l'interface. Liste séparée par des virgules. Par défaut, toutes les connexions sont acceptées pour les requêtes de l'interface. Si la prise en charge d'IPv6 est activée, alors `127.0.0.1`, `::127.0.0.1`, `::ffff:127.0.0.1` sont traités de la même manière et `::/0` autorisera n'importe quelle adresse IPv4 ou IPv6. `0.0.0.0/0` peut être utilisé pour autoriser n'importe quelle adresse IPv4.

Exemple :

```
FrontendAllowedIP=127.0.0.1,192.168.1.0/24,::1,2001:db8::/32,zabbix.example.com
```

HANodeName

Spécifiez le nom du nœud dans un cluster à haute disponibilité. Lorsque ce champ est vide, le serveur fonctionne en mode autonome et un nœud avec un nom vide est créé.

HistoryCacheSize

Définit la taille du cache d'historique (en octets). Taille de la mémoire partagée utilisée pour stocker les données d'historique.

Par défaut: 16M
 Plage: 128K-16G

HistoryIndexCacheSize

Définit la taille du cache d'index d'historique (en octets). Taille de la mémoire partagée utilisée pour indexer les données d'historique stockées dans le cache d'historique. La taille du cache d'index nécessite environ 100 octets pour mettre en cache un élément.

Par défaut: 4M
 Plage: 128K-16G

HistoryProvider

Définissez le fournisseur de stockage de l'historique en utilisant le format `<provider>[;<options>]`, où :

- `<provider>` - l'un des fournisseurs de stockage de l'historique pris en charge : `sql` (par défaut), `elasticsearch` ou `clickhouse`.
- `<options>` - paires `key=value` séparées par des virgules. Les valeurs peuvent être entre guillemets.

Options pour tous les fournisseurs de stockage de l'historique :

- `value_types` - définit les types de valeurs à stocker (`uint`, `dbl`, `str`, `log`, `text`, `json`). Chaque type ne peut être utilisé qu'une seule fois, et uniquement pour un seul fournisseur.
- `precache` - définissez 1 (par défaut) pour précharger en bloc les données du fournisseur de stockage de l'historique dans le cache de valeurs. Cela peut améliorer considérablement le réchauffement du cache de valeurs après un redémarrage.

Options pour Elasticsearch et ClickHouse :

- `url` - définit l'URL HTTP ou HTTPS utilisée pour se connecter au fournisseur de stockage de l'historique.
- `log_slow_queries` - définit la durée maximale (en millisecondes) qu'une requête au fournisseur de stockage de l'historique peut prendre avant d'être journalisée. Cette option devient disponible à partir de `DebugLevel=3`.
- `source_ip` - définit l'adresse IP source utilisée pour les connexions sortantes vers le fournisseur de stockage de l'historique.

Options pour Elasticsearch :

- `date_index` - définissez 1 pour activer le stockage des valeurs d'historique dans différents index en fonction de la date.

Options pour ClickHouse :

- `db` - définit le nom de la base de données.
- `username` - définit le nom d'utilisateur utilisé pour se connecter à la base de données.
- `password` - définit le mot de passe utilisé pour se connecter à la base de données.
- `ssl_cert_file`, `ssl_key_file`, `ssl_key_password`, `ssl_verify_peer`, `ssl_verify_host`, `ssl_ca_location`, `ssl_cert_location`, `ssl_key_location` - définissez ces options pour configurer une connexion sécurisée (TLS/SSL) à la base de données.

Exemple pour ClickHouse :

```
HistoryProvider=clickhouse;value_types="uint,dbl,str,log,text,json",url=http://localhost:8123,db=zabbix,us
```

Exemple pour Elasticsearch :

```
HistoryProvider=elasticsearch;value_types="uint,dbl,str,log,text,json",url=http://localhost:9200,date_inde
```

Exemple pour ClickHouse et Elasticsearch :

```
HistoryProvider=clickhouse;value_types="uint,dbl,str",url=http://localhost:8123,db=zabbix,username=zabbix,
HistoryProvider=elasticsearch;value_types="log,text,json",url=http://localhost:9200,date_index=0
```

HistoryStorageDateIndex

Ce paramètre est obsolète (utilisez plutôt HistoryProvider).
 Active le stockage des valeurs d'historique dans différents index en fonction de la date. Ce paramètre est utilisé pour la configuration d'Elasticsearch.

Par défaut: 0
 Valeurs: 0 - désactivé; 1 - activé

HistoryStorageURL

Ce paramètre est obsolète (utilisez plutôt HistoryProvider).
 Spécifiez l'URL de stockage de l'historique. Ce paramètre est utilisé pour la configuration Elasticsearch.

HistoryStorageTypes

Ce paramètre est obsolète (utilisez plutôt HistoryProvider).
 Spécifiez les types de valeurs à envoyer au stockage d'historique. Une liste séparée par des virgules. Ce paramètre est utilisé pour la configuration Elasticsearch.

Par défaut: `uint,dbl,str,log,text,json`

HousekeepingFrequency

Définissez la fréquence de la procédure de housekeeping (en heures). Le housekeeping consiste à supprimer les informations obsolètes de la base de données.
Note : afin d'éviter que le housekeeper ne soit surchargé (par exemple, lorsque les périodes d'historique et de tendances sont fortement réduites), au plus 4 fois `HousekeepingFrequency` heures d'informations obsolètes sont supprimées lors d'un cycle de housekeeping, pour chaque élément. Ainsi, si `HousekeepingFrequency` vaut 1, au plus 4 heures d'informations obsolètes (en commençant par l'entrée la plus ancienne) seront supprimées par cycle.
Note : pour réduire la charge au démarrage du serveur, le housekeeping est différé de 30 minutes après le démarrage du serveur. Ainsi, si `HousekeepingFrequency` vaut 1, la toute première procédure de housekeeping après le démarrage du serveur s'exécutera après 30 minutes, puis se répétera ensuite avec un délai d'une heure.
Il est possible de désactiver le housekeeping automatique en définissant `HousekeepingFrequency` à 0. Dans ce cas, la procédure de housekeeping ne peut être lancée que par l'option de contrôle à l'exécution `housekeeper_execute`, et la période d'informations obsolètes supprimée lors d'un cycle de housekeeping est égale à 4 fois la période écoulée depuis le dernier cycle de housekeeping, mais pas moins de 4 heures ni plus de 4 jours.
Voir aussi les options de contrôle à l'exécution et les détails sur la procédure de housekeeping.

Par défaut : 1
 Plage : 0-24

Inclure

Spécifiez un répertoire ou des fichiers individuels à inclure dans le fichier de configuration.
 Lors de l'installation de Zabbix, ce paramètre est défini sur `/usr/local/etc`, sauf si vous le modifiez au moment de la compilation.

Règles générales pour le paramètre `Include` :

- Le chemin peut être relatif à l'emplacement du fichier `zabbix_server.conf`.
- Vous pouvez spécifier plusieurs paramètres `Include` dans un même fichier de configuration.
- Zabbix traite le paramètre `Include` dès qu'il le lit, donc envisagez de le placer à la fin du fichier de configuration.

Règles pour les fichiers inclus :

- Chaque fichier doit être lisible.
- Chaque fichier doit avoir une syntaxe correcte. Sinon, le composant ne démarrera pas.

Règles pour l'inclusion de répertoires ou de modèles de fichiers :

- Ne supposez aucun ordre d'inclusion (par exemple, les fichiers ne sont pas inclus par ordre alphabétique).
- Ne définissez pas le même paramètre dans plus d'un fichier inclus (par exemple, pour remplacer un réglage général par un réglage spécifique), car vous ne pouvez pas prévoir quelle valeur sera utilisée.
- Faites attention aux copies de sauvegarde que certains éditeurs de texte créent automatiquement (par exemple, la modification de `include/my_specific.conf` peut créer `include/my_specific.conf.BAK`), car les deux fichiers seront inclus.

Pour inclure uniquement les fichiers pertinents d'un répertoire, utilisez le caractère générique (*) pour la correspondance de motifs.

Exemple :

```
Include=/absolute/path/to/config/files/*.conf
```

JavaGateway

Spécifiez l'adresse IP ou le nom d'hôte du Java gateway de Zabbix. Obligatoire uniquement si les collecteurs Java sont démarrés.

JavaGatewayPort

Spécifiez le port du Java gateway Zabbix.

Par défaut: 10052
 Plage: 1024-32767

ListenBacklog

Définit le nombre maximal de connexions en attente dans la file TCP.
La valeur par défaut est une constante codée en dur, qui dépend du système.
La valeur maximale prise en charge dépend également du système ; des valeurs trop élevées peuvent être tronquées silencieusement à la « valeur maximale spécifiée par l'implémentation ».

Par défaut: `SOMAXCONN`
 Plage: 0 - INT_MAX

ListenIP

Spécifiez l'adresse IP d'écoute du trapper. Une liste séparée par des virgules.
 Le trapper écoutera sur toutes les interfaces réseau si ce paramètre n'est pas spécifié.

Par défaut: `0.0.0.0`

ListenPort

Spécifiez le port d'écoute du trapper.

Par défaut: 10051
 Plage: 1024-32767

LoadModule

Spécifiez le module à charger au démarrage du serveur. Les modules servent à étendre les fonctionnalités du serveur. Le module doit se trouver dans le répertoire spécifié par LoadModulePath, ou le chemin doit précéder le nom du module. Si le chemin précédent est absolu (commence par /), LoadModulePath est ignoré.
 Formats :

- LoadModule=<module.so>
- LoadModule=<path/module.so>
- LoadModule=</abs_path/module.so>

Plusieurs paramètres LoadModule peuvent être spécifiés.

LoadModulePath

Spécifiez le chemin complet vers l'emplacement des modules du serveur. La valeur par défaut dépend des options de compilation.

LogFile

Spécifiez le fichier journal du serveur.

Obligatoire : Oui, si LogType est défini sur file ; sinon non

LogFileSize

Définissez la taille maximale du fichier journal (en MB).
0 - désactive la rotation automatique des journaux.
Note : si la limite de taille du fichier journal est atteinte et que la rotation du fichier échoue, quelle qu'en soit la raison, le fichier journal existant est tronqué et recréé.

Par défaut : 1
 Plage : 0-1024
 Obligatoire : Oui, si LogType est défini sur file ; sinon non

LogSlowQueries

Définit la durée maximale qu'une requête de base de données peut prendre avant d'être consignée (en millisecondes).
0 - ne pas consigner les requêtes lentes.
Cette option devient active à partir de DebugLevel=3.

Par défaut: 0
 Plage: 0-3600000

LogType

Spécifiez le type de sortie du journal : - *file* - écrire le journal dans le fichier spécifié par le paramètre LogFile ; - *system* - écrire le journal dans syslog ; - *console* - écrire le journal sur la sortie standard.

Par défaut : file

MaxConcurrentChecksPerPoller

Définissez le nombre maximal de vérifications asynchrones pouvant être exécutées simultanément par chaque collecteur HTTP agent, collecteur agent ou collecteur SNMP. Voir StartHTTPAgentPollers, StartAgentPollers et StartSNMPPollers.

Par défaut: 1000
 Plage: 1-1000

MaxHousekeeperDelete

Définit le nombre maximal de lignes à supprimer par tâche lors d'un cycle de housekeeping dans les tables history, history_str, history_log, history_uint, history_text, history_bin, history_json, trends, trends_uint et problem.
Les tâches de housekeeping existent dans la table housekeeper au format : [housekeeperid], [object], [objectid].
 Ce paramètre s'applique uniquement à la suppression des données laissées par des éléments déjà supprimés.
 S'il est défini à 0, aucune limite n'est appliquée.
 Dans ce cas, vous devez savoir ce que vous faites afin de ne pas surcharger la base de données.
 Lorsqu'un grand nombre d'éléments est supprimé, la charge sur la base de données augmente, car le housekeeper devra supprimer toutes les données d'historique que ces éléments avaient.
 Par exemple, si nous devons supprimer un seul prototype d'élément du modèle, mais que ce modèle est lié à 50 hôtes et que, pour chaque hôte, le prototype est développé en 100 éléments réels, 5000 éléments au total doivent être supprimés (1*50*100).
 Si 500 est défini pour MaxHousekeeperDelete (MaxHousekeeperDelete=500), le processus housekeeper devra supprimer jusqu'à 2500000 valeurs (5000*500) pour les éléments supprimés des tables history et trends en un seul cycle.

Voir aussi les détails sur la procédure de housekeeping.

Par défaut: 5000
 Plage: 0-1000000

NodeAddress

Spécifiez une adresse IP ou un nom d'hôte avec un port facultatif pour remplacer la manière dont l'interface se connecte au serveur.
Format : <address>[:<port>]

Si l'adresse IP ou le nom d'hôte n'est pas défini, la valeur de ListenIP sera utilisée. Si ListenIP n'est pas défini, la valeur localhost sera utilisée.
Si le port n'est pas défini, la valeur de ListenPort sera utilisée. Si ListenPort n'est pas défini, la valeur 10051 sera utilisée.

Cette option peut être remplacée par

l'adresse spécifiée dans la configuration de l'interface.

Voir aussi : paramètre HANodeName ; Activation de la haute disponibilité.

Valeur par défaut : `localhost:10051`

PidFile

Spécifiez le fichier PID.

Par défaut: `/tmp/zabbix_server.pid`

ProblemHousekeepingFrequency

Définit la fréquence de suppression des problèmes pour les déclencheurs supprimés (en secondes).
 Voir aussi les détails sur la procédure de housekeeping.

Par défaut: 60
 Plage: 1-3600

ProxyConfigFrequency

Définit la fréquence d'envoi des données de configuration au proxy Zabbix par le serveur. Utilisé uniquement pour les proxies en mode passif.

Par défaut: 10
 Plage: 1-604800

ProxyDataFrequency

Définit la fréquence de demande des données d'historique au proxy Zabbix par le serveur. Utilisé uniquement pour les proxies en mode passif.

Par défaut: 1
 Plage: 1-3600

ServiceManagerSyncFrequency

Définit la fréquence de synchronisation de la configuration d'un gestionnaire de services (en secondes).

Par défaut: 60
 Plage: 1-3600

SMSDevices

Spécifiez les fichiers de modem autorisés à être utilisés par le serveur Zabbix. Une liste séparée par des virgules.
 L'envoi de SMS n'est pas possible si ce paramètre n'est pas défini.

Exemple :

```
SMSDevices=/dev/ttyUSB0,/dev/ttyUSB1
```

SNMPTrapperFile

Spécifie le fichier temporaire utilisé pour transmettre les données du démon de trap SNMP au serveur.
Doit être identique à celui défini dans `zabbix_trap_receiver.pl` ou dans le fichier de configuration SNMPTT.

Par défaut: `/tmp/zabbix_traps.tmp`

SocketDir

Spécifiez le répertoire de stockage des sockets IPC utilisés par les services internes de Zabbix.

Par défaut: `/tmp`

SourceIP

Spécifiez l'adresse IP source pour :

- les connexions sortantes vers le proxy Zabbix et l'agent Zabbix
- les connexions sans agent (VMware, SSH, JMX, SNMP, Telnet et vérifications simples)
- les connexions de l'agent HTTP
- les requêtes HTTP JavaScript de l'élément script
- les requêtes HTTP JavaScript de prétraitement
- l'envoi des e-mails de notification (connexions au serveur SMTP)
- les notifications webhook (connexions HTTP JavaScript)
- les connexions au Vault

SSHKeyLocation

Spécifiez l'emplacement des clés publiques et privées pour les vérifications SSH et les actions.

SSLCertLocation

Indiquez l'emplacement des fichiers de certificat du client SSL pour l'authentification du client.
 Ce paramètre est utilisé pour la supervision Web, la communication avec Vault, le fournisseur de stockage de l'historique ClickHouse et la source de données de télémétrie ClickHouse.

SSLKeyLocation

Spécifiez l'emplacement des fichiers de clé privée SSL pour l'authentification du client.
 Ce paramètre est utilisé pour la supervision web, la communication avec Vault, le fournisseur de stockage de l'historique ClickHouse et la source de données de télémétrie ClickHouse.

SSLCALocation

Spécifiez l'emplacement des fichiers de l'autorité de certification (CA) utilisés pour la vérification des certificats du serveur SSL. S'il n'est pas défini, le répertoire à l'échelle du système sera utilisé.
 Notez que la valeur de ce paramètre sera définie comme option libcurl `CURLOPT_CAPATH`. Pour les versions de libcurl antérieures à 7.42.0, cela ne prend effet que si libcurl a été compilé pour utiliser OpenSSL. Pour plus d'informations, consultez la page web de cURL.
 Ce paramètre est utilisé pour la supervision web, l'authentification SMTP, les éléments agent HTTP, la communication avec Vault, le fournisseur de stockage de l'historique ClickHouse et la source de données de télémétrie ClickHouse.

StartAgentPollers

Définit le nombre d'instances pré-forkées de Zabbix agent pollers. Voir MaxConcurrentChecksPerPoller.

Par défaut: 1
 Plage: 0-1000

StartAlerters

Définit le nombre d'instances pré-forkées de alerters.

Par défaut: 3
 Plage: 1-100

StartBrowserPollers

Définit le nombre d'instances pré-forkées de pollers pour les éléments de navigateur.

Par défaut: 1
 Plage: 0-1000

StartConnectors

Définit le nombre d'instances pré-forkées de workers de connecteur. Le processus de gestionnaire de connecteur est démarré automatiquement lorsqu'un worker de connecteur est démarré.

Par défaut: 0
 Plage: 0-1000

StartDBSyncers

Définit le nombre d'instances pré-forkées de history syncers.
Remarque : soyez prudent lorsque vous modifiez cette valeur, car l'augmenter peut faire plus de mal que de bien. En règle générale, la valeur par défaut devrait suffire pour gérer jusqu'à 4000 NVPS.

Par défaut : 4
 Plage : 1-100

StartDiscoverers

Définit le nombre d'instances pré-forkées des workers de découverte[1].

Par défaut: 5
 Plage: 0-1000

StartEscalators

Définit le nombre d'instances pré-forkées de escalators.

Par défaut: 1
 Plage: 1-100

StartHistoryPollers

Définit le nombre d'instances pré-forkées de history pollers.
Requis uniquement pour les vérifications calculées.

Par défaut: 5
 Plage: 0-1000

StartHTTPAgentPollers

Définit le nombre d'instances pré-forkées des collecteurs de l'agent HTTP. Voir MaxConcurrentChecksPerPoller.

Par défaut: 1
 Plage: 0-1000

StartHTTPPollers

Définit le nombre d'instances pré-forkées de HTTP pollers[1].

Par défaut: 1
 Plage: 0-1000

StartIPMIPollers

Définit le nombre d'instances pré-forkées de IPMI pollers.

Par défaut: 0
 Plage: 0-1000

StartJavaPollers

Définit le nombre d'instances pré-forkées de Java pollers[1].

Par défaut: 0
 Plage: 0-1000

StartLLDProcessors

Définit le nombre d'instances pré-forkées de workers de découverte de bas niveau (LLD)[1].
 Le processus gestionnaire LLD est automatiquement démarré lorsqu'un worker LLD est démarré.

Par défaut: 2
 Plage: 1-100

StartODBCPollers

Définit le nombre d'instances pré-forkées de ODBC pollers[1].

Par défaut: 1
 Plage: 0-1000

StartPingers

Définit le nombre d'instances pré-forkées de ICMP pingers[1].

Par défaut: 1
 Plage: 0-1000

StartPollersUnreachable

Définit le nombre d'instances pré-forkées de pollers pour les hôtes injoignables (y compris IPMI et Java)[1].
 Au moins un poller pour les hôtes injoignables doit être en cours d'exécution si des pollers réguliers, IPMI ou Java sont démarrés.

Par défaut: 1
 Plage: 0-1000

StartPollers

Définit le nombre d'instances pré-forkées de pollers[1].

Par défaut: 5
 Plage: 0-1000

StartPreprocessors

Définissez le nombre d'instances de prétraitement worker[1] pré-démarrées. Le nombre de threads doit être défini à une valeur au moins égale au nombre de cœurs CPU disponibles. Il convient de définir davantage de workers si le prétraitement n'est pas limité par le CPU et comporte de nombreuses requêtes réseau.

Par défaut: 16
 Plage: 1-1000

StartProxyPollers

Définit le nombre d'instances pré-forkées de collecteurs pour les proxies passifs[1].

Par défaut: 1
 Plage: 0-250

StartReportWriters

Définit le nombre d'instances pré-forkées de report writers.
Si la valeur est définie sur 0, la génération planifiée des rapports est désactivée.
Le processus de gestionnaire de rapports est automatiquement démarré lorsqu'un report writer est démarré.

Par défaut: 0
 Plage: 0-100

StartSNMPPollers

Définit le nombre d'instances SNMP pollers pré-forkées. Voir `MaxConcurrentChecksPerPoller`.

Par défaut: 1
 Plage: 0-1000

StartSNMPTrapper

Définissez sur 1 pour démarrer un processus SNMP trapper.

Par défaut: 0
 Plage: 0-1

StartTelemetryQueryPollers

Définit le nombre d'instances pré-créées de pollers de requêtes Telemetry asynchrones. Voir MaxConcurrentChecksPerPoller.

Valeur par défaut : 1
 Plage : 0-1000

StartTimers

Définit le nombre d'instances pré-forkées de timers.
Les timers traitent les périodes de maintenance.

Par défaut: 1
 Plage: 1-1000

StartTrappers

Définit le nombre d'instances pré-forkées de trappers[1].
 Les trappers acceptent les connexions entrantes provenant de Zabbix sender, des agents actifs et des proxies actifs.

Par défaut: 5
 Plage: 0-1000

StartVMwareCollectors

Définit le nombre d'instances VMware collector pré-forkées.

Par défaut : 0
 Plage : 0-250

StatsAllowedIP

Spécifiez les adresses IP (éventuellement en notation CIDR) ou les noms DNS des instances Zabbix externes autorisées à demander des statistiques. Liste séparée par des virgules.
 Si ce paramètre n'est pas défini, aucune requête de statistiques ne sera acceptée.
 Si la prise en charge d'IPv6 est activée, alors `127.0.0.1`, `::127.0.0.1`, `::ffff:127.0.0.1` sont traitées de la même manière et `::/0` autorisera n'importe quelle adresse IPv4 ou IPv6. `0.0.0.0/0` peut être utilisé pour autoriser n'importe quelle adresse IPv4.

Exemple :

```
StatsAllowedIP=127.0.0.1,192.168.1.0/24,::1,2001:db8::/32,zabbix.example.com
```

TelemetryProvider

Définissez la source de données de télémétrie au format `<provider>[;<options>]` où :

- `<provider>` - l'une des sources de données de télémétrie prises en charge : `clickhouse`
- `<options>` - paires `key=value` séparées par des virgules. Les valeurs peuvent être placées entre guillemets.

Options pour ClickHouse :

- `url` - définissez l'URL HTTP ou HTTPS utilisée pour se connecter à la source de données de télémétrie.
- `db` - définissez le nom de la base de données.
- `username` - définissez le nom d'utilisateur utilisé pour se connecter à la base de données.
- `password` - définissez le mot de passe utilisé pour se connecter à la base de données.
- `ssl_cert_file`, `ssl_key_file`, `ssl_key_password`, `ssl_verify_peer`, `ssl_verify_host`, `ssl_ca_location`, `ssl_cert_location`, `ssl_key_location` - définissez ces options pour configurer une connexion sécurisée (TLS/SSL) à la base de données.
- `source_ip` - définissez l'adresse IP source utilisée pour les connexions sortantes vers la source de données de télémétrie.
- `vault_path` - définissez le chemin vers un secret Vault contenant les valeurs `username` et `password`. Ne l'utilisez pas avec les options `username` et `password`.

Exemple pour ClickHouse :

```
TelemetryProvider=clickhouse;url=http://localhost:8123,db=zabbix,username=zabbix,password=zabbix
```

Exemple pour ClickHouse avec Vault :

```
TelemetryProvider=clickhouse;url=http://localhost:8123,db=zabbix,vault_path=clickhouse
```

Timeout

Spécifiez combien de temps attendre (en secondes) pour établir la connexion et échanger des données avec le proxy, l'agent, le service web et les vérifications SNMP héritées (numéro ou chaîne OID unique).

Ce paramètre définit la durée de diverses opérations de communication :

- exécution de commandes distantes sur l'agent Zabbix
- exécution de commandes SSH/Telnet
- requêtes vers le service web Zabbix
- délai d'attente de communication pour la requête de test de type de média et la méthode `history.push`
- replanification des éléments lorsqu'une interface IPMI devient indisponible
- envoi d'une réponse au proxy Zabbix en cas d'échec d'échange de données en raison de problèmes de droits ou de chiffrement

- délai limite pour les sockets asynchrones IPC et les options de contrôle d'exécution
- connexions JMX
- récupération des statistiques depuis un proxy Zabbix ou un serveur distant
- envoi de réponses à l'interface Zabbix
- requêtes DNS du collecteur asynchrone
- réponse pour le heartbeat des vérifications actives
- récupération des données de l'agent Zabbix (valeurs) depuis les agents actifs
- récupération des données depuis Zabbix sender
- réponse lorsque le trapper ne parvient pas à traiter une requête
- envoi de la liste des vérifications actives à l'agent Zabbix

Ce délai d'attente **ne sera pas** utilisé pour les vérifications qui ont des paramètres de délai d'attente flexible configurés dans l'interface (au niveau global, du proxy ou de l'élément). Par exemple, les éléments SNMP `walk[OID]` et `get[OID]` utilisent le délai d'attente configuré dans l'interface ; les vérifications SNMP héritées utilisent toujours la valeur de délai d'attente du serveur.

Par défaut : 3
 Plage : 1-30

TLSCAFile

Spécifiez le chemin complet du fichier contenant les certificats de l'autorité de certification racine pour la vérification du certificat du pair, utilisé pour les communications chiffrées entre les composants Zabbix.

TLSCertFile

Spécifiez le chemin complet du fichier contenant le certificat du serveur ou la chaîne de certificats, utilisé pour les communications chiffrées entre les composants Zabbix.

TLSCipherAll

Spécifiez la chaîne de priorité GnuTLS ou la chaîne de chiffrement OpenSSL (TLS 1.2). Remplacez les critères de sélection de la suite de chiffrement par défaut pour le chiffrement basé sur des certificats et sur PSK.

Exemple :

```
TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256
```

TLSCipherAll13

Spécifiez la chaîne de chiffrement pour OpenSSL 1.1.1 ou une version plus récente en TLS 1.3. Remplacez les critères de sélection par défaut de la suite de chiffrement pour le chiffrement basé sur les certificats et sur PSK.

Exemple pour GnuTLS :

```
NONE:+VERS-TLS1.2:+ECDHE-RSA:+RSA:+ECDHE-PSK:+PSK:+AES-128-GCM:+AES-128-CBC:+AEAD:+SHA256:+SHA1:+CURVE-ALL
```

Exemple pour OpenSSL :

```
EECDH+aRSA+AES128:RSA+aRSA+AES128:kECDHEPSK+AES128:kPSK+AES128
```

TLSCipherCert

Spécifiez la chaîne de priorité GnuTLS ou la chaîne de chiffrement OpenSSL (TLS 1.2). Remplacez les critères de sélection de la suite de chiffrement par défaut pour le chiffrement basé sur les certificats.

Exemple pour GnuTLS :

```
NONE:+VERS-TLS1.2:+ECDHE-RSA:+RSA:+AES-128-GCM:+AES-128-CBC:+AEAD:+SHA256:+SHA1:+CURVE-ALL:+COMP-NULL:+SIG
```

Exemple pour OpenSSL :

```
EECDH+aRSA+AES128:RSA+aRSA+AES128
```

TLSCipherCert13

Spécifiez la chaîne de chiffrement pour OpenSSL 1.1.1 ou une version plus récente dans TLS 1.3. Remplacez les critères de sélection de la suite de chiffrement par défaut pour le chiffrement basé sur des certificats.

TLSCipherPSK

Spécifiez la chaîne de priorité GnuTLS ou la chaîne de chiffrement OpenSSL (TLS 1.2). Remplacez les critères de sélection par défaut de la suite de chiffrement pour le chiffrement basé sur PSK.

Exemple pour GnuTLS :

```
NONE:+VERS-TLS1.2:+ECDHE-PSK:+PSK:+AES-128-GCM:+AES-128-CBC:+AEAD:+SHA256:+SHA1:+CURVE-ALL:+COMP-NULL:+SIG
```

Exemple pour OpenSSL :

```
kECDHEPSK+AES128:kPSK+AES128
```

TLSCipherPSK13

Spécifiez la chaîne de chiffrement pour OpenSSL 1.1.1 ou une version plus récente en TLS 1.3. Remplacez les critères de sélection par défaut de la suite de chiffrement pour le chiffrement basé sur PSK.

Exemple :

```
TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256
```

TLSCRLFile

Spécifiez le chemin complet du fichier contenant les certificats révoqués. Ce paramètre est utilisé pour les communications chiffrées entre les composants Zabbix.

TLSFrontendAccept

Spécifiez le niveau de chiffrement pour les connexions entrantes depuis l'interface.

Plusieurs valeurs peuvent être spécifiées, séparées par des virgules :

- unencrypted - accepte les connexions sans chiffrement.
- cert - accepte les connexions sécurisées avec TLS et un certificat.

Par défaut : `unencrypted`

TLSFrontendCertIssuer

Spécifiez l'émetteur de certificat autorisé de l'interface.

TLSFrontendCertSubject

Spécifiez le sujet de certificat d'interface autorisé.

TLSKeyFile

Spécifiez le chemin d'accès complet du fichier contenant la clé privée du serveur, utilisée pour les communications chiffrées entre les composants Zabbix.

TLSListen

Exiger des connexions TLS uniquement sur le socket trapper.

Valeurs prises en charge :

- required - accepter uniquement les connexions TLS

TmpDir

Spécifiez le répertoire temporaire.

Par défaut: `/tmp`

TrapperTimeout

Spécifiez le délai d'attente du trapper en secondes pour :

- la récupération des données historiques depuis le proxy Zabbix
- l'envoi des données de configuration au proxy Zabbix
- l'exécution d'un script global ou l'exécution d'une commande distante sur le serveur Zabbix

Par défaut: 300
 Plage: 1-300

TrendCacheSize

Définissez la taille du cache des tendances, en octets.
La taille de la mémoire partagée utilisée pour stocker les tendances.

Par défaut: 4M
 Plage: 128K-16G

TrendFunctionCacheSize

Définissez la taille du cache des fonctions de tendance, en octets.
La taille de la mémoire partagée utilisée pour mettre en cache les données calculées des fonctions de tendance.

Par défaut: 4M
 Plage: 128K-2G

UnavailableDelay

Définissez la fréquence de vérification de la disponibilité de l'hôte pendant la période d'indisponibilité en secondes.

Par défaut: 60
 Plage: 1-3600

UnreachableDelay

Définit la fréquence de vérification de la disponibilité de l'hôte pendant la période d'inaccessibilité, en secondes.

Par défaut: 15
 Plage: 1-3600

UnreachablePeriod

Définit après combien de secondes d'inaccessibilité considérer l'hôte comme indisponible.

Par défaut: 45
 Plage: 1-3600

Utilisateur

Abaisser les privilèges à un utilisateur spécifique existant sur le système.
Ne prend effet que si l'exécution est effectuée en tant que 'root' et si AllowRoot est désactivé.

Par défaut: `zabbix`

ValueCacheSize

Définissez la taille du cache des valeurs d'historique (en octets).
 Taille de la mémoire partagée utilisée pour mettre en cache les demandes de données d'historique des éléments.
 La valeur 0 désactive le cache des valeurs (non recommandé).
 Lorsque le cache des valeurs n'a plus de mémoire partagée disponible, un message d'avertissement est écrit dans le journal du serveur toutes les 5 minutes.

Valeur par défaut : `8M`
 Plage : 0, 128K-64G

Vault

Spécifiez le fournisseur de coffre-fort : - *HashiCorp* - HashiCorp KV Secrets Engine version 2 ; - *CyberArk* - CyberArk Central Credential Provider
Doit correspondre au fournisseur de coffre-fort défini dans l'interface.

Par défaut : `HashiCorp`

VaultAppRoleID

Spécifiez l'ID AppRole du coffre HashiCorp.
 Si défini, active l'authentification AppRole. Ne peut pas être utilisé avec VaultToken.
 Pris en charge uniquement pour le coffre HashiCorp.

VaultAppSecretID

Spécifiez l'ID secret AppRole du coffre HashiCorp.
 Pris en charge uniquement pour le coffre HashiCorp.

VaultDBPath

Spécifie l'emplacement permettant de récupérer les identifiants de base de données par clés. Il s'agit d'un chemin ou d'une requête Vault, selon le Vault.

Les clés utilisées pour HashiCorp sont `password` et `username`.

Exemple de chemin avec `VaultPrefix=/v1/secret/data/zabbix/` :

```
database
```

Exemple de chemin sans VaultPrefix :

```
secret/zabbix/database
```

Les clés utilisées pour CyberArk sont `Content` et `UserName`.

Exemple :

```
AppID=zabbix_server&Query=Safe=passwordSafe;Object=zabbix_server_database
```

Cette option ne peut être utilisée que si `DBUser` et `DBPassword` ne sont pas spécifiés.

VaultPrefix

Spécifiez un préfixe personnalisé pour le chemin ou la requête du coffre, selon le Vault. Les valeurs par défaut les plus appropriées seront utilisées si ce paramètre n'est pas spécifié.
 Notez que `data` est automatiquement ajouté après le point de montage pour HashiCorp si `VaultPrefix` n'est pas spécifié.

Exemple de préfixe pour HashiCorp :

```
/v1/secret/data/zabbix/
```

Exemple de préfixe pour Cyberark :

```
/AIMWebService/api/Accounts?
```

VaultTLSCertFile

Spécifiez le nom du fichier de certificat SSL utilisé pour l'authentification du client. Le fichier de certificat doit être au format PEM1.
 Si le fichier de certificat contient également la clé privée, laissez le champ du fichier de clé SSL vide.
 Le répertoire contenant ce fichier est spécifié par le paramètre de configuration SSLCertLocation.
 Cette option peut être omise, mais elle est recommandée pour le coffre CyberArkCCP.

VaultTLSKeyFile

Spécifiez le nom du fichier de clé privée SSL utilisé pour l'authentification du client. Le fichier de clé privée doit être au format PEM1.
Le répertoire contenant ce fichier est spécifié par le paramètre de configuration SSLKeyLocation.
Cette option peut être omise, mais elle est recommandée pour le coffre CyberArkCCP.

VaultToken

Spécifiez le jeton d'authentification HashiCorp Vault. Le jeton d'authentification HashiCorp Vault doit avoir été généré exclusivement pour le serveur Zabbix, avec une autorisation en lecture seule sur les chemins spécifiés dans Vault macros et une autorisation en lecture seule sur le chemin spécifié dans le paramètre de configuration facultatif VaultDBPath.
 Il s'agit d'une erreur si `VaultToken` et la variable d'environnement `VAULT_TOKEN` sont définis en même temps.
 Ne peut pas être utilisé avec VaultAppRoleID.
 Pris en charge uniquement pour HashiCorp Vault.

VaultURL

Spécifiez l'URL du serveur vault. Le répertoire des certificats CA à l'échelle du système sera utilisé si SSLCALocation n'est pas spécifié.

Par défaut: `https://127.0.0.1:8200`

VMwareCacheSize

Définissez la taille de la mémoire partagée pour stocker les données VMware. Une vérification interne VMware `zabbix[vmware,buffer,...]` peut être utilisée pour surveiller l'utilisation du cache VMware (voir Vérifications internes).
Notez que la mémoire partagée n'est pas allouée s'il n'existe aucune instance de `vmware collector` configurée pour démarrer.

Valeur par défaut: 8M
 Plage: 256K-2G

VMwareFrequency

Définissez la fréquence de collecte des données à partir d'un seul service VMware (en secondes). Cette fréquence doit être définie sur l'intervalle de mise à jour le plus court de tout élément de surveillance VMware.

Par défaut: 60
 Plage: 10-86400

VMwarePerfFrequency

Définissez la fréquence de récupération des statistiques des compteurs de performance d'un seul service VMware (en secondes).
Cette fréquence doit être définie sur l'intervalle de mise à jour le plus court de tout élément de surveillance VMware qui utilise les compteurs de performance VMware.

Par défaut: 60
 Plage: 10-86400

VMwareTimeout

Définit le nombre maximal de secondes pendant lesquelles un `vmware collector` attendra une réponse du service VMware (vCenter ou hyperviseur ESX).

Par défaut: 10
 Plage: 1-300

WebDriverURL

Spécifiez l'URL de l'interface WebDriver.

Exemple (utilisé avec le serveur autonome Selenium WebDriver) :

```
WebDriverURL=http://localhost:4444
```

WebServiceURL

Spécifiez l'URL du service web au format `http[s]://host:port/report`.

Exemple :

```
WebServiceURL=http://localhost:10053/report
```

Le schéma `http://` peut être omis uniquement pour les connexions non TLS (HTTP) ; avec TLS configuré, `https://` doit être utilisé.

Notes de bas de page

[1] Notez qu'un trop grand nombre de processus de collecte de données (pollers, unreachable pollers, ODBC pollers, HTTP pollers, Java pollers, pingers, trappers, proxypollers), ainsi que le gestionnaire IPMI, le SNMP trapper, les workers de prétraitement et les workers de découverte, peut épuiser la limite de descripteurs de fichiers par processus pour le gestionnaire de prétraitement.

> **Warning:**
> L'épuisement de la limite de descripteurs de fichiers entraînera l'arrêt du serveur Zabbix, généralement peu après le démarrage, mais parfois après un délai plus long. Pour éviter de tels problèmes, consultez le fichier de configuration du serveur Zabbix afin d'optimiser le nombre de vérifications et de processus simultanés. De plus, si nécessaire, assurez-vous que la limite de descripteurs de fichiers est suffisamment élevée en vérifiant et en ajustant les limites du système.

proxy

Vue d'ensemble Le proxy Zabbix est un processus qui peut collecter les données de supervision d'un ou plusieurs dispositifs supervisés et envoyer ces informations au serveur Zabbix, agissant essentiellement au nom du serveur. Toutes les données collectées sont mises en mémoire tampon localement, puis transférées au serveur Zabbix auquel le proxy appartient.

Le déploiement d'un proxy est facultatif, mais peut être très utile pour répartir la charge d'un seul serveur Zabbix. Si seuls les proxies collectent les données, le traitement sur le serveur nécessite moins de ressources CPU et d'entrées/sorties disque.

Un proxy Zabbix est la solution idéale pour la supervision centralisée de sites distants, de succursales et de réseaux ne disposant pas d'administrateurs locaux.

Un proxy Zabbix nécessite une base de données distincte.

> **Attention:**
> Notez que les bases de données prises en charge par le proxy Zabbix sont SQLite, MySQL et PostgreSQL.

Voir également :

- Supervision distribuée à l'aide de proxies
- Équilibrage de charge et haute disponibilité des proxies
- Synchronisation de la configuration de supervision
- Protocole d'échange de données entre le serveur et le proxy
- Paramètres de configuration : proxy Zabbix
- Configurer un proxy Zabbix Cloud

Exécution du proxy Si installé en tant que paquet

Zabbix proxy s'exécute comme un processus démon. Le proxy peut être démarré en exécutant :

```
systemctl start zabbix-proxy
```

Cela fonctionnera sur la plupart des systèmes GNU/Linux. Sur d'autres systèmes, vous devrez peut-être exécuter :

```
/etc/init.d/zabbix-proxy start
```

De même, pour arrêter/redémarrer/afficher l'état de Zabbix proxy, utilisez les commandes suivantes :

```
systemctl stop zabbix-proxy
systemctl restart zabbix-proxy
systemctl status zabbix-proxy
```

Démarrer manuellement

Si ce qui précède ne fonctionne pas, vous devez le démarrer manuellement. Trouvez le chemin vers le binaire `zabbix_proxy` et exécutez :

```
zabbix_proxy
```

Vous pouvez utiliser les paramètres de ligne de commande suivants avec le proxy Zabbix :

Parameter	Description
`-c --config <config-file>`	Chemin vers le fichier de configuration.
`-f --foreground`	Exécuter le proxy Zabbix au premier plan.
`-R --runtime-control <option>`	Effectuer des fonctions administratives.
`-T --test-config`	Valider le fichier de configuration et quitter.
`-h --help`	Afficher les informations d'aide et quitter.
`-V --version`	Afficher le numéro de version.

Exemples d'exécution du proxy Zabbix avec des paramètres de ligne de commande :

```
zabbix_proxy -c /usr/local/etc/zabbix_proxy.conf
zabbix_proxy --help
zabbix_proxy -V
```

Contrôle à chaud

Options de contrôle à chaud :

Option	Description	Cible
`config_cache_reload`	Recharge le cache de configuration. Ignoré si le cache est en cours de chargement. Le proxy Zabbix actif se connectera au serveur Zabbix et demandera les données de configuration. Le proxy Zabbix passif demandera les données de configuration au serveur Zabbix lors de la prochaine connexion du serveur au proxy.	
`history_cache_clear[<target>]`	Vider le cache d'historique pour l'élément spécifié par son ID. Affecte toutes les valeurs de l'élément, à l'exception de la première et de la dernière valeur.	**target** - ID de l'élément.
`diaginfo[=<section>]`	Collecte des informations de diagnostic dans le fichier journal du proxy.	`historycache` - statistiques du cache d'historique; `preprocessing` - statistiques du gestionnaire de prétraitement; `locks` - liste des mutex (vide sur les systèmes *BSD*).
`snmp_cache_reload`	Recharge le cache SNMP - efface les propriétés du moteur SNMP (engine time, engine boots, engine id, credentials) pour tous les hôtes. À utiliser pour forcer un effacement global du cache lors du dépannage des problèmes SNMP.	
`housekeeper_execute`	Démarre la procédure de housekeeping. Ignoré si la procédure de housekeeping est actuellement en cours.	
`log_level_increase[=<target>]`	Augmente le niveau de journalisation, affecte tous les processus si aucune cible n'est spécifiée. Non pris en charge sur les systèmes *BSD*.	**process type** - tous les processus du type spécifié (par exemple, `poller`). Voir tous les types de processus du proxy. **process type,N** - type de processus et numéro (par exemple, `poller,3`) **pid** - identifiant de processus (1 à 65535). Pour des valeurs plus élevées, spécifiez la cible sous la forme 'process type,N'.
`log_level_decrease[=<target>]`	Diminue le niveau de journalisation, affecte tous les processus si aucune cible n'est spécifiée. Non pris en charge sur les systèmes *BSD*.	

Option	Description	Cible
`prof_enable[=<target>]`	Activer le profilage. Affecte tous les processus si aucune cible n'est spécifiée. Le profilage activé fournit des détails sur tous les rwlocks/mutexes par nom de fonction.	**process type** - tous les processus du type spécifié (par exemple, `history syncer`). Voir tous les types de processus du proxy. **process type,N** - type de processus et numéro (par exemple, `history syncer,1`). **pid** - identifiant de processus (1 à 65535). Pour des valeurs plus élevées, spécifiez la cible sous la forme 'process type,N'. **scope** - `rwlock`, `mutex`, `processing` peuvent être utilisés avec le type de processus et le numéro (par exemple, `history syncer,1,processing`) ou avec tous les processus d'un type (par exemple, `history syncer,rwlock`).
`prof_disable[=<target>]`	Désactiver le profilage. Affecte tous les processus si aucune cible n'est spécifiée.	**process type** - tous les processus du type spécifié (par exemple, `history syncer`). Voir tous les types de processus du proxy. **process type,N** - type de processus et numéro (par exemple, `history syncer,1`). **pid** - identifiant de processus (1 à 65535). Pour des valeurs plus élevées, spécifiez la cible sous la forme 'process type,N'.

Lorsqu'un type de processus est spécifié comme cible pour le contrôle à chaud, la commande s'applique uniquement aux processus en cours d'exécution de ce type. Si aucun processus du type spécifié n'est en cours d'exécution, la commande renvoie une erreur :

```
zabbix_proxy -R log_level_increase='data sender'
Cannot redirect signal: "data sender" process does not exist
```

Exemple d'utilisation du contrôle à chaud pour recharger le cache de configuration du proxy :

```
zabbix_proxy -c /usr/local/etc/zabbix_proxy.conf -R config_cache_reload
```

Exemple d'utilisation du contrôle à chaud pour vider le cache d'historique d'un élément :

```
zabbix_proxy -c /usr/local/etc/zabbix_proxy.conf -R history_cache_clear=42243
```

Exemples d'utilisation du contrôle à chaud pour collecter des informations de diagnostic :

```
### Collecter toutes les informations de diagnostic disponibles dans le fichier journal du proxy :
zabbix_proxy -R diaginfo

### Collecter les statistiques du cache d'historique dans le fichier journal du proxy :
zabbix_proxy -R diaginfo=historycache
```

Exemple d'utilisation du contrôle à chaud pour recharger le cache SNMP :

```
zabbix_proxy -R snmp_cache_reload
```

Attention:
Lorsqu'une interface SNMPv3 est mise à jour via l'interface Zabbix, Zabbix recharge automatiquement les nouveaux identifiants SNMPv3 pour cette interface dans la plupart des cas ; utilisez `-R snmp_cache_reload` uniquement si la collecte échoue encore après un changement d'identifiants (par exemple, en raison d'incohérences engineBoots/engineID ou de périphériques non conformes RFC), ou lorsque vous devez forcer un effacement global du cache SNMP pour le dépannage.

Exemple d'utilisation du contrôle à chaud pour déclencher l'exécution du housekeeper :

```
zabbix_proxy -c /usr/local/etc/zabbix_proxy.conf -R housekeeper_execute
```

Exemples d'utilisation du contrôle à chaud pour modifier le niveau de journalisation :

```
### Augmenter le niveau de journalisation de tous les processus :
zabbix_proxy -c /usr/local/etc/zabbix_proxy.conf -R log_level_increase
```

```
### Augmenter le niveau de journalisation du deuxième processus poller :
zabbix_proxy -c /usr/local/etc/zabbix_proxy.conf -R log_level_increase=poller,2

### Augmenter le niveau de journalisation du processus avec le PID 1234 :
zabbix_proxy -c /usr/local/etc/zabbix_proxy.conf -R log_level_increase=1234

### Diminuer le niveau de journalisation de tous les processus http poller :
zabbix_proxy -c /usr/local/etc/zabbix_proxy.conf -R log_level_decrease="http poller"
```

Exécuter le proxy en tant qu'utilisateur

Le proxy Zabbix est conçu pour s'exécuter en tant qu'utilisateur non root. Il s'exécutera avec l'utilisateur non root sous lequel il a été démarré. Vous pouvez donc exécuter le proxy avec n'importe quel utilisateur non root sans problème.

Si vous essayez de l'exécuter en tant que « root », il basculera vers un utilisateur « zabbix » codé en dur, qui doit être présent sur votre système. Vous ne pouvez exécuter le proxy en tant que « root » que si vous modifiez en conséquence le paramètre `AllowRoot` dans le fichier de configuration du proxy.

Fichier de configuration

Le chemin par défaut du fichier de configuration dépend du système d'exploitation et de la méthode d'installation de Zabbix :

- Pour une installation à partir des sources : `/usr/local/etc/zabbix_proxy.conf`, ou tel que défini par les variables de compilation `--sysconfdir` ou `--prefix`.
- Pour une installation à partir de paquets : `/etc/zabbix/zabbix_proxy.conf`.

Consultez les options du fichier de configuration pour plus de détails sur la configuration de `zabbix_proxy`.

Types de processus et threads du proxy

- `agent poller` - processus de sondage asynchrone pour les vérifications passives avec un thread de travail ;
- `apm manager` - processus gestionnaire pour la surveillance des performances des applications (APM) ;
- `apm worker` - processus chargé de traiter les tâches de surveillance des performances des applications (APM) provenant de `apm manager` ;
- `availability manager` - processus chargé de mettre à jour la disponibilité des hôtes ;
- `browser poller` - processus de sondage pour les vérifications des éléments de type navigateur ;
- `configuration syncer` - processus chargé de gérer le cache en mémoire des données de configuration ;
- `data sender` - processus d'envoi des données du proxy ;
- `discovery manager` - processus gestionnaire chargé de la découverte des périphériques ;
- `discovery worker` - processus chargé de traiter les tâches de découverte provenant de `discovery manager` ;
- `history syncer` - processus d'écriture dans la base de données d'historique ;
- `housekeeper` - processus chargé de supprimer l'historique obsolète des éléments ;
- `http agent poller` - processus de sondage asynchrone pour les vérifications HTTP avec un thread de travail ;
- `http poller` - processus de sondage de la surveillance web ;
- `icmp pinger` - processus de sondage pour les vérifications icmpping ;
- `internal poller` - processus de sondage pour les vérifications internes ;
- `ipmi manager` - gestionnaire des processus de sondage IPMI ;
- `ipmi poller` - processus de sondage pour les vérifications IPMI ;
- `java poller` - processus de sondage pour les vérifications Java ;
- `odbc poller` - processus de sondage pour les vérifications ODBC ;
- `poller` - processus de sondage normal pour les vérifications passives ;
- `preprocessing manager` - gestionnaire des tâches de prétraitement avec des threads de travail de prétraitement ;
- `preprocessing worker` - thread chargé du prétraitement des données ;
- `self-monitoring` - processus chargé de collecter les statistiques internes du serveur ;
- `snmp poller` - processus de sondage asynchrone pour les vérifications SNMP avec un thread de travail (uniquement les éléments `walk[OID]` et `get[OID]`) ;
- `snmp trapper` - processus trapper pour les traps SNMP ;
- `supervisor` - coordonne le démarrage et l'arrêt des processus et threads du serveur, et surveille leur état de préparation ;
- `task manager` - processus chargé de l'exécution à distance des tâches demandées par d'autres composants (par exemple, fermer un problème, confirmer un problème, vérifier immédiatement la valeur d'un élément, fonctionnalité de commande à distance) ;
- `telemetry query poller` - processus de sondage pour les vérifications des éléments de type requête de télémétrie ;
- `trapper` - processus trapper pour les vérifications actives, les traps et la communication avec le proxy ;
- `unreachable poller` - processus de sondage pour les périphériques inaccessibles ;
- `vmware collector` - collecteur de données VMware chargé de recueillir les données des services VMware.

Le fichier journal du proxy peut être utilisé pour observer ces types de processus.

Le fichier journal du proxy est créé avec des autorisations de lecture-écriture uniquement pour son propriétaire. De plus, le fichier est lisible par le groupe propriétaire. Toutes les autres autorisations sont refusées.

Différents types de processus du proxy Zabbix peuvent être surveillés à l'aide de l'élément interne `zabbix[process,<type>,<mode>,<sta`

Statistiques des transactions du synchroniseur d'historique

Le titre du processus du synchroniseur d'historique affiche des statistiques détaillées sur les transactions du synchroniseur d'historique.

```
205276 ?        S      0:00  zabbix_proxy: history syncer #1 [processed 1 values in 0.001179 (0.001167,0.0
205277 ?        S      0:00  zabbix_proxy: history syncer #2 [processed 0 values in 0.000022 (0.000000,0.0
```

Les durées, dans `processed...in N (<timings>) sec`, sont :

- Temps passé à écrire les valeurs des éléments dans la base de données.
- Temps passé à mettre à jour les données de l'élément (état, erreurs).

Procédure de housekeeping

Le proxy Zabbix dispose du processus `housekeeper` qui supprime l'historique et les tendances obsolètes des éléments. Il s'exécute par cycles, avec une fréquence déterminée par `HousekeepingFrequency`, et la limite de suppression par cycle déterminée par `ProxyLocalBuffer` et `ProxyOfflineBuffer`. Contrairement à la procédure de housekeeping du serveur Zabbix, le processus `housekeeper` du proxy n'utilise pas la table `housekeeper` — il supprime toutes les données obsolètes une fois par cycle de housekeeping.

Plates-formes prises en charge Le proxy Zabbix fonctionne sur la même liste de plates-formes prises en charge que le serveur Zabbix.

Tampon mémoire Le tampon mémoire permet de stocker de nouvelles données (valeurs d'éléments, découverte réseau, enregistrement automatique des hôtes) dans le tampon et de les envoyer au serveur Zabbix sans accéder à la base de données. Le tampon mémoire a été introduit pour le proxy dans Zabbix 7.0.

Dans les installations antérieures à Zabbix 7.0, les données collectées étaient stockées dans la base de données avant d'être envoyées au serveur Zabbix. Pour ces installations, ce comportement reste celui par défaut après la mise à niveau vers Zabbix 7.0.

Pour optimiser les performances, il est recommandé de configurer l'utilisation du tampon mémoire sur le proxy. Cela est possible en modifiant la valeur de `ProxyBufferMode` de `disk` (valeur par défaut codée en dur pour les installations existantes) à `hybrid` (recommandé) ou `memory`. Il est également nécessaire de définir la taille du tampon mémoire (paramètre `ProxyMemoryBufferSize`).

En mode hybride, le tampon est protégé contre la perte de données en vidant les données non envoyées dans la base de données si le proxy est arrêté, si le tampon est plein ou si les données sont trop anciennes. Lorsque toutes les valeurs ont été vidées dans la base de données, le proxy recommence à utiliser le tampon mémoire.

En mode mémoire, le tampon mémoire est utilisé, mais aucune protection contre la perte de données n'est assurée. Si le proxy est arrêté ou si la mémoire est saturée, les données non envoyées sont supprimées.

Le mode hybride (`ProxyBufferMode=hybrid`) est appliqué à toutes les nouvelles installations depuis Zabbix 7.0.

Des paramètres supplémentaires tels que `ProxyMemoryBufferSize` et `ProxyMemoryBufferAge` définissent respectivement la taille du tampon mémoire et l'ancienneté maximale des données dans le tampon.

Notez qu'en cas de configuration contradictoire, le proxy affiche une erreur et ne démarre pas, par exemple si :

- `ProxyBufferMode` est défini sur `hybrid` ou `memory` et que `ProxyMemoryBufferSize` vaut 0.
- `ProxyBufferMode` est défini sur `hybrid` ou `memory` et que `ProxyLocalBuffer` n'est pas égal à 0.

Locale Notez que le proxy nécessite une locale UTF-8 afin que certains éléments textuels puissent être interprétés correctement. La plupart des systèmes Unix modernes ont une locale UTF-8 par défaut, cependant, sur certains systèmes, il peut être nécessaire de la définir explicitement.

Calcul des files d'attente pendant la maintenance

> **Attention:**
> Le proxy Zabbix n'est pas informé des périodes de maintenance ; voir Calcul d
> pour plus de détails.

Utilisation de l'interface Zabbix derrière un proxy Si l'interface Zabbix s'exécute derrière un serveur proxy, le chemin du cookie dans le fichier de configuration du proxy doit être réécrit afin de correspondre au chemin utilisé par le proxy inverse. Voir les exemples ci-dessous. Si le chemin du cookie n'est pas réécrit, les utilisateurs peuvent rencontrer des problèmes d'autorisation lorsqu'ils tentent de se connecter à l'interface Zabbix.

Exemple de configuration pour NGINX

```
### ..
location / {
### ..
proxy_cookie_path /zabbix /;
proxy_pass http://192.168.0.94/zabbix/;
### ..
```

Exemple de configuration pour Apache

```
### ..
ProxyPass "/" http://host/zabbix/
ProxyPassReverse "/" http://host/zabbix/
ProxyPassReverseCookiePath /zabbix /
ProxyPassReverseCookieDomain host zabbix.example.com
### ..
```

Surveillance distribuée à l'aide de proxies

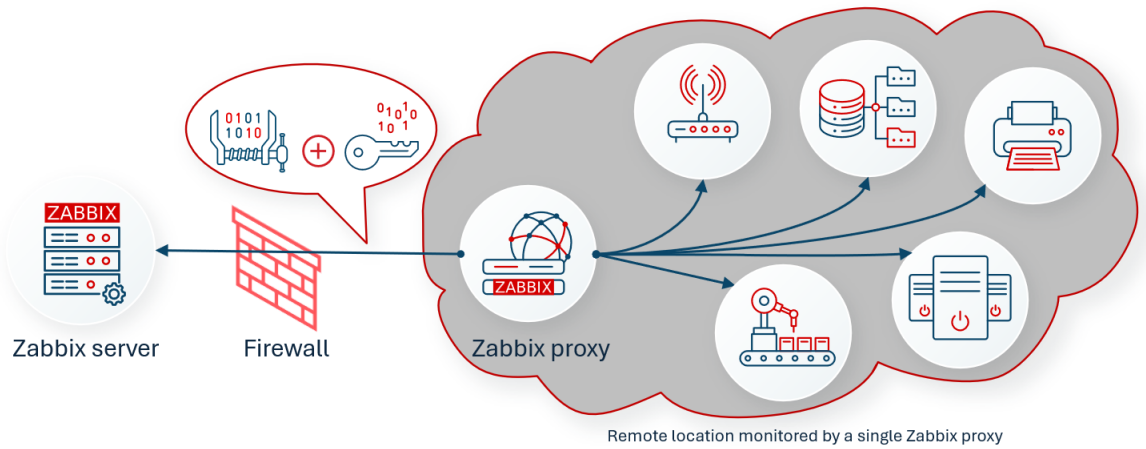
Vue d'ensemble

Les proxy Zabbix peuvent être utilisés pour créer une configuration de supervision distribuée efficace et fiable.

Les proxy sont déployés localement afin de collecter les données de performance et de disponibilité. Les données sont ensuite transmises à un serveur centralisé.

Un proxy Zabbix peut être utilisé pour :

- Superviser des sites distants
- Superviser des sites dont les communications sont peu fiables
- Décharger le serveur Zabbix lors de la supervision de milliers d'appareils
- Collecter des données de télémétrie pour la supervision des performances des applications (APM)



Le proxy nécessite une seule connexion TCP au serveur Zabbix. Il est ainsi plus facile de contourner un pare-feu, car il suffit de configurer une seule règle de pare-feu.

> **Attention:**
> Le proxy Zabbix doit utiliser une base de données distincte. Le pointer vers la base de données du serveur Zabbix entraînera une corruption de la configuration.

Toutes les données collectées par le proxy sont stockées localement avant d'être transmises au serveur. Ainsi, aucune donnée n'est perdue en raison de problèmes temporaires de communication avec le serveur. Les paramètres `ProxyLocalBuffer` et `ProxyOfflineBuffer` du fichier de configuration du proxy contrôlent la durée pendant laquelle les données sont conservées localement.

Attention:
Il peut arriver qu'un proxy, qui reçoit directement les dernières modifications de configuration depuis la base de données du serveur Zabbix, dispose d'une configuration plus à jour que le serveur Zabbix, dont la configuration peut ne pas être actualisée aussi rapidement en raison de la valeur de CacheUpdateFrequency. Par conséquent, le proxy peut commencer à collecter des données et les envoyer au serveur Zabbix, qui ignore ces données.

Le proxy Zabbix est un collecteur de données. Il ne calcule pas les déclencheurs, ne traite pas les événements et n'envoie pas d'alertes. Pour obtenir une vue d'ensemble des fonctionnalités du proxy, consultez le tableau suivant :

Fonction		Prise en charge par le proxy
Éléments		
	Vérifications par Zabbix agent	**Oui**
	Vérifications par Zabbix agent (actives)	**Oui** [1]
	Vérifications simples	**Oui**
	Éléments trapper	**Oui**
	Vérifications SNMP	**Oui**
	Traps SNMP	**Oui**
	Vérifications IPMI	**Oui**
	Vérifications JMX	**Oui**
	Supervision des fichiers journaux	**Oui**
	Vérifications internes	**Oui**
	Vérifications SSH	**Oui**
	Vérifications Telnet	**Oui**
	Vérifications externes	**Oui**
	Éléments dépendants	**Oui**
	Éléments de type script	**Oui**
	Éléments de type navigateur	**Oui**
	Éléments de requête de télémétrie	**Oui**
Supervision web intégrée		**Oui**
Prétraitement des valeurs des éléments		**Oui**
Découverte réseau		**Oui**
Enregistrement automatique des agents actifs		**Oui**
Découverte de bas niveau		**Oui** [2]
Commandes à distance		**Oui**
Collecte de données de télémétrie (APM)		**Oui** [3]
Calcul des déclencheurs		*Non*
Traitement des événements		*Non*
Corrélation des événements		*Non*
Envoi d'alertes		*Non*

Note:
[1] Pour s'assurer qu'un agent demande les vérifications actives au proxy (et non au serveur), le proxy doit être indiqué dans le paramètre **ServerActive** du fichier de configuration de l'agent.
 [2] Pour la découverte de bas niveau, le proxy Zabbix collecte et prétraite uniquement les données, puis les envoie au serveur Zabbix pour un traitement ultérieur.
 [3] Les données de télémétrie sont collectées uniquement si l'APM est activé sur le proxy (voir l'onglet *APM* dans la configuration du proxy) ; le serveur Zabbix ne peut pas recevoir de données de télémétrie.

Fonctionnalités du proxy

Lors du choix d'utiliser ou non un proxy, plusieurs considérations doivent être prises en compte.

	Proxy
Léger	**Oui**
GUI	Non
Fonctionne de manière autonome	**Oui**
Maintenance facile	**Oui**
Création automatique de la base de données	**Oui**[1]
Administration locale	Non
Prêt pour le matériel embarqué	**Oui**
Connexions TCP à sens unique	**Oui**

	Proxy
Configuration centralisée	**Oui**
Génère des notifications	Non

[1] La fonctionnalité de création automatique de la base de données fonctionne uniquement avec SQLite. Les autres bases de données prises en charge nécessitent une configuration manuelle.

Attention:
Le proxy Zabbix n'a pas connaissance des périodes de maintenance ; voir Calcul des files d'attente pendant la maintenance pour plus de détails.

Protection contre la surcharge

Si le serveur Zabbix a été arrêté pendant un certain temps, et que des proxies ont collecté beaucoup de données, puis que le serveur démarre, il peut être surchargé (l'utilisation du cache d'historique reste à 95-100 % pendant un certain temps). Cette surcharge peut entraîner une baisse des performances, les vérifications étant traitées plus lentement qu'elles ne le devraient. Une protection contre ce scénario a été mise en place afin d'éviter les problèmes liés à la surcharge du cache d'historique.

Lorsque le cache d'historique du serveur Zabbix est plein, l'accès en écriture au cache d'historique est limité, ce qui bloque les processus de collecte de données du serveur. Le cas le plus courant de surcharge du cache d'historique se produit après une indisponibilité du serveur, lorsque les proxies téléversent les données collectées. Pour éviter cela, une limitation du proxy a été ajoutée (actuellement, elle ne peut pas être désactivée).

Lorsque l'utilisation du cache d'historique atteint 80 %, le serveur Zabbix passe en mode de limitation. En mode de limitation, le serveur n'accepte les données des proxies que lorsque l'utilisation du cache d'historique est inférieure à 60 %, en faisant tourner les proxies acceptés. Une fois que l'utilisation du cache d'historique descend sous 20 %, le serveur revient en mode normal.

De plus, en mode normal, le serveur Zabbix limite les proxies individuels qui envoient des paquets très volumineux (plus de 10 000 enregistrements) si l'utilisation du cache d'historique dépasse 60 %. Cette décision est prise au moment où le serveur évalue un téléversement de proxy et peut donc ne pas toujours apparaître immédiatement dans les graphiques d'utilisation du cache d'historique (l'élément interne `zabbix[wcache,history,pused]` et son intervalle de mise à jour peuvent manquer de courtes pointes).

Ce mode de limitation restera actif jusqu'à ce que l'utilisation du cache atteigne à nouveau 80 %, ou redescende à 20 %, ou que la liste de limitation soit vide. Dans le premier cas, le serveur cessera à nouveau d'accepter les données des proxies. Dans les deux autres cas, le serveur recommencera à fonctionner normalement, en acceptant les données de tous les proxies.

Les informations ci-dessus peuvent être illustrées dans le tableau suivant :

Utilisation du cache d'écriture de l'historique	Mode du serveur Zabbix	Action du serveur Zabbix
Atteint 80 %	Attente	Cesse d'accepter les données des proxies, mais conserve une *liste de limitation* (liste priorisée des proxies à contacter ultérieurement).
Atteint 60 %	Normal, mais prêt à être limité	Peut refuser les téléversements de proxy très volumineux (plus de 10 000 enregistrements) lors de la décision d'acceptation des données ; continue d'accepter les autres données de proxy.
Descend à 20 %	Normal	Supprime la liste de limitation et recommence à accepter normalement les données des proxies.

Vous pouvez utiliser l'élément interne zabbix[wcache,history,pused] pour corréler ce comportement du serveur Zabbix avec une métrique.

Configuration

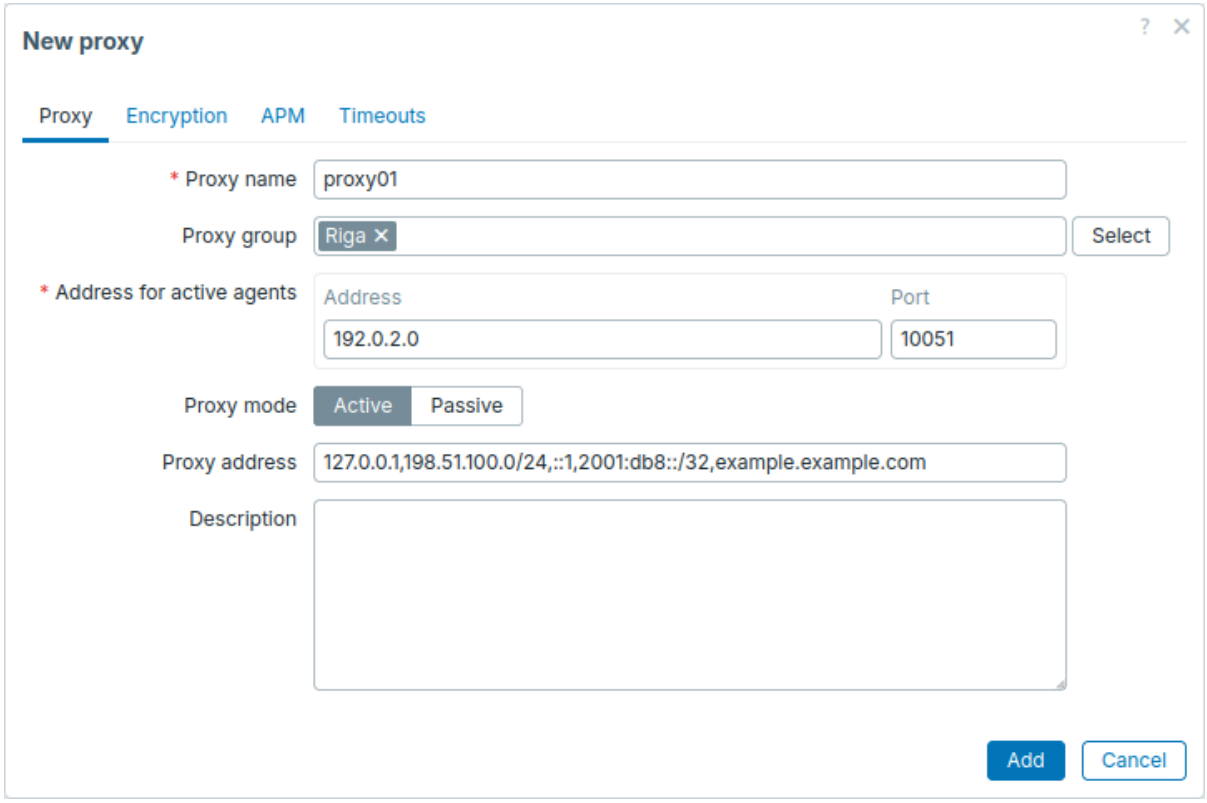
Une fois que vous avez installé et configuré un proxy, il est temps de le configurer dans l'interface Zabbix.

Ajout de proxies

Note:
Au lieu d'installer un proxy sur votre propre infrastructure, vous pouvez en déployer un dans Zabbix Cloud. Pour ce faire, cliquez sur *Create proxy in Cloud* dans *Administration* > *Proxies* ; Zabbix Cloud s'ouvrira dans un nouvel onglet du navigateur. Après avoir créé un nœud proxy, configurez le proxy dans l'interface Zabbix comme décrit ci-dessous. Pour plus d'informations, consultez Configurer un proxy Zabbix Cloud.

Pour configurer un proxy dans l'interface Zabbix :

1. Accédez à *Administration* > *Proxies*.
2. Cliquez sur *Create proxy*.
3. Saisissez les paramètres du proxy dans le formulaire.



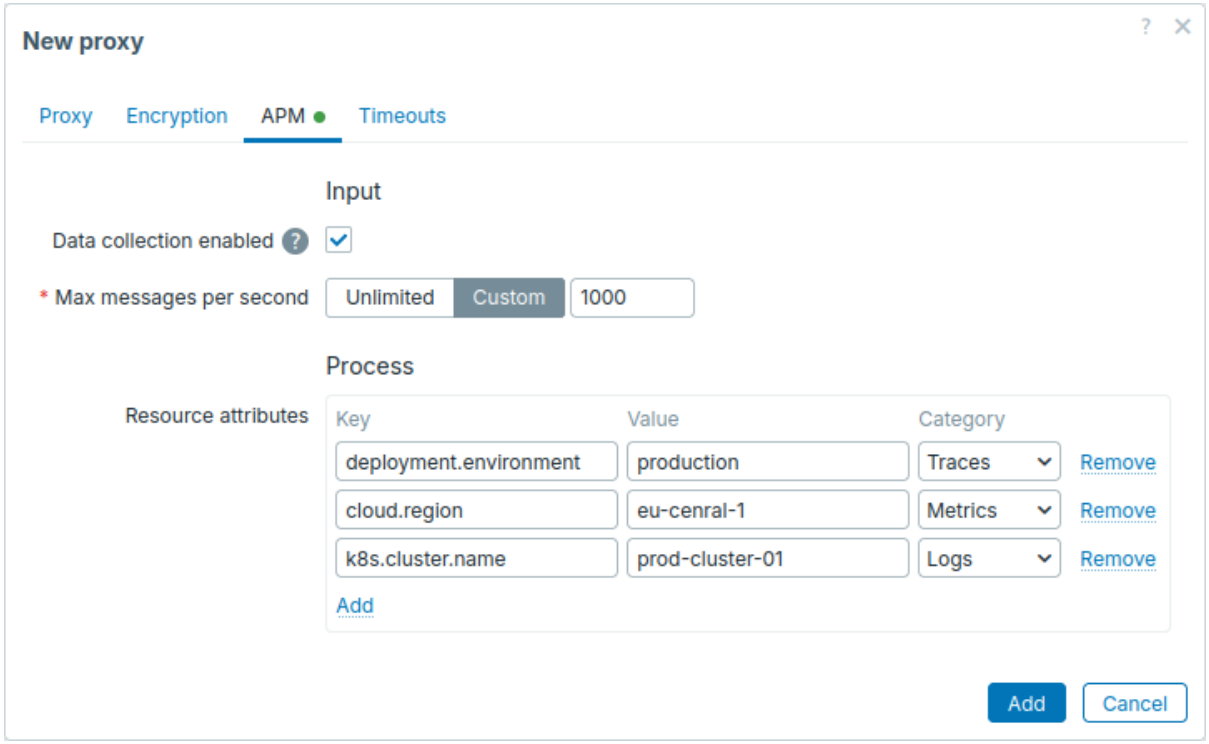
Parameter	Description
Proxy name	Saisissez le nom du proxy. Il doit être identique au nom indiqué dans le paramètre *Hostname* du fichier de configuration du proxy.
Proxy group	Sélectionnez un groupe de proxies pour l'équilibrage de charge/la haute disponibilité des proxies.
Address for active agents	Saisissez l'adresse à laquelle les agents actifs ou les émetteurs surveillés doivent se connecter. Pris en charge **uniquement** par les agents Zabbix 7.0 ou versions ultérieures. Cette adresse est utilisée pour se connecter aux proxies actifs et passifs. Ce champ est disponible uniquement si un groupe de proxies est sélectionné dans le champ *Proxy group*.
Address	Adresse IP/nom DNS auquel se connecter.
Port	Numéro de port TCP (10051 par défaut) auquel se connecter. Les macros utilisateur sont prises en charge.
Proxy mode	Sélectionnez le mode du proxy. **Active** - le proxy se connecte au serveur Zabbix et demande les données de configuration **Passive** - le serveur Zabbix se connecte au proxy *Notez* qu'en l'absence de communications chiffrées (sensibles), les données de configuration du proxy peuvent devenir accessibles à des personnes ayant accès au port trapper du serveur Zabbix lors de l'utilisation d'un proxy actif. Cela est possible, car n'importe qui peut se faire passer pour un proxy actif et demander les données de configuration si aucune authentification n'a lieu ou si les adresses des proxies ne sont pas limitées dans le champ *Proxy address*.
Proxy address	Si ce champ est renseigné, les demandes d'un proxy actif ne sont acceptées qu'à partir de cette liste d'adresses IP séparées par des virgules, éventuellement au format CIDR, ou de noms DNS du proxy Zabbix actif. Ce champ est disponible uniquement si un proxy actif est sélectionné dans le champ *Proxy mode*. Les macros ne sont pas prises en charge.
Interface	Saisissez les détails de l'interface d'un proxy passif. Ce champ est disponible uniquement si un proxy passif est sélectionné dans le champ *Proxy mode*.
Address	Adresse IP/nom DNS du proxy passif.

Parameter	Description
Port	Numéro de port TCP du proxy passif (10051 par défaut). Les macros utilisateur sont prises en charge.
Description	Saisissez la description du proxy.

L'onglet **Encryption** vous permet d'exiger des connexions chiffrées avec le proxy.

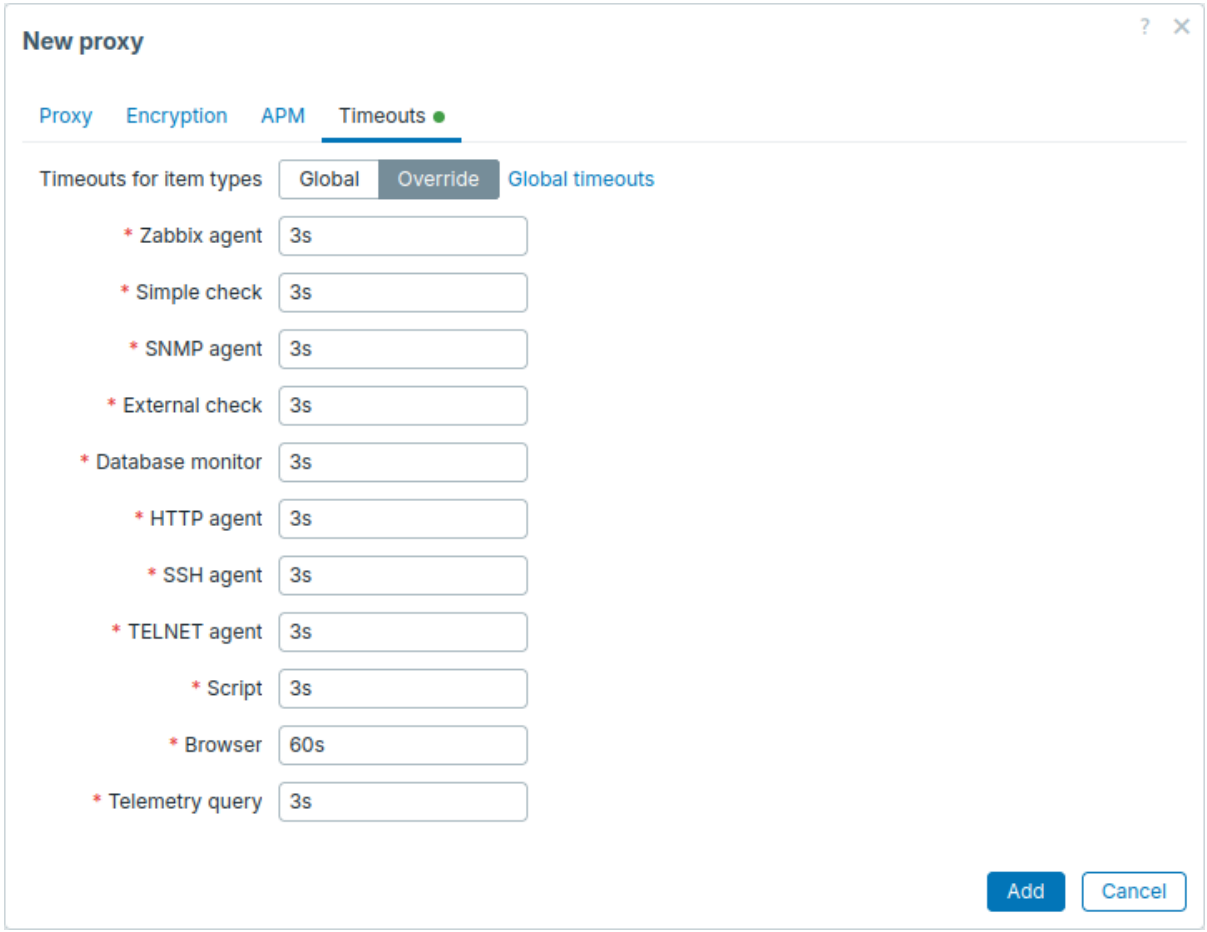
Parameter	Description
Connections to proxy	Méthode utilisée par le serveur pour se connecter au proxy passif : sans chiffrement (par défaut), avec une clé PSK (clé prépartagée) ou un certificat.
Connections from proxy	Sélectionnez les types de connexions autorisés depuis le proxy actif. Plusieurs types de connexions peuvent être sélectionnés simultanément (utile pour les tests et le passage à un autre type de connexion). La valeur par défaut est « No encryption ».
Issuer	Émetteur autorisé du certificat. Le certificat est d'abord validé avec l'AC (autorité de certification). S'il est valide et signé par l'AC, le champ *Issuer* peut être utilisé pour restreindre davantage les AC autorisées. Ce champ est facultatif et destiné à être utilisé si votre installation Zabbix utilise des certificats provenant de plusieurs AC.
Subject	Sujet autorisé du certificat. Le certificat est d'abord validé avec l'AC. S'il est valide et signé par l'AC, le champ *Subject* peut être utilisé pour n'autoriser qu'une seule valeur de la chaîne *Subject*. Si ce champ est vide, tout certificat valide signé par l'AC configurée est accepté.
PSK identity	Chaîne d'identité de la clé prépartagée. Ne placez pas d'informations sensibles dans l'identité PSK : elle est transmise sans chiffrement sur le réseau afin d'indiquer au récepteur quelle PSK utiliser.
PSK	Clé prépartagée (chaîne hexadécimale). Longueur maximale : 512 chiffres hexadécimaux (PSK de 256 octets) si Zabbix utilise la bibliothèque GnuTLS ou OpenSSL, 64 chiffres hexadécimaux (PSK de 32 octets) si Zabbix utilise la bibliothèque mbed TLS (PolarSSL). Exemple : 1f87b595725ac58dd977beef14b97461a7c1045b9a1c963065002c5473194952

L'onglet **APM** vous permet de configurer la supervision des performances des applications (APM) sur le proxy. Le proxy peut collecter les données de télémétrie (traces, métriques et journaux) envoyées par des applications ou des collecteurs OpenTelemetry via le protocole OpenTelemetry (OTLP/gRPC), et écrire ces données dans la source de données de télémétrie.



Parameter	Description
Data collection enabled	Cochez la case pour activer la collecte des données de télémétrie sur le proxy. Avant d'activer la collecte des données de télémétrie, définissez le paramètre `StartAPMCollector` sur 1 dans le fichier de configuration du proxy (nécessite un proxy Zabbix avec prise en charge de l'APM).
Max messages per second	Définissez le nombre maximal de messages par seconde que le proxy accepte : **Unlimited** - accepter tous les messages entrants. **Custom** - accepter jusqu'au nombre de messages par seconde indiqué. Un message est une requête OTLP envoyée par une application, un collecteur OpenTelemetry ou une autre source de télémétrie. Sa taille dépend de l'émetteur ; par exemple, un collecteur OpenTelemetry peut regrouper des milliers de spans dans un seul message, tandis qu'une application simple peut envoyer des messages plus petits (par exemple, un par trace). Si la limite est dépassée, le proxy répond avec le code d'état gRPC `UNAVAILABLE` (erreur temporaire), de sorte que l'émetteur puisse réessayer d'envoyer le message ultérieurement, selon sa configuration.
Add resource attributes	Indiquez les attributs de ressource à ajouter aux données de télémétrie entrantes du type sélectionné (par exemple, pour marquer toutes les données reçues par ce proxy avec son environnement ou sa région). Vous pouvez ensuite utiliser ces attributs pour filtrer les données de télémétrie, par exemple dans les conditions des éléments de requête de télémétrie. Si les données entrantes contiennent déjà un attribut de ressource portant le même nom, l'attribut entrant est conservé à la place de celui indiqué ici.
Name	Saisissez le nom de l'attribut (par exemple, `deployment.environment`, `cloud.region`). Il doit être unique pour le *Type* sélectionné.
Value	Saisissez la valeur de l'attribut (par exemple, `production`, `eu-central-1`). Peut être vide.
Type	Sélectionnez le type de données entrantes auquel l'attribut s'applique : *Traces*, *Metrics* ou *Logs*.

L'onglet **Timeouts** vous permet de remplacer les délais d'expiration globaux pour les types d'éléments qui prennent cette fonctionnalité en charge.



Parameter	Description
Timeouts for item types	Définissez le délai d'expiration de l'élément (en fonction de son type) : **Global** - utiliser le délai d'expiration global (affiché dans le champ *Timeout* grisé pour chaque type d'élément) ; **Override** - définir un délai d'expiration personnalisé (dans le champ *Timeout* de chaque type d'élément). Plage autorisée : 1 à 600 s (valeur par défaut : héritée des délais d'expiration globaux). Les suffixes de durée, par exemple 30s et 1m, ainsi que les macros utilisateur sont pris en charge. Cliquer sur le lien *Global timeouts* permet de configurer les délais d'expiration globaux. Notez que le lien *Global timeouts* est visible uniquement par les utilisateurs de type *Super admin* disposant des autorisations nécessaires pour la section d'interface *Administration* > *General*. Notez que, bien que les délais d'expiration au niveau du proxy remplacent les délais globaux, ils sont eux-mêmes remplacés par les délais d'expiration individuels des éléments, s'ils sont configurés.

Note:

Si la version majeure du proxy ne correspond pas à celle du serveur, l'icône s'affiche à côté de *Timeouts for item types*, avec le message au survol « Timeouts disabled because the proxy and server versions do not match ». Dans ce cas, le proxy utilise le paramètre `Timeout` du fichier de configuration du proxy.

Le formulaire de modification d'un proxy existant comporte les boutons supplémentaires suivants :

- *Refresh configuration* - actualiser la configuration du proxy ;
- *Clone* - créer un nouveau proxy à partir des propriétés du proxy existant ;
- *Delete* - supprimer le proxy.

Configuration de l'hôte

Vous pouvez indiquer qu'un hôte individuel doit être surveillé par un proxy ou un groupe de proxy dans le formulaire de configuration de l'hôte, à l'aide du champ *Surveillé par*.



La mise à jour en masse des hôtes est une autre façon d'indiquer que les hôtes doivent être surveillés par un proxy ou un groupe de proxy.

Équilibrage de charge et haute disponibilité du proxy

Présentation générale

Les proxy Zabbix peuvent être organisés en groupes de proxy afin d'activer l'équilibrage de charge et la haute disponibilité des proxy.

L'équilibrage de charge et la haute disponibilité des proxy correspondent à la redistribution automatique des hôtes entre les proxy d'un groupe de proxy :

- Si un proxy se déconnecte, ses hôtes sont déplacés vers d'autres proxy, ce qui permet de maintenir une haute disponibilité des proxy.
- Si un proxy possède un nombre d'hôtes beaucoup plus élevé ou plus faible que les autres proxy, ses hôtes sont déplacés vers d'autres proxy afin d'équilibrer la charge des proxy.

La redistribution des hôtes fonctionne uniquement entre les proxy d'un groupe qui remplissent les conditions suivantes :

- Les proxy exécutent Zabbix 7.0 ou une version ultérieure.
- La version du proxy correspond à la version du serveur Zabbix. Si vous utilisez Zabbix agent (passif), la version du proxy doit correspondre à la version de l'agent. Les agents actifs nécessitent uniquement Zabbix 7.0 ou une version ultérieure.
- Le groupe de proxy possède un état en ligne.
- Les hôtes sont configurés pour être supervisés par un groupe de proxy plutôt que par des proxy individuels.

Note:
L'état d'un groupe de proxy peut être surveillé à l'aide de vérifications internes par n'importe quel hôte affecté à un groupe de proxy. Cependant, pour surveiller l'état d'un proxy individuel dans un groupe, affectez l'hôte à ce proxy ; sinon, les résultats peuvent être incohérents.

Redistribution des hôtes

L'équilibrage de charge des proxy et la haute disponibilité sont gérés par le serveur Zabbix via le gestionnaire de groupe de proxy, qui surveille en continu l'état de tous les proxy de chaque groupe de proxy ainsi que leur répartition des hôtes.

La haute disponibilité des proxy au sein d'un groupe est assurée par le basculement des proxy: lorsqu'un proxy passe hors ligne, ses hôtes sont immédiatement redistribués vers d'autres proxy. L'équilibrage de charge des proxy a également lieu, car les hôtes sont réaffectés aux proxy ayant le moins d'hôtes attribués.

En outre, l'équilibrage de charge des proxy est déclenché lorsque le nombre d'hôtes d'un proxy diffère de la moyenne du groupe d'au moins 10 hôtes et d'un facteur de 2 (excédent d'hôtes ou déficit d'hôtes). Si le déséquilibre persiste après une période de grâce (10 x délai de basculement), le groupe de proxy est placé en file d'attente pour la redistribution des hôtes.

Le gestionnaire de groupe de proxy redistribue les hôtes selon la logique suivante:

1. Calculer le nombre moyen d'hôtes par proxy.
2. Pour les proxy avec un excédent d'hôtes, déplacer les hôtes excédentaires vers le pool d'hôtes non attribués des proxy.
3. Pour les proxy avec un déficit d'hôtes, calculer combien d'hôtes sont nécessaires pour atteindre l'équilibre.
4. Retirer le nombre requis d'hôtes des proxy ayant le plus d'hôtes.
5. Déplacer les hôtes non attribués vers les proxy ayant le moins d'hôtes.

Exemples de redistribution des hôtes:

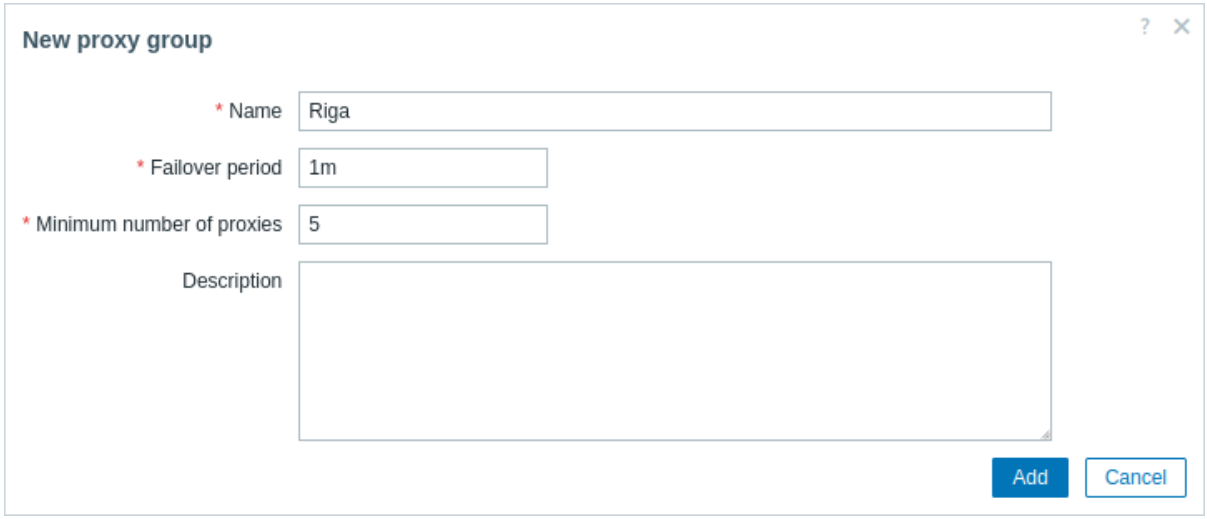
Hosts on proxy	Group average	Host reassignment
100	50	Yes
60	50	No
40	50	No
25	50	Yes
15	5	Yes
10	5	No

Attention:
Avoir moins de 10 hôtes surveillés par un groupe de proxy peut entraîner une répartition inégale des hôtes entre les proxy du groupe.

Configuration d'un groupe de proxy

Pour configurer un groupe de proxy dans l'interface Zabbix :

1. Accédez à *Administration > Groupes de proxy*.
2. Cliquez sur *Créer un groupe de proxy*.
3. Saisissez les paramètres du groupe de proxy dans le formulaire.



Paramètre	Description
Nom	Nom du groupe de proxy.
Période de basculement	Période, en secondes, pendant laquelle un proxy du groupe de proxy doit communiquer avec le serveur Zabbix pour être considéré comme étant en ligne (valeur par défaut : 1m ; plage : 10s-15m). Si le proxy ne communique pas pendant cette période, son état passe à *Hors ligne* et ses hôtes sont immédiatement redistribués vers d'autres proxy. L'équilibrage de charge des proxy commence après 10 x cette période. Prend en charge les suffixes de temps (par exemple, 30s, 1m) et les macros utilisateur.
Nombre minimal de proxy	Nombre minimal de proxy en ligne requis pour maintenir le groupe de proxy en ligne (valeur par défaut : 1 ; plage : 1-1000). Prend en charge les macros utilisateur. Cette valeur doit être inférieure au nombre total de proxy dans le groupe. Par exemple, dans un groupe de 10 proxy, définir le minimum sur 10 entraînera la mise hors ligne du groupe si un proxy tombe en panne. Notez que les proxy en ligne d'un groupe hors ligne continuent de fonctionner normalement, mais l'équilibrage de charge/la haute disponibilité ne seront pas assurés.
Description	Description du groupe de proxy.
Proxy	Affiche une liste de cinq proxy au maximum (sous forme de liens ou en texte brut, selon les autorisations de l'utilisateur sur les proxy) lors de la modification d'un groupe contenant des proxy.

Configuration de l'équilibrage de charge des proxy

Pour utiliser l'équilibrage de charge des proxy, vous devez configurer un groupe de proxy dans l'interface Zabbix (voir ci-dessus) et vous assurer que les hôtes sont surveillés par un groupe de proxy, et non par des proxy individuels (vous pouvez utiliser la mise à jour en masse des hôtes pour déplacer les hôtes des proxy vers le groupe de proxy).

Si vous utilisez Zabbix agent, configurez-le également comme suit :

- Pour les vérifications passives, répertoriez tous les proxy du groupe de proxy dans le paramètre Server.
- Pour les vérifications actives, il est recommandé de répertorier tous les proxy du groupe de proxy ou le serveur Zabbix dans le paramètre ServerActive. Notez que seul Zabbix agent 7.0 (ou version ultérieure) fonctionnera avec les groupes de proxy en mode actif.

Si le paramètre `ServerActive` ne contient qu'un seul proxy d'un groupe de proxy (ou le serveur Zabbix), l'agent pourra tout de même se connecter au proxy approprié. Lorsque le service agent démarre et se connecte au proxy spécifié, l'agent reçoit et met

en cache la liste complète des adresses IP des proxy ainsi que leur charge actuelle au sein du groupe. Les vérifications actives sont alors redirigées vers le proxy en ligne approprié pour l'hôte, en fonction de l'affectation actuelle des hôtes aux proxy au sein du groupe de proxy.

Warning:
Le fait de ne spécifier qu'un seul proxy dans le paramètre `ServerActive` de Zabbix agent peut entraîner la perte de données de supervision si l'agent est démarré ou redémarré alors que le proxy spécifié est hors ligne.

Note:
Lors de l'utilisation de Zabbix sender, les demandes de données sont également redirigées vers le proxy en ligne approprié pour l'hôte, en fonction de l'affectation actuelle des hôtes aux proxy au sein du groupe de proxy. Cependant, si vous envoyez les valeurs de plusieurs hôtes depuis un fichier d'entrée, utilisez l'option `-g` pour éviter d'envoyer les données au mauvais proxy.

Zabbix agent doit également pouvoir se connecter à tous les proxy du groupe de proxy à travers le pare-feu. Dans le cas contraire, les vérifications actives peuvent rester bloquées ou échouer lors d'une redirection ou d'un basculement. Par exemple :

- Lors des vérifications actives, un proxy peut rediriger l'agent vers un autre proxy. Si ce proxy est bloqué par un pare-feu, la communication restera bloquée en attendant une réponse.
- Dans les configurations haute disponibilité stables sans rééquilibrage récent, les agents peuvent ne jamais contacter les proxy de secours. Si les règles du pare-feu ont été modifiées et n'ont pas été testées, le basculement peut échouer.

Tester l'équilibrage de charge des proxy

Pour tester l'équilibrage de charge des proxy :

1. Configurez un groupe de proxy.
2. Assurez-vous que le groupe de proxy est en ligne.
3. Assurez-vous que les hôtes sont supervisés par un groupe de proxy, et non par des proxy individuels (vous pouvez utiliser la mise à jour en masse des hôtes pour déplacer les hôtes des proxy vers le groupe de proxy).
4. Attendez quelques secondes pour la mise à jour de la configuration et la répartition des hôtes entre les proxy du groupe de proxy. Observez le changement en actualisant la liste des hôtes dans *Administration* > *Proxy*.

Remarques importantes

- Les traps SNMP ne sont pas pris en charge par les proxy d'un groupe de proxy.
- Les vérifications qui dépendent d'une configuration externe (par exemple, les scripts pour les vérifications externes ou la configuration ODBC pour les vérifications de base de données) doivent disposer de la même configuration sur tous les proxy du groupe de proxy.
- Les vérifications de base de données nécessitent des autorisations étendues sur l'objet ou le serveur de base de données.
- Les hôtes VMware surveillés par un groupe de proxy sont répartis aléatoirement entre les proxy du groupe. Cela entraîne la mise en cache de toutes les données VMware par chaque proxy, ce qui augmente la charge sur vCenter.
- Les hôtes créés à partir de données d'auto-enregistrement provenant d'un proxy appartenant à un groupe de proxy sont définis comme étant surveillés par ce groupe de proxy. Cependant, les hôtes créés à partir de données de découverte réseau provenant d'un proxy appartenant à un groupe de proxy sont définis comme étant surveillés par ce proxy.

Synchronisation de la configuration de surveillance

Vue d'ensemble

Cette page fournit des détails sur la mise à jour de la configuration de surveillance pour le proxy, c'est-à-dire sur la manière dont les modifications apportées à la configuration de surveillance sur le serveur sont synchronisées vers le proxy.

Mise à jour incrémentielle

La mise à jour de la configuration du proxy est incrémentielle. Lors d'une synchronisation de la configuration, seules les entités modifiées sont mises à jour (ainsi, si aucune entité n'a été modifiée, rien ne sera envoyé). Cette approche permet d'économiser des ressources et de définir un intervalle plus court (presque instantané) pour la mise à jour de la configuration du proxy.

Les modifications de la configuration du proxy sont suivies à l'aide de numéros de révision. Seules les entités dont les révisions sont supérieures à la révision de la configuration du proxy sont incluses dans les données de configuration envoyées au proxy.

Les entités concernées par une synchronisation de la configuration sont les suivantes :

Entity	Details
données TLS d'enregistrement automatique	Toutes les données TLS d'enregistrement automatique.
expressions	Toutes les expressions (expressions régulières, tables d'expressions).
configuration globale	Configuration globale définie dans la table 'config'
hôte	Toutes les propriétés, interfaces, informations d'inventaire, éléments, prétraitements des éléments, paramètres des éléments et scénarios web d'un hôte.
macros d'hôte	Toutes les macros définies sur un hôte et tous les ID de modèles qui lui sont liés.
règle de découverte du proxy	Règles de découverte et vérifications attribuées à un proxy.

Cela signifie que :

- Si un élément est modifié sur un **hôte**, toute la configuration de cet hôte sera synchronisée.
- Si une **expression régulière** est modifiée, toutes les expressions régulières seront synchronisées.

Les macros d'hôte constituent une exception : elles sont également envoyées si un élément quelconque de l'hôte a été modifié.

La commande `-R config_cache_reload` exécutée sur le proxy lancera également une mise à jour incrémentielle.

Notez qu'une synchronisation complète de la configuration aura lieu au démarrage ou au redémarrage d'un proxy, lors d'un basculement HA, si le jeton de session a changé ou si la mise à jour de la configuration a échoué sur le proxy, par exemple si la connexion a été interrompue pendant la réception des données de configuration.

Paramètres de configuration

Le paramètre **ProxyConfigFrequency** détermine la fréquence à laquelle la configuration du proxy est synchronisée avec le serveur (10 secondes par défaut).

Notez que ProxyConfigFrequency est :

- un paramètre du serveur pour les proxies passifs
- un paramètre du proxy pour les proxies actifs

Sur les proxies actifs, ProxyConfigFrequency est un nouveau paramètre depuis Zabbix 6.4 et doit être utilisé à la place de ConfigFrequency, désormais obsolète.

> **Attention:**
> Si ProxyConfigFrequency et ConfigFrequency sont tous deux utilisés, le proxy consignera une erreur et s'arrêtera.

Calcul des files d'attente pendant la maintenance

> **Attention:**
> Le proxy Zabbix n'a pas connaissance des périodes de maintenance ; voir Calcul des files d'attente pendant la maintenance pour plus de détails.

Protocole d'échange de données entre serveur et proxy

Serveur - l'échange de données avec le proxy est basé sur le format JSON.

Les messages de requête et de réponse doivent commencer par en-tête et longueur des données.

proxy passif

Demande de configuration

Le serveur envoie d'abord une requête `proxy config` vide. Cette requête est envoyée toutes les `ProxyConfigFrequency` secondes (paramètre de configuration du serveur).

Le proxy répond avec la version actuelle du proxy, le jeton de session et la révision de la configuration. Le serveur répond avec les données de configuration qui doivent être mises à jour.

name	value type	description
server→proxy: **request**	*string*	'proxy config'

name	value type	description
proxy→server:		
version	*string*	Version du proxy (<major>.<minor>.<build>).
session	*string*	Jeton de session de la configuration du proxy.
config_revision	*number*	Révision de la configuration du proxy.
server→proxy:		
full_sync	*number*	1 - si les données complètes de configuration sont envoyées ; absent dans le cas contraire (facultatif).
data	*array*	Objet contenant les données des tables. Absent si la configuration n'a pas été modifiée (facultatif).
<table>	*object*	Un ou plusieurs objets contenant les données de <table> (facultatif, selon les modifications).
fields	*array*	Tableau des noms de champs.
-	*string*	Nom du champ.
data	*array*	Tableau de lignes.
-	*array*	Tableau de colonnes.
-	*string*,*number*	Valeur de colonne dont le type dépend du type de colonne dans le schéma de la base de données.
macro.secrets	*object*	Informations sur les macros secrètes, absentes s'il n'y a aucune modification des macros du coffre (facultatif).
config_revision	*number*	Révision du cache de configuration - envoyée avec les données de configuration (facultatif).
del_hostids	*array*	Tableau des hostids supprimés (facultatif).
-	*number*	Identifiant du hôte.
del_macro_hostids	*array*	Tableau des hostids dont toutes les macros ont été supprimées (facultatif).
-	*number*	Identifiant du hôte.
proxy→server:		
response	*string*	Informations sur la réussite de la requête ('success' ou 'failed').
version	*string*	Version du proxy (<major>.<minor>.<build>).

Exemple :

serveur→proxy :

```
{
  "request":"proxy config"
}
```

proxy→serveur :

```
{
  "version": "8.0.0",
  "session": "0033124949800811e5686dbfd9bcea98",
  "config_revision": 0
}
```

serveur→proxy:

```
{
    "full_sync": 1,
    "data": {
        "hosts": {
            "fields": ["hostid", "host", "status", "ipmi_authtype", "ipmi_privilege", "ipmi_username", "ip
            "data": [
                [10084, "Zabbix server", 0, -1, 2, "", "", "Zabbix server", 1, 1, "", "", "", ""]
            ]
        },
        "interface": {
            "fields": ["interfaceid", "hostid", "main", "type", "useip", "ip", "dns", "port", "available"]
```

```
            "data": [
                [1, 10084, 1, 1, 1, "127.0.0.1", "", "10053", 1]
            ]
        },
        "interface_snmp": {
            "fields": ["interfaceid", "version", "bulk", "community", "securityname", "securitylevel", "au
            "data": []
        },
        "host_inventory": {
            "fields": ["hostid", "type", "type_full", "name", "alias", "os", "os_full", "os_short", "seria
            "data": [
                [10084, "", "", ..., "56.95387", "24.22067", ..., ""]
            ]
        },
        "items": {
            "fields": ["itemid", "type", "snmp_oid", "hostid", "key_", "delay", "history", "status", "valu
            "data": [
                [44161, 7, "", 10084, "agent.hostmetadata", "10s", "90d", 0, 1, "", "", "", "", 0, "", "",
                [44162, 0, "", 10084, "agent.ping", "10s", "90d", 0, 3, "", "", "", "", 0, "", "", "", "",
            ]
        },
        "item_rtdata": {
            "fields": ["itemid", "lastlogsize", "mtime"],
            "data": [
                [44161, 0, 0],
                [44162, 0, 0]
            ]
        },
        "item_preproc": {
            "fields": ["item_preprocid", "itemid", "step", "type", "params", "error_handler", "error_handl
            "data": []
        },
        "item_parameter": {
            "fields": ["item_parameterid", "itemid", "name", "value"],
            "data": []
        },
        "globalmacro": {
            "fields": ["globalmacroid", "macro", "value", "type"],
            "data": [
                [2, "{$SNMP_COMMUNITY}", "public", 0]
            ]
        },
        "hosts_templates": {
            "fields": ["hosttemplateid", "hostid", "templateid", "link_type"],
            "data": []
        },
        "hostmacro": {
            "fields": ["hostmacroid", "hostid", "macro", "value", "type", "automatic"],
            "data": [
                [5676, 10084, "{$M}", "AppID=zabbix_server&Query=Safe=passwordSafe;Object=zabbix:Content",
            ]
        },
        "drules": {
            "fields": ["druleid", "name", "iprange", "delay"],
            "data": [
                [2, "Local network", "127.0.0.1", "10s"]
            ]
        },
        "dchecks": {
            "fields": ["dcheckid", "druleid", "type", "key_", "snmp_community", "ports", "snmpv3_securityn
            "data": [
                [2, 2, 9, "system.uname", "", "10052", "", 0, "", "", 0, 0, 0, "", 1, 0]
```

```
            ]
        },
        "regexps": {
            "fields": ["regexpid", "name"],
            "data": [
                [1, "File systems for discovery"],
                [2, "Network interfaces for discovery"],
                [3, "Storage devices for SNMP discovery"],
                [4, "Windows service names for discovery"],
                [5, "Windows service startup states for discovery"]
            ]
        },
        "expressions": {
            "fields": ["expressionid", "regexpid", "expression", "expression_type", "exp_delimiter", "case
            "data": [
                [1, 1, "^(btrfs|ext2|ext3|ext4|reiser|xfs|ffs|ufs|jfs|jfs2|vxfs|hfs|apfs|refs|ntfs|fat32|z
                [3, 3, "^(Physical memory|Virtual memory|Memory buffers|Cached memory|Swap space)$", 4, ",
                [5, 4, "^(MMCSS|gupdate|SysmonLog|clr_optimization_v2.0.50727_32|clr_optimization_v4.0.303
                [6, 5, "^(automatic|automatic delayed)$", 3, ",", 1],
                [7, 2, "^Software Loopback Interface", 4, ",", 1],
                [8, 2, "^(In)?[Ll]oop[Bb]ack[0-9._]*$", 4, ",", 1],
                [9, 2, "^NULL[0-9.]*$", 4, ",", 1],
                [10, 2, "^[Ll]o[0-9.]*$", 4, ",", 1],
                [11, 2, "^[Ss]ystem$", 4, ",", 1],
                [12, 2, "^Nu[0-9.]*$", 4, ",", 1]
            ]
        },
        "settings": {
            "fields": ["name", "type", "value_str", "value_int"],
            "data": [
                ["autoreg_tls_accept", 2, "", 1],
                ["hk_history_global", 2, "", 0],
                ["snmptrap_logging", 2, "", 1],
                ["proxy_secrets_provider", 2, "", 0],
                ["hk_history", 1, "31d", 0],
                ["timeout_db_monitor", 1, "3s", 0],
                ["timeout_external_check", 1, "3s", 0],
                ["timeout_http_agent", 1, "3s", 0],
                ["timeout_simple_check", 1, "3s", 0],
                ["timeout_snmp_agent", 1, "3s", 0],
                ["timeout_ssh_agent", 1, "3s", 0],
                ["timeout_telnet_agent", 1, "3s", 0],
                ["timeout_zabbix_agent", 1, "3s", 0],
                ["timeout_browser", 1, "30s", 0]
            ]
        },
        "httptest": {
            "fields": ["httptestid", "name", "delay", "agent", "authentication", "http_user", "http_passwo
            "data": []
        },
        "httptestitem": {
            "fields": ["httptestitemid", "httptestid", "itemid", "type"],
            "data": []
        },
        "httptest_field": {
            "fields": ["httptest_fieldid", "httptestid", "type", "name", "value"],
            "data": []
        },
        "httpstep": {
            "fields": ["httpstepid", "httptestid", "name", "no", "url", "timeout", "posts", "required", "s
            "data": []
        },
```

```
        "httpstepitem": {
            "fields": ["httpstepitemid", "httpstepid", "itemid", "type"],
            "data": []
        },
        "httpstep_field": {
            "fields": ["httpstep_fieldid", "httpstepid", "type", "name", "value"],
            "data": []
        },
        "config_autoreg_tls": {
            "fields": ["autoreg_tlsid", "tls_psk_identity", "tls_psk"],
            "data": [
                [1, "", ""]
            ]
        }
    },
    "macro.secrets": {
        "AppID=zabbix_server&Query=Safe=passwordSafe;Object=zabbix": {
            "Content": "738"
        }
    },
    "config_revision": 2
}
```

proxy→serveur:

```
{
  "response": "success",
  "version": "8.0.0"
}
```

Demande de données

La requête `proxy data` est utilisée pour obtenir la disponibilité des interfaces d'hôte, ainsi que les données d'historique, de découverte et d'enregistrement automatique depuis le proxy. Cette requête est envoyée toutes les `ProxyDataFrequency` secondes (paramètre de configuration du serveur).

name	value type	description
server→proxy:		
request	*string*	'proxy data'
proxy→server:		
session	*string*	Jeton de session de données.
interface availability	*array*	*(optional)* Tableau d'objets de disponibilité des interfaces.
interfaceid	*number*	Identifiant de l'interface.
available	*number*	Disponibilité de l'interface: **0**, *INTERFACE_AVAILABLE_UNKNOWN* - inconnue **1**, *INTERFACE_AVAILABLE_TRUE* - disponible **2**, *INTERFACE_AVAILABLE_FALSE* - indisponible
error	*string*	Message d'erreur de l'interface ou chaîne vide.
history data	*array*	*(optional)* Tableau d'objets de données d'historique.
itemid	*number*	Identifiant de l'élément.
clock	*number*	Horodatage de la valeur de l'élément (secondes).
ns	*number*	Horodatage de la valeur de l'élément (nanosecondes).
value	*string*	*(optional)* Valeur de l'élément.
id	*number*	Identifiant de la valeur (compteur croissant, unique au sein d'une session de données).
timestamp	*number*	*(optional)* Horodatage des éléments de type journal.
source	*string*	*(optional)* Valeur de la source de l'élément eventlog.
severity	*number*	*(optional)* Valeur de gravité de l'élément eventlog.
eventid	*number*	*(optional)* Valeur eventid de l'élément eventlog.

name	value type	description
state	*string*	*(optional)* État de l'élément: **0**, *ITEM_STATE_NORMAL* **1**, *ITEM_STATE_NOTSUPPORTED*
lastlogsize	*number*	*(optional)* Dernière taille de journal des éléments de type journal.
mtime	*number*	*(optional)* Heure de modification des éléments de type journal.
discovery data	*array*	*(optional)* Tableau d'objets de données de découverte.
clock	*number*	Horodatage des données de découverte.
druleid	*number*	Identifiant de la règle de découverte.
dcheckid	*number*	Identifiant du contrôle de découverte ou null pour les données de règle de découverte.
type	*number*	Type de contrôle de découverte: **-1** données de règle de découverte **0**, *SVC_SSH* - contrôle du service SSH **1**, *SVC_LDAP* - contrôle du service LDAP **2**, *SVC_SMTP* - contrôle du service SMTP **3**, *SVC_FTP* - contrôle du service FTP **4**, *SVC_HTTP* - contrôle du service HTTP **5**, *SVC_POP* - contrôle du service POP **6**, *SVC_NNTP* - contrôle du service NNTP **7**, *SVC_IMAP* - contrôle du service IMAP **8**, *SVC_TCP* - contrôle de disponibilité du port TCP **9**, *SVC_AGENT* - agent Zabbix **10**, *SVC_SNMPv1* - agent SNMPv1 **11**, *SVC_SNMPv2* - agent SNMPv2 **12**, *SVC_ICMPPING* - ping ICMP **13**, *SVC_SNMPv3* - agent SNMPv3 **14**, *SVC_HTTPS* - contrôle du service HTTPS **15**, *SVC_TELNET* - contrôle de disponibilité Telnet
ip	*string*	Adresse IP de l'hôte.
dns	*string*	Nom DNS de l'hôte.
port	*number*	*(optional)* Numéro de port du service.
key_	*string*	*(optional)* Clé de l'élément pour le contrôle de découverte de type **9** *SVC_AGENT*
value	*string*	*(optional)* Valeur reçue du service, peut être vide pour la plupart des services.
status	*number*	*(optional)* État du service: **0**, *DOBJECT_STATUS_UP* - service UP **1**, *DOBJECT_STATUS_DOWN* - service DOWN
auto registration	*array*	*(optional)* Tableau d'objets de données d'enregistrement automatique.
clock	*number*	Horodatage des données d'enregistrement automatique.
host	*string*	Nom de l'hôte.
ip	*string*	*(optional)* Adresse IP de l'hôte.
dns	*string*	*(optional)* Nom DNS résolu à partir de l'adresse IP.
port	*string*	*(optional)* Port de l'hôte.
host_metadata	*string*	*(optional)* Métadonnées de l'hôte envoyées par l'agent (selon le paramètre de configuration de l'agent HostMetadata ou HostMetadataItem).
tasks	*array*	*(optional)* Tableau de tâches.
type	*number*	Type de tâche: **0**, *ZBX_TM_TASK_PROCESS_REMOTE_COMMAND_RESULT* - résultat de commande distante

name	value type	description
status	*number*	État d'exécution de la commande distante: **0**, *ZBX_TM_REMOTE_COMMAND_COMPLETED* - commande distante exécutée avec succès **1**, *ZBX_TM_REMOTE_COMMAND_FAILED* - échec de la commande distante
error	*string*	*(optional)* Message d'erreur.
parent_taskid	*number*	ID de la tâche parente.
more	*number*	*(optional)* 1 - il y a d'autres données d'historique à envoyer.
clock	*number*	*(optional)* Horodatage du transfert de données (secondes).
ns	*number*	*(optional)* Horodatage du transfert de données (nanosecondes).
version	*string*	Version du proxy (<major>.<minor>.<build>).
server→proxy:		
response	*string*	Information de réussite de la requête ('success' ou 'failed').
tasks	*array*	*(optional)* Tableau de tâches.
type	*number*	Type de tâche: **1**, *ZBX_TM_TASK_PROCESS_REMOTE_COMMAND* - commande distante
clock	*number*	Heure de création de la tâche.
ttl	*number*	Temps en secondes après lequel la tâche expire.
commandtype	*number*	Type de commande distante: **0**, *ZBX_SCRIPT_TYPE_CUSTOM_SCRIPT* - utiliser un script personnalisé **1**, *ZBX_SCRIPT_TYPE_IPMI* - utiliser IPMI **2**, *ZBX_SCRIPT_TYPE_SSH* - utiliser SSH **3**, *ZBX_SCRIPT_TYPE_TELNET* - utiliser Telnet **4**, *ZBX_SCRIPT_TYPE_GLOBAL_SCRIPT* - utiliser un script global (actuellement fonctionnellement équivalent à un script personnalisé)
command	*string*	Commande distante à exécuter.
execute_on	*number*	Cible d'exécution pour les scripts personnalisés: **0**, *ZBX_SCRIPT_EXECUTE_ON_AGENT* - exécuter le script sur l'agent **1**, *ZBX_SCRIPT_EXECUTE_ON_SERVER* - exécuter le script sur le serveur **2**, *ZBX_SCRIPT_EXECUTE_ON_PROXY* - exécuter le script sur le proxy
port	*number*	*(optional)* Port pour les commandes Telnet et SSH.
authtype	*number*	*(optional)* Type d'authentification pour les commandes SSH.
username	*string*	*(optional)* Nom d'utilisateur pour les commandes Telnet et SSH.
password	*string*	*(optional)* Mot de passe pour les commandes Telnet et SSH.
publickey	*string*	*(optional)* Clé publique pour les commandes SSH.
privatekey	*string*	*(optional)* Clé privée pour les commandes SSH.
parent_taskid	*number*	ID de la tâche parente.
hostid	*number*	ID de l'hôte cible.

Exemple :

server→proxy :

```
{
  "request": "proxy data"
}
```

proxy→server :

```
{
    "session": "12345678901234567890123456789012"
    "interface availability": [
        {
            "interfaceid": 1,
            "available": 1,
            "error": ""
    },
        {
            "interfaceid": 2,
```

```
            "available": 2,
            "error": "Get value from agent failed: cannot connect to [[127.0.0.1]:10049]: [111] Connection
    },
        {
            "interfaceid": 3,
            "available": 1,
            "error": ""
    },
        {
            "interfaceid": 4,
            "available": 1,
            "error": ""
    }
    ],
    "history data":[
        {
            "itemid":"12345",
            "clock":1478609647,
            "ns":332510044,
            "value":"52956612",
            "id": 1
        },
        {
            "itemid":"12346",
            "clock":1478609647,
            "ns":330690279,
            "state":1,
            "value":"Cannot find information for this network interface in /proc/net/dev.",
            "id": 2
        }
    ],
    "discovery data":[
        {
            "clock":1478608764,
            "drule":2,
            "dcheck":3,
            "type":12,
            "ip":"10.3.0.10",
            "dns":"vdebian",
            "status":1
        },
        {
            "clock":1478608764,
            "drule":2,
            "dcheck":null,
            "type":-1,
            "ip":"10.3.0.10",
            "dns":"vdebian",
            "status":1
        }
    ],
    "auto registration":[
        {
            "clock":1478608371,
            "host":"Logger1",
            "ip":"10.3.0.1",
            "dns":"localhost",
            "port":"10050"
        },
        {
            "clock":1478608381,
            "host":"Logger2",
```

```
            "ip":"10.3.0.2",
            "dns":"localhost",
            "port":"10050"
        }
    ],
    "tasks":[
        {
            "type": 0,
            "status": 0,
            "parent_taskid": 10
        },
        {
            "type": 0,
            "status": 1,
            "error": "No permissions to execute task.",
            "parent_taskid": 20
        }
    ],
    "version":"8.0.0"
}
```


server→proxy :


```
{
  "response": "success",
  "tasks":[
      {
         "type": 1,
         "clock": 1478608371,
         "ttl": 600,
         "commandtype": 2,
         "command": "restart_service1.sh",
         "execute_on": 2,
         "port": 80,
         "authtype": 0,
         "username": "userA",
         "password": "password1",
         "publickey": "MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQCqGKukO1De7zhZj6+H0qtjTkVxwTCpvKe",
         "privatekey": "lsuusFncCzWBQ7RKNUSesmQRMSGkVb1/3j+skZ6UtW+5u09lHNsj6tQ5QCqGKukO1De7zhd",
         "parent_taskid": 10,
         "hostid": 10070
      },
      {
         "type": 1,
         "clock": 1478608381,
         "ttl": 600,
         "commandtype": 1,
         "command": "restart_service2.sh",
         "execute_on": 0,
         "authtype": 0,
         "username": "",
         "password": "",
         "publickey": "",
         "privatekey": "",
         "parent_taskid": 20,
         "hostid": 10084
      }
  ]
}
```


Requête de tâches

L'échange `proxy tasks` gère la livraison et l'accusé de réception des tâches de commandes à distance entre le serveur et un proxy passif. Lorsque le serveur interroge le proxy pour obtenir les résultats des tâches, il envoie une requête `proxy tasks` vide

à l'intervalle défini par `ProxyDataFrequency`. Le proxy répond avec sa version actuelle et tous les résultats de tâches en attente (y compris les horodatages). Le serveur accuse ensuite réception en renvoyant une `response` et peut inclure de nouvelles tâches à exécuter par le proxy.

name	value type	description
server→proxy:		
request	*string*	'proxy tasks'
proxy→server:		
version	*string*	Version du proxy (<major>.<minor>.<build>).
clock	*number*	*(optional)* Horodatage du transfert de données (secondes).
ns	*number*	*(optional)* Horodatage du transfert de données (nanosecondes).
server→proxy:		
response	*string*	Informations de réussite de la requête ('success' or 'failed').

Exemple :

server→proxy :

```
{
  "request":"proxy tasks"
}
```

proxy→server :

```
{
  "version":"7.0.0",
  "clock":1721059872,
  "ns":848141522
}
```

server→proxy :

```
{
  "response":"success"
}
```

Proxy actif

Requête de configuration

La requête `proxy config` est envoyée par le proxy actif pour obtenir les données de configuration du proxy. Cette requête est envoyée toutes les `ProxyConfigFrequency` secondes (paramètre de configuration du proxy).

name	value type	description
proxy→server:		
request	*string*	'proxy config'
host	*string *	Nom du proxy.
version	*string*	Version du proxy (<major>.<minor>.<build>).
session	*string*	Jeton de session de configuration du proxy.
config_revision	*number*	Révision de la configuration du proxy.
hostmap_revision	*number*	Révision actuelle du mappage hôte-proxy.
server→proxy:		
full_sync	*number*	1 - si les données de configuration complètes sont envoyées, absent dans le cas contraire (facultatif).
data	*array*	Objet contenant les données des tables. Absent si la configuration n'a pas été modifiée (facultatif).
<table>	*object*	Un ou plusieurs objets contenant les données de <table> (facultatif, selon les modifications).
fields	*array*	Tableau des noms de champs.
-	*string*	Nom du champ.
data	*array*	Tableau de lignes.
-	*array*	Tableau de colonnes.
-	*string*,*number*	Valeur de colonne dont le type dépend du type de colonne dans le schéma de la base de données.

name	value type	description
macro.secrets	*object*	Informations sur les macros secrètes, absentes s'il n'y a aucune modification des macros du coffre (facultatif).
proxy_group	*string*	Nom du groupe de proxy auquel appartient le proxy.
config_revision	*number*	Révision du cache de configuration - envoyée avec les données de configuration (facultatif).
del_hostids	*array*	Tableau des identifiants d'hôtes supprimés (facultatif).
-	*number*	Identifiant de l'hôte.
del_macro_hostids	*array*	Tableau des identifiants d'hôtes dont toutes les macros ont été supprimées (facultatif).
-	*number*	Identifiant de l'hôte.

Exemple :

proxy→server:

```
{
  "request": "proxy config",
  "host": "Zabbix proxy",
  "version":"8.0.0",
  "session": "fd59a09ff4e9d1fb447de1f04599bcf6",
  "config_revision": 0
}
```

server→proxy:

```
{
"full_sync": 1,
"data": {
"hosts": {
"fields": ["hostid", "host", "status", "ipmi_authtype", "ipmi_privilege", "ipmi_username", "ipmi_password"
"data": [
[10084, "Zabbix server", 0, -1, 2, "", "", "Zabbix server", 1, 1, "", "", "", ""]
]
},
"interface": {
"fields": ["interfaceid", "hostid", "main", "type", "useip", "ip", "dns", "port", "available"],
"data": [
[1, 10084, 1, 1, 1, "127.0.0.1", "", "10053", 1]
]
},
"interface_snmp": {
"fields": ["interfaceid", "version", "bulk", "community", "securityname", "securitylevel", "authpassphrase
"data": []
},
"host_inventory": {
"fields": ["hostid", "type", "type_full", "name", "alias", "os", "os_full", "os_short", "serialno_a", "ser
"data": [
[10084, "", "", ..., "56.95387", "24.22067", ..., ""]
]
},
"items": {
"fields": ["itemid", "type", "snmp_oid", "hostid", "key_", "delay", "history", "status", "value_type", "tr
"data": [
[44161, 7, "", 10084, "agent.hostmetadata", "10s", "90d", 0, 1, "", "", "", "", 0, "", "", "", "", 0, null
[44162, 0, "", 10084, "agent.ping", "10s", "90d", 0, 3, "", "", "", "", 0, "", "", "", "", 0, 1, 0, "", nu
]
},
"item_rtdata": {
"fields": ["itemid", "lastlogsize", "mtime"],
"data": [
[44161, 0, 0],
```

```
[44162, 0, 0]
]
},
"item_preproc": {
"fields": ["item_preprocid", "itemid", "step", "type", "params", "error_handler", "error_handler_params"],
"data": []
},
"item_parameter": {
"fields": ["item_parameterid", "itemid", "name", "value"],
"data": []
},
"globalmacro": {
"fields": ["globalmacroid", "macro", "value", "type"],
"data": [
[2, "{$SNMP_COMMUNITY}", "public", 0]
]
},
"hosts_templates": {
"fields": ["hosttemplateid", "hostid", "templateid", "link_type"],
"data": []
},
"hostmacro": {
"fields": ["hostmacroid", "hostid", "macro", "value", "type", "automatic"],
"data": [
[5676, 10084, "{$M}", "AppID=zabbix_server&Query=Safe=passwordSafe;Object=zabbix:Content", 2, 0]
]
},
"drules": {
"fields": ["druleid", "name", "iprange", "delay"],
"data": [
[2, "Local network", "127.0.0.1", "10s"]
]
},
"dchecks": {
"fields": ["dcheckid", "druleid", "type", "key_", "snmp_community", "ports", "snmpv3_securityname", "snmpv
"data": [
[2, 2, 9, "system.uname", "", "10052", "", 0, "", "", 0, 0, 0, "", 1, 0]
]
},
"regexps": {
"fields": ["regexpid", "name"],
"data": [
[1, "File systems for discovery"],
[2, "Network interfaces for discovery"],
[3, "Storage devices for SNMP discovery"],
[4, "Windows service names for discovery"],
[5, "Windows service startup states for discovery"]
]
},
"expressions": {
"fields": ["expressionid", "regexpid", "expression", "expression_type", "exp_delimiter", "case_sensitive"]
"data": [
[1, 1, "^(btrfs|ext2|ext3|ext4|reiser|xfs|ffs|ufs|jfs|jfs2|vxfs|hfs|apfs|refs|ntfs|fat32|zfs)$", 3, ",", 0
[3, 3, "^(Physical memory|Virtual memory|Memory buffers|Cached memory|Swap space)$", 4, ",", 1],
[5, 4, "^(MMCSS|gupdate|SysmonLog|clr_optimization_v2.0.50727_32|clr_optimization_v4.0.30319_32)$", 4, ","
[6, 5, "^(automatic|automatic delayed)$", 3, ",", 1],
[7, 2, "^Software Loopback Interface", 4, ",", 1],
[8, 2, "^(In)?[Ll]oop[Bb]ack[0-9._]*$", 4, ",", 1],
[9, 2, "^NULL[0-9.]*$", 4, ",", 1],
[10, 2, "^[Ll]o[0-9.]*$", 4, ",", 1],
[11, 2, "^[Ss]ystem$", 4, ",", 1],
[12, 2, "^Nu[0-9.]*$", 4, ",", 1]
```

```
]
},
"settings": {
"fields": ["name", "type", "value_str", "value_int"],
"data": [
["autoreg_tls_accept", 2, "", 1],
["hk_history_global", 2, "", 0],
["snmptrap_logging", 2, "", 1],
["proxy_secrets_provider", 2, "", 0],
["hk_history", 1, "31d", 0],
["timeout_db_monitor", 1, "3s", 0],
["timeout_external_check", 1, "3s", 0],
["timeout_http_agent", 1, "3s", 0],
["timeout_simple_check", 1, "3s", 0],
["timeout_snmp_agent", 1, "3s", 0],
["timeout_ssh_agent", 1, "3s", 0],
["timeout_telnet_agent", 1, "3s", 0],
["timeout_zabbix_agent", 1, "3s", 0],
["timeout_browser", 1, "30s", 0]
]
},
"httptest": {
"fields": ["httptestid", "name", "delay", "agent", "authentication", "http_user", "http_password", "hostid
"data": []
},
"httptestitem": {
"fields": ["httptestitemid", "httptestid", "itemid", "type"],
"data": []
},
"httptest_field": {
"fields": ["httptest_fieldid", "httptestid", "type", "name", "value"],
"data": []
},
"httpstep": {
"fields": ["httpstepid", "httptestid", "name", "no", "url", "timeout", "posts", "required", "status_codes"
"data": []
},
"httpstepitem": {
"fields": ["httpstepitemid", "httpstepid", "itemid", "type"],
"data": []
},
"httpstep_field": {
"fields": ["httpstep_fieldid", "httpstepid", "type", "name", "value"],
"data": []
},
"config_autoreg_tls": {
"fields": ["autoreg_tlsid", "tls_psk_identity", "tls_psk"],
"data": [
[1, "", ""]
]
}
},
"macro.secrets": {
"AppID=zabbix_server&Query=Safe=passwordSafe;Object=zabbix": {
"Content": "738"
}
},
"config_revision": 2
}
```


Demande de données du proxy

La requête `proxy data` est envoyée par le proxy pour fournir les données de disponibilité des interfaces d'hôte, d'historique, de

découverte et d'autodétection. Cette requête est envoyée toutes les `DataSenderFrequency` secondes (paramètre de configuration du proxy). Notez qu'un proxy actif interrogera toujours le serveur Zabbix chaque seconde pour les tâches de commande distante (avec une requête `proxy data` vide).

name	value type	description
proxy→server:		
request	*string*	'proxy data'
host	*string*	Nom du proxy.
session	*string*	Jeton de session de données.
interface availability	*array*	*(optional)* Tableau d'objets de disponibilité des interfaces.
interfaceid	*number*	Identifiant de l'interface.
available	*number*	Disponibilité de l'interface : **0**, *INTERFACE_AVAILABLE_UNKNOWN* - inconnue **1**, *INTERFACE_AVAILABLE_TRUE* - disponible **2**, *INTERFACE_AVAILABLE_FALSE* - indisponible
error	*string*	Message d'erreur de l'interface ou chaîne vide.
history data	*array*	*(optional)* Tableau d'objets de données d'historique.
itemid	*number*	Identifiant de l'élément.
clock	*number*	Horodatage de la valeur de l'élément (secondes).
ns	*number*	Horodatage de la valeur de l'élément (nanosecondes).
value	*string*	*(optional)* Valeur de l'élément.
id	*number*	Identifiant de la valeur (compteur croissant, unique au sein d'une session de données).
timestamp	*number*	*(optional)* Horodatage des éléments de type journal.
source	*string*	*(optional)* Valeur de source de l'élément eventlog.
severity	*number*	*(optional)* Valeur de gravité de l'élément eventlog.
eventid	*number*	*(optional)* Valeur eventid de l'élément eventlog.
state	*string*	*(optional)* État de l'élément : **0**, *ITEM_STATE_NORMAL* **1**, *ITEM_STATE_NOTSUPPORTED*
lastlogsize	*number*	*(optional)* Dernière taille de journal des éléments de type journal.
mtime	*number*	*(optional)* Heure de modification des éléments de type journal.
discovery data	*array*	*(optional)* Tableau d'objets de données de découverte.
clock	*number*	Horodatage des données de découverte.
druleid	*number*	Identifiant de la règle de découverte.
dcheckid	*number*	Identifiant du contrôle de découverte ou null pour les données de règle de découverte.
type	*number*	Type de contrôle de découverte : **-1** données de règle de découverte **0**, *SVC_SSH* - contrôle du service SSH **1**, *SVC_LDAP* - contrôle du service LDAP **2**, *SVC_SMTP* - contrôle du service SMTP **3**, *SVC_FTP* - contrôle du service FTP **4**, *SVC_HTTP* - contrôle du service HTTP **5**, *SVC_POP* - contrôle du service POP **6**, *SVC_NNTP* - contrôle du service NNTP **7**, *SVC_IMAP* - contrôle du service IMAP **8**, *SVC_TCP* - contrôle de disponibilité du port TCP **9**, *SVC_AGENT* - agent Zabbix **10**, *SVC_SNMPv1* - agent SNMPv1 **11**, *SVC_SNMPv2* - agent SNMPv2 **12**, *SVC_ICMPPING* - ping ICMP **13**, *SVC_SNMPv3* - agent SNMPv3 **14**, *SVC_HTTPS* - contrôle du service HTTPS **15**, *SVC_TELNET* - contrôle de disponibilité Telnet

name	value type	description
ip	*string*	Adresse IP de l'hôte.
dns	*string*	Nom DNS de l'hôte.
port	*number*	*(optional)* Numéro de port du service.
key_	*string*	*(optional)* Clé de l'élément pour le contrôle de découverte de type **9** *SVC_AGENT*
value	*string*	*(optional)* Valeur reçue du service, peut être vide pour la plupart des services.
status	*number*	*(optional)* État du service : **0**, *DOBJECT_STATUS_UP* - service actif **1**, *DOBJECT_STATUS_DOWN* - service inactif
autoregistration	*array*	*(optional)* Tableau d'objets de données d'autodétection.
clock	*number*	Horodatage des données d'autodétection.
host	*string*	Nom de l'hôte.
ip	*string*	*(optional)* Adresse IP de l'hôte.
dns	*string*	*(optional)* Nom DNS résolu à partir de l'adresse IP.
port	*string*	*(optional)* Port de l'hôte.
host_metadata	*string*	*(optional)* Métadonnées de l'hôte envoyées par l'agent (basées sur le paramètre de configuration de l'agent HostMetadata ou HostMetadataItem).
tasks	*array*	*(optional)* Tableau de tâches.
type	*number*	Type de tâche : **0**, *ZBX_TM_TASK_PROCESS_REMOTE_COMMAND_RESULT* - résultat de commande distante
status	*number*	État d'exécution de la commande distante : **0**, *ZBX_TM_REMOTE_COMMAND_COMPLETED* - commande distante exécutée avec succès **1**, *ZBX_TM_REMOTE_COMMAND_FAILED* - échec de la commande distante
error	*string*	*(optional)* Message d'erreur.
parent_taskid	*number*	ID de la tâche parente.
more	*number*	*(optional)* 1 - il reste d'autres données d'historique à envoyer
clock	*number*	*(optional)* Horodatage du transfert de données (secondes).
ns	*number*	*(optional)* Horodatage du transfert de données (nanosecondes).
version	*string*	Version du proxy (<major>.<minor>.<build>).
server→proxy:		
response	*string*	Information de réussite de la requête ('success' ou 'failed').
upload	*string*	Contrôle de l'envoi des données historiques (historique, autodétection, disponibilité des hôtes, découverte réseau). Valeurs possibles : **enabled** - fonctionnement normal **disabled** - le serveur n'accepte pas les données (éventuellement en raison d'un dépassement du cache interne)
tasks	*array*	*(optional)* Tableau de tâches.
type	*number*	Type de tâche : **1**, *ZBX_TM_TASK_PROCESS_REMOTE_COMMAND* - commande distante
clock	*number*	Heure de création de la tâche.
ttl	*number*	Temps en secondes après lequel la tâche expire.
commandtype	*number*	Type de commande distante : **0**, *ZBX_SCRIPT_TYPE_CUSTOM_SCRIPT* - utiliser un script personnalisé **1**, *ZBX_SCRIPT_TYPE_IPMI* - utiliser IPMI **2**, *ZBX_SCRIPT_TYPE_SSH* - utiliser SSH **3**, *ZBX_SCRIPT_TYPE_TELNET* - utiliser Telnet **4**, *ZBX_SCRIPT_TYPE_GLOBAL_SCRIPT* - utiliser un script global (actuellement fonctionnellement équivalent à un script personnalisé)
command	*string*	Commande distante à exécuter.

name	value type	description
execute_on	*number*	Cible d'exécution pour les scripts personnalisés : **0**, *ZBX_SCRIPT_EXECUTE_ON_AGENT* - exécuter le script sur l'agent **1**, *ZBX_SCRIPT_EXECUTE_ON_SERVER* - exécuter le script sur le serveur **2**, *ZBX_SCRIPT_EXECUTE_ON_PROXY* - exécuter le script sur le proxy
port	*number*	*(optional)* Port pour les commandes Telnet et SSH.
authtype	*number*	*(optional)* Type d'authentification pour les commandes SSH.
username	*string*	*(optional)* Nom d'utilisateur pour les commandes Telnet et SSH.
password	*string*	*(optional)* Mot de passe pour les commandes Telnet et SSH.
publickey	*string*	*(optional)* Clé publique pour les commandes SSH.
privatekey	*string*	*(optional)* Clé privée pour les commandes SSH.
parent_taskid	*number*	ID de la tâche parente.
hostid	*number*	ID de l'hôte cible.

Exemple :

proxy→server :


```
{
"request": "proxy data",
"host": "Zabbix proxy",
"session": "818cdd1b537bdc5e50c09ed4969235b6",
"interface availability": [{
"interfaceid": 1,
"available": 1,
"error": ""
}],
"history data": [{
"id": 1114,
"itemid": 44162,
"clock": 1665730632,
"ns": 798953105,
"value": "1"
}, {
"id": 1115,
"itemid": 44161,
"clock": 1665730633,
"ns": 811684663,
"value": "58"
}],
"auto registration": [{
"clock": 1665730633,
"host": "Zabbix server",
"ip": "127.0.0.1",
"dns": "localhost",
"port": "10053",
"host_metadata": "58",
"tls_accepted": 1
}],
"discovery data": [{
"clock": 1665732232,
"drule": 2,
"dcheck": 2,
"ip": "127.0.0.1",
"dns": "localhost",
"port": 10052,
"status": 1
}, {
"clock": 1665732232,
"drule": 2,
"dcheck": null,
"ip": "127.0.0.1",
```

```
"dns": "localhost",
"status": 1
}],
"host data": [{
"hostid": 10084,
"active_status": 1
}],
"tasks": [{
"type": 3,
"clock": 1665730985,
"ttl": 0,
"status": -1,
"info": "Remote commands are not enabled",
"parent_taskid": 3
}],
"version": "8.0.0",
"clock": 1665730643,
"ns": 65389964
}
```


server→proxy :


```
{
"upload": "enabled",
"response": "success",
"tasks": [{
"type": 2,
"clock": 1665730986,
"ttl": 600,
"commandtype": 0,
"command": "ping -c 3 127.0.0.1; case $? in [01]) true;; *) false;; esac",
"execute_on": 2,
"port": 0,
"authtype": 0,
"username": "",
"password": "",
"publickey": "",
"privatekey": "",
"alertid": 0,
"parent_taskid": 4,
"hostid": 10084
}]
}
```


Paramètres de configuration: proxy Zabbix

Vue d'ensemble

Les paramètres de configuration permettent de personnaliser le comportement du proxy Zabbix.

Indiquez les valeurs des paramètres dans le fichier de configuration du proxy Zabbix (`zabbix_proxy.conf`). Tous les paramètres sont **facultatifs**, sauf indication explicite contraire.

Cette page est structurée comme suit :

- Vue d'ensemble des paramètres (pour afficher tous les détails d'un paramètre, cliquez sur son nom)
- Détails des paramètres

Paramètre	Description
AllowRoot	Autoriser l'exécution du proxy en tant que `root`.
AllowUnsupportedDBVersions	Autoriser le proxy à fonctionner avec des versions de base de données non prises en charge.
APMListenIP	Indiquer l'adresse IP sur laquelle le collecteur APM doit écouter les données de télémétrie entrantes.
APMListenPort	Indiquer le port sur lequel le collecteur APM doit écouter les données de télémétrie entrantes.

Paramètre	Description
APMTLSCAFile	Indiquer le chemin complet du fichier contenant le ou les certificats CA de niveau supérieur utilisés pour vérifier les certificats des applications ou des collecteurs OpenTelemetry qui envoient des données au collecteur APM.
APMTLSCertFile	Indiquer le chemin complet du fichier contenant le certificat ou la chaîne de certificats du collecteur APM.
APMTLSKeyFile	Indiquer le chemin complet du fichier contenant la clé privée du collecteur APM.
CacheSize	Définir la taille du cache de configuration.
ConfigFrequency	Ce paramètre est **obsolète** (utilisez ProxyConfigFrequency). Définir la fréquence de récupération des données de configuration depuis le serveur Zabbix.
DataSenderFrequency	Définir la fréquence d'envoi des données collectées au serveur Zabbix.
DBHost	Indiquer l'hôte de la base de données (ou le répertoire du socket).
DBName	Indiquer le nom de la base de données ou le chemin du fichier de base de données pour SQLite3.
DBPassword	Indiquer le mot de passe de la base de données.
DBPort	Indiquer le port de l'hôte de la base de données Zabbix.
DBSchema	Indiquer le nom du schéma de la base de données. Utilisé pour PostgreSQL.
DBSocket	Indiquer le chemin du fichier socket MySQL.
DBUser	Indiquer l'utilisateur de la base de données.
DBTLSConnect	Indiquer une valeur pour imposer une connexion TLS à la base de données.
DBTLSCAFile	Indiquer le chemin complet du fichier contenant le ou les certificats CA de niveau supérieur utilisés pour vérifier le certificat de la base de données.
DBTLSCertFile	Indiquer le chemin complet du fichier contenant le certificat du proxy Zabbix utilisé pour l'authentification auprès de la base de données.
DBTLSKeyFile	Indiquer le chemin complet du fichier contenant la clé privée utilisée pour l'authentification auprès de la base de données.
DBTLSCipher	Indiquer la liste des chiffrements de chiffrement que le proxy Zabbix autorise pour les protocoles TLS jusqu'à TLS v1.2 inclus. Pris en charge uniquement pour MySQL.
DBTLSCipher13	Indiquer la liste des suites de chiffrement que le proxy Zabbix autorise pour le protocole TLS v1.3. Prise en charge uniquement pour MySQL à partir de la version 8.0.16.
DebugLevel	Définir le niveau de débogage.
EnableRemoteCommands	Autoriser les commandes distantes provenant du serveur Zabbix.
ExternalScripts	Indiquer l'emplacement des scripts externes.
Fping6Location	Indiquer l'emplacement de fping6.
FpingLocation	Indiquer l'emplacement de fping.
HistoryCacheSize	Définir la taille du cache de l'historique.
HistoryIndexCacheSize	Définir la taille du cache des index de l'historique.
Hostname	Indiquer le nom du proxy.
HostnameItem	Indiquer l'élément utilisé pour définir Hostname (s'il n'est pas défini).
HousekeepingFrequency	Définir la fréquence de la procédure de maintenance (en heures).
Include	Indiquer un répertoire ou des fichiers individuels à inclure dans le fichier de configuration.
JavaGateway	Indiquer l'adresse IP (ou le nom d'hôte) du Java gateway Zabbix.
JavaGatewayPort	Indiquer le port du Java gateway Zabbix.
ListenBacklog	Définir le nombre maximal de connexions en attente dans la file TCP.
ListenIP	Indiquer l'adresse IP d'écoute du trapper.
ListenPort	Indiquer le port d'écoute du trapper.
LoadModule	Indiquer le module à charger au démarrage du proxy.
LoadModulePath	Indiquer le chemin complet de l'emplacement des modules du proxy.
LogFile	Indiquer le fichier journal du proxy.
LogFileSize	Définir la taille maximale du fichier journal.
LogRemoteCommands	Activer la journalisation des commandes shell exécutées en tant qu'avertissements.
LogSlowQueries	Définir la durée maximale d'exécution d'une requête de base de données avant sa journalisation (en millisecondes).
LogType	Indiquer le type de sortie du journal.
MaxConcurrentChecksPerPoller	Définir le nombre maximal de vérifications asynchrones pouvant être exécutées simultanément par chaque poller HTTP agent, poller agent ou poller SNMP.
PidFile	Indiquer le fichier PID.
ProxyBufferMode	Indiquer le mécanisme de stockage des données d'historique, de découverte et d'enregistrement automatique (disque/mémoire/hybride).
ProxyConfigFrequency	Définir la fréquence de récupération des données de configuration depuis le serveur Zabbix, en secondes.
ProxyLocalBuffer	Définir la durée de conservation locale des données (en heures), même si elles ont déjà été synchronisées avec le serveur.

Paramètre	Description
ProxyMemoryBufferAge	Définir l'ancienneté maximale des données dans le tampon mémoire du proxy, en secondes.
ProxyMemoryBufferSize	Définir la taille du cache de mémoire partagée pour l'historique, la découverte et les données d'enregistrement automatique collectés.
ProxyMode	Définir le mode de fonctionnement du proxy (actif/passif).
ProxyOfflineBuffer	Définir la durée de conservation des données (en heures) en cas d'absence de connectivité avec le serveur Zabbix.
Server	Si ProxyMode est défini sur `active` : indiquer l'adresse IP ou le nom DNS du serveur Zabbix (address:port) ou du cluster (address:port;address2:port) depuis lequel récupérer les données de configuration et auquel envoyer les données. Si ProxyMode est défini sur `passive` : indiquer les adresses IP ou les noms DNS du serveur Zabbix.
SNMPTrapperFile	Indiquer le fichier temporaire utilisé pour transmettre les données du démon de traps SNMP au proxy.
SocketDir	Indiquer le répertoire de stockage des sockets IPC utilisés par les services Zabbix internes.
SourceIP	Indiquer l'adresse IP source.
SSHKeyLocation	Indiquer l'emplacement des clés publiques et privées pour les vérifications et actions SSH.
SSLCertLocation	Indiquer l'emplacement des fichiers de certificats SSL client pour l'authentification du client.
SSLKeyLocation	Indiquer l'emplacement des fichiers de clés privées SSL pour l'authentification du client.
SSLCALocation	Indiquer l'emplacement des fichiers d'autorité de certification (CA) pour la vérification du certificat du serveur SSL.
StartAgentPollers	Définir le nombre d'instances pré-forkées de pollers agent Zabbix asynchrones.
StartAPMCollector	Définir sur 1 pour démarrer le collecteur APM.
StartBrowserPollers	Définir le nombre d'instances pré-forkées de pollers d'éléments navigateur.
StartDBSyncers	Définir le nombre d'instances pré-forkées de synchroniseurs d'historique.
StartDiscoverers	Définir le nombre d'instances pré-forkées de processus de découverte.
StartHTTPAgentPollers	Définir le nombre d'instances pré-forkées de pollers HTTP agent asynchrones.
StartHTTPPollers	Définir le nombre d'instances pré-forkées de pollers HTTP.
StartIPMIPollers	Définir le nombre d'instances pré-forkées de pollers IPMI.
StartJavaPollers	Définir le nombre d'instances pré-forkées de pollers Java.
StartODBCPollers	Définir le nombre d'instances pré-forkées de pollers ODBC.
StartPingers	Définir le nombre d'instances pré-forkées de pingers ICMP.
StartPollersUnreachable	Définir le nombre d'instances pré-forkées de pollers pour les hôtes injoignables (y compris IPMI et Java).
StartPollers	Définir le nombre d'instances pré-forkées de pollers.
StartPreprocessors	Définir le nombre d'instances pré-démarrées de processus de prétraitement.
StartSNMPPollers	Définir le nombre d'instances pré-forkées de pollers SNMP asynchrones.
StartSNMPTrapper	Définir sur « 1 » pour démarrer un processus trapper SNMP.
StartTelemetryQueryPollers	Définir le nombre d'instances pré-forkées de pollers de requêtes de télémétrie asynchrones.
StartTrappers	Définir le nombre d'instances pré-forkées de trappers.
StartVMwareCollectors	Définir le nombre d'instances pré-forkées de collecteurs VMware.
StatsAllowedIP	Indiquer les adresses IP ou les noms DNS des instances Zabbix externes autorisées à demander des statistiques.
TelemetryProvider	Définir la source des données de télémétrie.
Timeout	Indiquer le délai d'attente (en secondes) pour établir une connexion et échanger des données avec le proxy Zabbix, l'agent, le service web et pour les vérifications SNMP (à l'exception des éléments SNMP `walk[OID]` et `get[OID]`).
TLSAccept	Indiquer le niveau de chiffrement requis pour les connexions entrantes provenant du serveur Zabbix.
TLSCAFile	Indiquer le chemin complet du fichier contenant le ou les certificats CA de niveau supérieur utilisés pour vérifier les certificats des pairs, dans le cadre des communications chiffrées entre les composants Zabbix.
TLSCertFile	Indiquer le chemin complet du fichier contenant le certificat du serveur ou la chaîne de certificats, utilisé pour les communications chiffrées entre les composants Zabbix.
TLSCipherAll	Indiquer la chaîne de priorité GnuTLS ou la chaîne de chiffrement OpenSSL (TLS 1.2). Remplacer les critères de sélection par défaut des suites de chiffrement pour le chiffrement basé sur des certificats et sur des clés prépartagées (PSK).
TLSCipherAll13	Indiquer la chaîne de chiffrement pour OpenSSL 1.1.1 ou version ultérieure avec TLS 1.3. Remplacer les critères de sélection par défaut des suites de chiffrement pour le chiffrement basé sur des certificats et sur des clés prépartagées (PSK).
TLSCipherCert	Indiquer la chaîne de priorité GnuTLS ou la chaîne de chiffrement OpenSSL (TLS 1.2). Remplacer les critères de sélection par défaut des suites de chiffrement pour le chiffrement basé sur des certificats.

Paramètre	Description
TLSCipherCert13	Indiquer la chaîne de chiffrement pour OpenSSL 1.1.1 ou version ultérieure avec TLS 1.3. Remplacer les critères de sélection par défaut des suites de chiffrement pour le chiffrement basé sur des certificats.
TLSCipherPSK	Indiquer la chaîne de priorité GnuTLS ou la chaîne de chiffrement OpenSSL (TLS 1.2). Remplacer les critères de sélection par défaut des suites de chiffrement pour le chiffrement basé sur des clés prépartagées (PSK).
TLSCipherPSK13	Indiquer la chaîne de chiffrement pour OpenSSL 1.1.1 ou version ultérieure avec TLS 1.3. Remplacer les critères de sélection par défaut des suites de chiffrement pour le chiffrement basé sur des clés prépartagées (PSK).
TLSConnect	Indiquer le niveau de chiffrement requis pour les connexions au serveur Zabbix.
TLSCRLFile	Indiquer le chemin complet du fichier contenant les certificats révoqués. Ce paramètre est utilisé pour les communications chiffrées entre les composants Zabbix.
TLSKeyFile	Indiquer le chemin complet du fichier contenant la clé privée du proxy, utilisée pour les communications chiffrées entre les composants Zabbix.
TLSListen	Exiger des connexions TLS uniquement sur le socket trapper.
TLSPSKFile	Indiquer le chemin complet du fichier contenant la clé prépartagée du proxy, utilisée pour les communications chiffrées avec le serveur Zabbix.
TLSPSKIdentity	Indiquer la chaîne d'identité de la clé prépartagée, utilisée pour les communications chiffrées avec le serveur Zabbix.
TLSServerCertIssuer	Indiquer l'émetteur de certificat du serveur autorisé.
TLSServerCertSubject	Indiquer le sujet de certificat du serveur autorisé.
TmpDir	Indiquer le répertoire temporaire.
TrapperTimeout	Indiquer le délai d'attente en secondes pour : - la récupération des données de configuration depuis le serveur Zabbix ; - l'exécution d'un script global ou d'une commande distante.
UnavailableDelay	Définir la fréquence de vérification de la disponibilité de l'hôte pendant la période d'indisponibilité.
UnreachableDelay	Définir la fréquence de vérification de la disponibilité de l'hôte pendant la période d'injoignabilité.
UnreachablePeriod	Définir après combien de secondes d'injoignabilité l'hôte doit être considéré comme indisponible.
User	Abandonner les privilèges pour utiliser un utilisateur spécifique existant sur le système.
Vault	Indiquer le fournisseur de coffre-fort.
VaultAppRoleID	Indiquer l'ID AppRole du coffre-fort HashiCorp.
VaultAppSecretID	Indiquer l'ID secret AppRole du coffre-fort HashiCorp.
VaultDBPath	Indiquer l'emplacement de récupération des identifiants de la base de données par clés.
VaultPrefix	Indiquer un préfixe personnalisé pour le chemin ou la requête du coffre-fort.
VaultTLSCertFile	Indiquer le nom du fichier de certificat SSL utilisé pour l'authentification du client.
VaultTLSKeyFile	Indiquer le nom du fichier de clé privée SSL utilisé pour l'authentification du client.
VaultToken	Indiquer le jeton d'authentification du coffre-fort HashiCorp.
VaultURL	Indiquer l'URL du serveur de coffre-fort.
VMwareCacheSize	Définir la taille de la mémoire partagée utilisée pour stocker les données VMware.
VMwareFrequency	Définir la fréquence de collecte des données depuis un service VMware donné, en secondes.
VMwarePerfFrequency	Définir la fréquence de récupération des statistiques des compteurs de performance depuis un service VMware donné, en secondes.
VMwareTimeout	Définir le nombre maximal de secondes pendant lesquelles un `vmware collector` attend une réponse du service VMware.
WebDriverURL	Indiquer l'URL de l'interface WebDriver.

Notez que :

- Les valeurs par défaut correspondent aux valeurs par défaut des démons, et non aux valeurs des fichiers de configuration fournis.
- Les valeurs prennent en charge les variables d'environnement.
- Zabbix prend uniquement en charge les fichiers de configuration encodés en UTF-8 sans BOM.
- Les commentaires commençant par # sont pris en charge uniquement au début de la ligne.

Détails du paramètre

AllowRoot

Autoriser le proxy à s'exécuter en tant que `root`. Si cette option est désactivée et que le proxy est démarré par `root`, le proxy tentera à la place de basculer vers l'utilisateur `zabbix`. N'a aucun effet s'il est démarré sous un utilisateur standard.

Par défaut: 0
 Valeurs: 0 - ne pas autoriser; 1 - autoriser

AllowUnsupportedDBVersions

Autoriser le proxy à fonctionner avec des versions de base de données non prises en charge.

Par défaut: 0
 Valeurs: 0 - ne pas autoriser; 1 - autoriser

APMListenIP

Spécifiez l'adresse IP sur laquelle le collecteur APM doit écouter les données de télémétrie entrantes.
 Si elle n'est pas spécifiée, le collecteur APM écoutera sur toutes les interfaces réseau.

Valeur par défaut : `0.0.0.0`

APMListenPort

Spécifiez le port sur lequel le collecteur APM doit écouter les données de télémétrie entrantes.
 Le port par défaut est le port standard OpenTelemetry (OTLP/gRPC), utilisé par défaut par les SDK et les collecteurs OpenTelemetry.

Valeur par défaut : 4317
 Plage : 1024-32767

APMTLSCAFile

Spécifiez le chemin complet du fichier contenant les certificats des autorités de certification racines permettant de vérifier les certificats des applications ou des collecteurs OpenTelemetry qui envoient des données au collecteur APM.
 Si ce paramètre est défini, le collecteur APM accepte uniquement les connexions chiffrées ; sinon, il accepte uniquement les connexions non chiffrées.

APMTLSCertFile

Indiquez le chemin complet du fichier contenant le certificat ou la chaîne de certificats du collecteur APM.
 Ce paramètre doit être défini avec `APMTLSKeyFile`, et uniquement si `APMTLSCAFile` est défini.

APMTLSKeyFile

Spécifiez le chemin complet du fichier contenant la clé privée du collecteur APM.
 Ce paramètre doit être défini avec `APMTLSCertFile`, et uniquement si `APMTLSCAFile` est défini.

CacheSize

Définissez la taille du cache de configuration, en octets. Taille de la mémoire partagée utilisée pour stocker les données des hôtes et des éléments. Par défaut: 32M
 Plage: 128K-64G

ConfigFrequency

Ce paramètre est **obsolète** (utilisez plutôt `ProxyConfigFrequency`).
Définit la fréquence de récupération des données de configuration depuis le serveur Zabbix (en secondes).
Paramètre du proxy actif. Ignoré pour les proxies passifs (voir le paramètre `ProxyMode`).

Par défaut: 3600
 Plage: 1-604800

DataSenderFrequency

Définit la fréquence d'envoi des données collectées du proxy vers le serveur Zabbix. Notez qu'un proxy actif continuera à interroger le serveur Zabbix chaque seconde pour les tâches de commande à distance.
Paramètre du proxy actif. Ignoré pour les proxies passifs (voir le paramètre `ProxyMode`).

Par défaut: 1
 Plage: 1-3600

DBHost

Spécifiez l'hôte de base de données (ou le répertoire du socket).

Pour MySQL :

- `localhost` ou une chaîne vide utilise le socket UNIX par défaut.

Pour PostgreSQL :

- `localhost` est résolu via DNS (généralement vers `127.0.0.1`).
- Une chaîne vide utilise le socket UNIX par défaut.
- Un chemin (par exemple, `/var/run/pgbouncer`) entraîne l'utilisation du socket UNIX à ce chemin.
- Une liste séparée par des virgules peut contenir plusieurs valeurs `host:port`. Chaque hôte est essayé dans l'ordre jusqu'à ce qu'une connexion en lecture-écriture soit établie. Exemple : `DBHost=localhost:5431,127.0.0.1:20051,zabbix.domain,[:`

Par défaut : `localhost`

DBName

Indiquez le nom de la base de données ou le chemin d'accès au fichier de base de données pour SQLite3 (l'architecture multiprocessus de Zabbix ne permet pas d'utiliser une base de données en mémoire, par exemple `:memory:`, `file::memory:?cache=shared` ou `file:memdb1?mode=memory&cache=shared`).
Avertissement : n'essayez pas d'utiliser la même base de données que celle utilisée par le serveur Zabbix.

Obligatoire : Oui

DBPassword

Spécifiez le mot de passe de la base de données. Commentez cette ligne si aucun mot de passe n'est utilisé. Ignoré pour SQLite.

DBPort

Spécifiez le port de l'hôte de la base de données Zabbix (voir `DBHost`). Ignoré pour SQLite.
 DBSocket et DBPort s'excluent mutuellement dans la configuration du proxy. Spécifiez-en un seul, ou laissez les deux non définis.

Valeur par défaut pour MySQL: 3306

Valeur par défaut pour PostgreSQL: 5432

Plage: 1024-65535

DBSchema

Spécifiez le nom du schéma de base de données. Utilisé pour PostgreSQL.

DBSocket

Spécifiez le chemin vers le fichier socket MySQL.
DBSocket et DBPort s'excluent mutuellement dans la configuration du proxy. Spécifiez-en un seul, ou laissez les deux non définis.

DBUser

Spécifiez l'utilisateur de la base de données. Ignoré pour SQLite.

DBTLSConnect

Spécifiez une valeur pour imposer une connexion TLS à la base de données : - *required* - se connecter en utilisant TLS ; - *verify_ca* - se connecter en utilisant TLS et vérifier le certificat ; - *verify_full* - se connecter en utilisant TLS, vérifier le certificat et vérifier que l'identité de la base de données spécifiée par DBHost correspond à son certificat
Sur MySQL à partir de la version 5.7.11 et sur PostgreSQL, les valeurs suivantes sont prises en charge : "required", "verify", "verify_full".
Sur MariaDB à partir de la version 10.2.6, les valeurs "required" et "verify_full" sont prises en charge.
Par défaut, aucune option n'est définie et le comportement dépend de la configuration de la base de données.

DBTLSCAFile

Spécifiez le chemin complet du fichier contenant le ou les certificats CA de niveau supérieur pour la vérification du certificat de la base de données.

Obligatoire: non (oui, si DBTLSConnect est défini sur *verify_ca* ou *verify_full*)

DBTLSCertFile

Spécifiez le chemin complet du fichier contenant le certificat du proxy Zabbix pour l'authentification à la base de données.

DBTLSKeyFile

Spécifiez le chemin d'accès complet du fichier contenant la clé privée utilisée pour l'authentification à la base de données.

DBTLSCipher

Spécifiez la liste des chiffrements de chiffrement que le proxy Zabbix autorise pour les protocoles TLS jusqu'à TLS v1.2. Pris en charge uniquement pour MySQL.

DBTLSCipher13

Spécifiez la liste des suites de chiffrement que le proxy Zabbix autorise pour le protocole TLS v1.3. Pris en charge uniquement pour MySQL, à partir de la version 8.0.16.

DebugLevel

Définissez le niveau de débogage : - *0* - informations de base sur le démarrage et l'arrêt des processus Zabbix ; - *1* - informations critiques ; - *2* - informations d'erreur ; - *3* - avertissements ; - *4* - pour le débogage (génère beaucoup d'informations) ; - *5* - débogage étendu (génère encore plus d'informations).

Par défaut : 3
 Plage : 0-5

EnableRemoteCommands

Autoriser les commandes à distance depuis le serveur Zabbix.

Par défaut: 0
 Valeurs: 0 - ne pas autoriser; 1 - autoriser

ExternalScripts

Spécifiez l'emplacement du script externe (cela dépend de la variable d'installation au moment de la compilation `datadir`).

Par défaut: `/usr/local/share/zabbix/externalscripts`

Fping6Location

Spécifiez l'emplacement de fping6. Assurez-vous que le binaire fping6 appartient à root et que le bit SUID est défini. Laissez vide (`Fping6Location=`) si votre utilitaire `fping` est capable de traiter les adresses IPv6.

Par défaut: `/usr/sbin/fping6`

FpingLocation

Spécifiez l'emplacement de fping. Assurez-vous que le binaire fping appartient à root et que l'attribut SUID est défini.

Par défaut : `/usr/sbin/fping`

HistoryCacheSize

Définissez la taille du cache d'historique, en octets. La taille de la mémoire partagée utilisée pour stocker les données d'historique.

Par défaut: 16M
 Plage: 128K-16G

HistoryIndexCacheSize

Définissez la taille du cache d'index d'historique, en octets. Taille de la mémoire partagée utilisée pour l'indexation des données d'historique stockées dans le cache d'historique. La taille du cache d'index nécessite environ 100 octets pour mettre en cache un élément.

Par défaut: 4M
 Plage: 128K-16G

Nom d'hôte

Spécifiez un nom de proxy unique, sensible à la casse. Assurez-vous que le nom du proxy est connu du serveur.
Caractères autorisés: alphanumériques, '.', ' ', '_' et '-'. Longueur maximale: 128

Par défaut: défini par HostnameItem

HostnameItem

Spécifiez un élément pour définir `Hostname` (s'il n'est pas défini).
 Cet élément sera exécuté sur le proxy de la même manière que sur un agent.
 Ignoré si `Hostname` est défini.
Ne prend pas en charge les paramètres utilisateur, les compteurs de performance ni les alias ; prend en charge `system.run[]`.

Par défaut: `system.hostname`

HousekeepingFrequency

Définissez la fréquence de la procédure de housekeeping (en heures). Le housekeeping consiste à supprimer les informations obsolètes de la base de données.
Remarque : afin de réduire la charge au démarrage du proxy, le housekeeping est reporté de 30 minutes après le démarrage du proxy. Ainsi, si `HousekeepingFrequency` vaut 1, la toute première procédure de housekeeping après le démarrage du proxy s'exécutera après 30 minutes, puis se répétera ensuite toutes les heures.
Il est possible de désactiver le housekeeping automatique en définissant HousekeepingFrequency à 0. Dans ce cas, la procédure de housekeeping ne peut être lancée qu'à l'aide de l'option de contrôle à l'exécution `housekeeper_execute`.
Voir aussi les options et détails de runtime control et de la procédure de housekeeping.

Valeur par défaut : 1
 Plage : 0-24

Inclure

Spécifiez un répertoire ou des fichiers individuels à inclure dans le fichier de configuration.
 Lors de l'installation de Zabbix, ce paramètre est défini sur `/usr/local/etc`, sauf si vous le modifiez au moment de la compilation.

Règles générales pour le paramètre `Include` :

- Le chemin peut être relatif à l'emplacement du fichier `zabbix_proxy.conf`.
- Vous pouvez spécifier plusieurs paramètres `Include` dans un seul fichier de configuration.
- Zabbix traite le paramètre `Include` dès qu'il le lit, donc pensez à le placer à la fin du fichier de configuration.

Règles pour les fichiers inclus :

- Chaque fichier doit être lisible.

- Chaque fichier doit avoir une syntaxe correcte. Sinon, le composant ne démarrera pas.

Règles pour l'inclusion de répertoires ou de modèles de fichiers :

- Ne supposez aucun ordre d'inclusion (par exemple, les fichiers ne sont pas inclus par ordre alphabétique).
- Ne définissez pas le même paramètre dans plusieurs fichiers inclus (par exemple, pour remplacer un réglage général par un réglage spécifique), car vous ne pouvez pas prévoir quelle valeur sera utilisée.
- Faites attention aux copies de sauvegarde que certains éditeurs de texte créent automatiquement (par exemple, la modification de `include/my_specific.conf` peut créer `include/my_specific.conf.BAK`), car les deux fichiers seront inclus.

Pour inclure uniquement les fichiers pertinents d'un répertoire, utilisez le caractère générique (`*`) pour la correspondance de motifs.

Exemple :

```
Include=/absolute/path/to/config/files/*.conf
```

JavaGateway

Spécifiez l'adresse IP ou le nom d'hôte du Java gateway de Zabbix. Obligatoire uniquement si les collecteurs Java sont démarrés.

JavaGatewayPort

Spécifiez le port du Java gateway Zabbix.

Par défaut: 10052
 Plage: 1024-32767

ListenBacklog

Définit le nombre maximal de connexions en attente dans la file TCP.
La valeur par défaut est une constante codée en dur, qui dépend du système.
La valeur maximale prise en charge dépend également du système ; des valeurs trop élevées peuvent être tronquées silencieusement à la « valeur maximale spécifiée par l'implémentation ».

Par défaut: `SOMAXCONN`
 Plage: 0 - INT_MAX

ListenIP

Spécifiez l'adresse IP d'écoute du trapper. Une liste séparée par des virgules.
Le trapper écoutera sur toutes les interfaces réseau si ce paramètre n'est pas spécifié.

Par défaut: `0.0.0.0`

ListenPort

Spécifiez le port d'écoute du trapper.

Par défaut: 10051
 Plage: 1024-32767

LoadModule

Spécifiez le module à charger au démarrage du proxy. Les modules servent à étendre les fonctionnalités du proxy. Le module doit se trouver dans le répertoire spécifié par `LoadModulePath`, ou le chemin doit précéder le nom du module. Si le chemin précédent est absolu (commence par `/`), `LoadModulePath` est ignoré.
 Formats : - `LoadModule=<module.so>` - `LoadModule=<path/module.so>` - `LoadModule=</abs_path/module.so>`

Plusieurs paramètres `LoadModule` peuvent être spécifiés.

LoadModulePath

Le chemin complet vers l'emplacement des modules du proxy. La valeur par défaut dépend des options de compilation.

LogFile

Spécifiez le fichier journal du proxy.

Obligatoire: Oui, si `LogType` est défini sur `file`; sinon non

LogFileSize

Définissez la taille maximale du fichier journal (en Mo).
0 - désactive la rotation automatique des journaux.
Remarque : si la limite de taille du fichier journal est atteinte et que la rotation du fichier échoue, quelle qu'en soit la raison, le fichier journal existant est tronqué et recréé.

Par défaut : 1
 Plage : 0-1024

LogRemoteCommands

Active l'enregistrement des commandes shell exécutées en tant qu'avertissements.

Par défaut: 0
 Valeurs: 0 - désactivé, 1 - activé

LogSlowQueries

Définissez la durée maximale qu'une requête de base de données peut prendre avant d'être consignée dans les journaux (en millisecondes).
0 - ne pas consigner les requêtes lentes.
Cette option devient active à partir de `DebugLevel=3`.

Valeur par défaut: 0
 Plage: 0-3600000

LogType

Spécifiez le type de sortie du journal : - *file* - écrire le journal dans le fichier spécifié par le paramètre `LogFile` ; - *system* - écrire le journal dans syslog ; - *console* - écrire le journal vers la sortie standard.

Par défaut : `file`

MaxConcurrentChecksPerPoller

Définit le nombre maximal de vérifications asynchrones pouvant être exécutées simultanément par chaque poller d'agent HTTP, poller d'agent ou poller SNMP. Voir `StartHTTPAgentPollers`, `StartAgentPollers` et `StartSNMPPollers`.

Par défaut: 1000
 Plage: 1-1000

PidFile

Spécifiez le fichier PID.

Par défaut : `/tmp/zabbix_proxy.pid`

ProxyBufferMode

Spécifiez le mécanisme de stockage pour les données d'historique, de découverte et d'autorégistration : - *disk* - les données sont stockées dans la base de données et téléversées depuis la base de données ; - *memory* - les données sont stockées en mémoire et téléversées depuis la mémoire. Si le tampon manque de mémoire, les anciennes données seront supprimées. À l'arrêt, le tampon est supprimé. - *hybrid* - le tampon du proxy fonctionne normalement comme en mode mémoire jusqu'à ce qu'il manque de mémoire ou que l'enregistrement le plus ancien dépasse l'âge configuré. Si cela se produit, le tampon est vidé vers la base de données et fonctionne comme en mode disque jusqu'à ce que toutes les données aient été téléversées, puis il recommence à fonctionner en mémoire. À l'arrêt, le tampon mémoire est vidé vers la base de données.

Voir aussi : Tampon mémoire du proxy.

Par défaut : `disk`
 Valeurs : disk; memory; hybrid

ProxyConfigFrequency

Définit la fréquence de récupération des données de configuration depuis le serveur Zabbix, en secondes.
Paramètre du proxy actif. Ignoré pour les proxies passifs (voir le paramètre `ProxyMode`).

Par défaut: 10
 Plage: 1-604800

ProxyLocalBuffer

Définissez la durée de conservation des données localement (en heures), même si les données ont déjà été synchronisées avec le serveur..
Ce paramètre peut être utilisé si les données locales seront utilisées par des applications tierces.

Par défaut: 0
 Plage: 0-720

ProxyMemoryBufferAge

Définit l'âge maximal des données dans le tampon mémoire du proxy, en secondes.
 Lorsqu'il est activé (différent de zéro) et que les enregistrements dans le tampon mémoire du proxy sont plus anciens, cela force le tampon du proxy à passer en mode base de données jusqu'à ce que tous les enregistrements soient envoyés au serveur. Ce paramètre doit être inférieur ou égal au paramètre `ProxyOfflineBuffer`.

Par défaut: 0
 Plage: 0;600-864000

ProxyMemoryBufferSize

Définit la taille du cache en mémoire partagée pour les données d'historique, de découverte et d'enregistrement automatique collectées (en octets). Si elle est activée (valeur différente de zéro), le proxy conservera en mémoire les données d'historique, de découverte et d'enregistrement automatique, sauf si le cache est plein ou si les enregistrements stockés sont plus anciens que la valeur définie de `ProxyMemoryBufferAge`. Ce paramètre ne peut pas être utilisé avec le paramètre `ProxyLocalBuffer`.

Par défaut: 0
 Plage: 0;128K-2G

ProxyMode

Définissez le mode de fonctionnement du proxy.
0 - proxy en mode actif
1 - proxy en mode passif
Remarque : les données de configuration sensibles du proxy peuvent devenir accessibles à des tiers ayant accès au port trapper du serveur Zabbix lors de l'utilisation d'un proxy actif. Cela est possible car n'importe qui peut se faire passer pour un proxy actif et demander des données de configuration ; aucune authentification n'a lieu.

Par défaut : 0
 Plage : 0-1

ProxyOfflineBuffer

Définissez la durée de conservation des données (en heures) en cas d'absence de connectivité avec le serveur Zabbix.
Les données plus anciennes seront perdues.

Par défaut: 1
 Plage: 1-720

Serveur

Si ProxyMode est défini sur `active` : spécifiez l'adresse IP ou le nom DNS du serveur Zabbix (adresse:port) ou cluster (adresse:port;adresse2:port) à partir duquel récupérer les données de configuration et vers lequel envoyer les données.
Si le port n'est pas spécifié, le port par défaut est utilisé.
Les nœuds du cluster doivent être séparés par un point-virgule.

Si ProxyMode est défini sur `passive` : spécifiez les adresses IP, éventuellement en notation CIDR, ou les noms DNS du serveur Zabbix. Une liste séparée par des virgules.
 Les connexions entrantes ne seront acceptées que depuis les adresses listées ici. Si la prise en charge d'IPv6 est activée, alors `127.0.0.1`, `::127.0.0.1`, `::ffff:127.0.0.1` sont traités de la même manière.
`::/0` autorisera toute adresse IPv4 ou IPv6. `0.0.0.0/0` peut être utilisé pour autoriser toute adresse IPv4.

Exemple :

```
Server=127.0.0.1,192.168.1.0/24,::1,2001:db8::/32,zabbix.example.com
```

Obligatoire : oui

SNMPTrapperFile

Spécifie le fichier temporaire utilisé pour transmettre les données du démon de traps SNMP au proxy.
Doit être identique à celui défini dans `zabbix_trap_receiver.pl` ou dans le fichier de configuration SNMPTT.

Par défaut: `/tmp/zabbix_traps.tmp`

SocketDir

Spécifiez le répertoire de stockage des sockets IPC utilisés par les services internes de Zabbix.

Par défaut: `/tmp`

SourceIP

Spécifiez l'adresse IP source pour :

- les connexions sortantes vers le serveur Zabbix
- les connexions sans agent (VMware, SSH, JMX, SNMP, Telnet et vérifications simples)
- les connexions de l'agent HTTP
- les requêtes HTTP JavaScript de l'élément script
- les requêtes HTTP JavaScript de prétraitement
- les connexions au Vault

SSHKeyLocation

Spécifiez l'emplacement des clés publiques et privées pour les vérifications SSH et les actions.

SSLCertLocation

Indiquez l'emplacement des fichiers de certificat SSL-client pour l'authentification du client.
Ce paramètre est utilisé pour la surveillance web, la communication avec Vault et la source de données de télémétrie ClickHouse.

SSLKeyLocation

Spécifiez l'emplacement des fichiers de clé privée SSL pour l'authentification du client.
Ce paramètre est utilisé pour la supervision Web, la communication avec Vault et la source de données de télémétrie ClickHouse.

SSLCALocation

Indiquez l'emplacement des fichiers de l'autorité de certification (CA) utilisés pour la vérification du certificat du serveur SSL.
Notez que la valeur de ce paramètre sera définie comme option `CURLOPT_CAPATH` de libcurl. Pour les versions de libcurl antérieures à 7.42.0, ce paramètre n'a d'effet que si libcurl a été compilé pour utiliser OpenSSL. Pour plus d'informations, consultez la page web de cURL.
Ce paramètre est utilisé pour la supervision web, l'authentification SMTP, les éléments de type agent HTTP, la communication avec Vault et la source de données de télémétrie ClickHouse.

StartAgentPollers

Définit le nombre d'instances pré-forkées des `pollers` de l'agent Zabbix. Voir `MaxConcurrentChecksPerPoller`.

Par défaut: 1
 Plage: 0-1000

StartAPMCollector

Définissez cette valeur sur 1 pour démarrer une instance pré-forkée d'un collecteur APM (les processus `apm manager` et `apm worker`).

Par défaut : 0
 Plage : 0-1

StartBrowserPollers

Définit le nombre d'instances pré-forkées de `pollers` d'élément de navigateur `pollers`.

Par défaut: 1
 Plage: 0-1000

StartDBSyncers

Définit le nombre d'instances pré-forkées de `history syncers`.
Remarque : soyez prudent lorsque vous modifiez cette valeur, car l'augmenter peut faire plus de mal que de bien.

Par défaut : 4
 Plage : 1-100

StartDiscoverers

Définit le nombre d'instances pré-forkées de `discovery workers`.

Par défaut: 5
 Plage: 0-1000

StartHTTPAgentPollers

Définit le nombre d'instances pré-forkées de `HTTP agent pollers`. Voir `MaxConcurrentChecksPerPoller`.

Par défaut: 1
 Plage: 0-1000

StartHTTPPollers

Définit le nombre d'instances pré-forkées de `HTTP pollers`.

Par défaut: 1
 Plage: 0-1000

StartIPMIPollers

Définit le nombre d'instances pré-forkées de `IPMI pollers`.

Par défaut: 0
 Plage: 0-1000

StartJavaPollers

Définit le nombre d'instances pré-forkées de `Java pollers`.

Par défaut: 0
 Plage: 0-1000

StartODBCPollers

Définit le nombre d'instances pré-forkées de `ODBC pollers`.

Par défaut: 1
 Plage: 0-1000

StartPingers

Définit le nombre d'instances pré-forkées de `ICMP pingers`.

Par défaut: 1
 Plage: 0-1000

StartPollersUnreachable

Définit le nombre d'instances pré-forkées de pollers pour hôtes injoignables (y compris IPMI et Java). Au moins un poller pour hôtes injoignables doit être en cours d'exécution si des pollers réguliers, IPMI ou Java sont démarrés.

Par défaut: 1
 Plage: 0-1000

StartPollers

Définit le nombre d'instances pré-forkées de `pollers`.

Par défaut: 5
 Plage: 0-1000

StartPreprocessors

Définissez le nombre d'instances pré-démarrées de threads de prétraitement `worker` à une valeur au moins égale au nombre de cœurs CPU disponibles. Un nombre plus élevé de workers doit être défini si le prétraitement n'est pas limité par le CPU et comporte de nombreuses requêtes réseau.

Par défaut: 16
 Plage: 1-1000

StartSNMPPollers

Définit le nombre d'instances pré-forkées de `SNMP pollers`. Voir `MaxConcurrentChecksPerPoller`.

Par défaut: 1
 Plage: 0-1000

StartTelemetryQueryPollers

Définit le nombre d'instances pré-forkées de pollers de requêtes Telemetry asynchrones. Voir MaxConcurrentChecksPerPoller.

Valeur par défaut : 1
 Plage : 0-1000

StartSNMPTrapper

Définissez sur "1" pour démarrer un processus `SNMP trapper`.

Par défaut: 0
 Plage: 0-1

StartTrappers

Définit le nombre d'instances pré-forkées de `trappers`.
Les trappers acceptent les connexions entrantes de Zabbix sender et des agents actifs.

Par défaut: 5
 Plage: 0-1000

StartVMwareCollectors

Définit le nombre d'instances de `vmware collector` pré-forkées.

Par défaut: 0
 Plage: 0-250

StatsAllowedIP

Spécifiez les adresses IP (éventuellement en notation CIDR) ou les noms DNS des instances Zabbix externes autorisées à demander des statistiques. Liste séparée par des virgules.
 Si ce paramètre n'est pas défini, aucune requête de statistiques ne sera acceptée.
Si la prise en charge d'IPv6 est activée, alors `127.0.0.1`, `::127.0.0.1`, `::ffff:127.0.0.1` sont traitées de la même manière et `::/0` autorisera toute adresse IPv4 ou IPv6. `0.0.0.0/0` peut être utilisé pour autoriser toute adresse IPv4.

Exemple :

```
StatsAllowedIP=127.0.0.1,192.168.1.0/24,::1,2001:db8::/32,zabbix.example.com
```

TelemetryProvider

Définissez la source des données de télémétrie au format `<provider>[;<options>]` où :

- `<provider>` - l'une des sources de données de télémétrie prises en charge : `clickhouse`
- `<options>` - paires `key=value` séparées par des virgules. Les valeurs peuvent être placées entre guillemets.

Options pour ClickHouse :

- `url` - définissez l'URL HTTP ou HTTPS utilisée pour se connecter à la source des données de télémétrie.
- `db` - définissez le nom de la base de données.
- `username` - définissez le nom d'utilisateur utilisé pour se connecter à la base de données.
- `password` - définissez le mot de passe utilisé pour se connecter à la base de données.
- `ssl_cert_file`, `ssl_key_file`, `ssl_key_password`, `ssl_verify_peer`, `ssl_verify_host`, `ssl_ca_location`, `ssl_cert_location`, `ssl_key_location` - définissez ces options pour configurer une connexion sécurisée (TLS/SSL) à la base de données.
- `source_ip` - définissez l'adresse IP source utilisée pour les connexions sortantes vers la source des données de télémétrie.
- `vault_path` - définissez le chemin vers un secret Vault contenant les valeurs `username` et `password`. Ne l'utilisez pas avec les options `username` et `password`.

Exemple pour ClickHouse :

```
TelemetryProvider=clickhouse;url=http://localhost:8123,db=zabbix,username=zabbix,password=zabbix
```

Exemple pour ClickHouse avec Vault :

```
TelemetryProvider=clickhouse;url=http://localhost:8123,db=zabbix,vault_path=clickhouse
```

Délai d'attente

Spécifiez combien de temps attendre (en secondes) pour établir la connexion et échanger des données avec le serveur Zabbix, l'agent, le service web et les vérifications SNMP héritées (numéro d'OID unique ou chaîne).

Ce paramètre définit la durée de diverses opérations de communication :

- Exécution de commandes distantes sur l'agent Zabbix
- Exécution de commandes SSH / Telnet
- Replanification des éléments lorsque l'interface IPMI devient indisponible
- Envoi d'une réponse au serveur Zabbix en cas d'échec de l'échange de données en raison de problèmes de droits ou de chiffrement
- Délai limite pour les sockets asynchrones IPC et les options de contrôle d'exécution
- Requêtes DNS du collecteur asynchrone
- Réponse pour le heartbeat des vérifications actives
- Récupération des données (valeurs) de l'agent Zabbix auprès des agents actifs
- Récupération des données depuis Zabbix sender
- Envoi de la liste des vérifications actives à l'agent Zabbix

Ce délai d'attente **ne sera pas** utilisé pour les vérifications qui ont des paramètres de délai d'attente flexible configurés dans l'interface (au niveau global, du proxy ou de chaque élément). Par exemple, les éléments SNMP `walk[OID]` et `get[OID]` utilisent le délai d'attente configuré dans l'interface ; les vérifications SNMP héritées utilisent toujours la valeur de délai d'attente du serveur.

Par défaut : 3
 Plage : 1-30

TLSAccept

Spécifiez le niveau de chiffrement requis pour les connexions entrantes provenant du serveur Zabbix. Utilisé pour un proxy passif, ignoré sur un proxy actif. Plusieurs valeurs peuvent être spécifiées, séparées par des virgules : - *unencrypted* - accepter les connexions sans chiffrement (par défaut) ; - *psk* - accepter les connexions avec TLS et une clé prépartagée (PSK) ; - *cert* - accepter les connexions avec TLS et un certificat.

Obligatoire : oui pour un proxy passif, si des paramètres de certificat TLS ou de PSK sont définis (même pour une connexion *unencrypted*) ; sinon non

TLSCAFile

Spécifiez le chemin d'accès complet du fichier contenant les certificats de l'autorité de certification racine pour la vérification du certificat du pair, utilisé pour les communications chiffrées entre les composants Zabbix.

TLSCertFile

Spécifiez le chemin complet du fichier contenant le certificat du proxy ou la chaîne de certificats, utilisé pour les communications chiffrées entre les composants Zabbix.

TLSCipherAll

Spécifiez la chaîne de priorité GnuTLS ou la chaîne de chiffrement OpenSSL (TLS 1.2). Remplacez les critères de sélection de la suite de chiffrement par défaut pour le chiffrement basé sur les certificats et sur PSK.

Exemple :

```
TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256
```

TLSCipherAll13

Spécifiez la chaîne de chiffrement pour OpenSSL 1.1.1 ou une version plus récente dans TLS 1.3. Remplacez les critères de sélection par défaut de la suite de chiffrement pour le chiffrement basé sur les certificats et sur PSK.

Exemple pour GnuTLS :

```
NONE:+VERS-TLS1.2:+ECDHE-RSA:+RSA:+ECDHE-PSK:+PSK:+AES-128-GCM:+AES-128-CBC:+AEAD:+SHA256:+SHA1:+CURVE-ALL
```

Exemple pour OpenSSL :

```
EECDH+aRSA+AES128:RSA+aRSA+AES128:kECDHEPSK+AES128:kPSK+AES128
```

TLSCipherCert

Spécifiez la chaîne de priorité GnuTLS ou la chaîne de chiffrement OpenSSL (TLS 1.2). Remplacez les critères de sélection de la suite de chiffrement par défaut pour le chiffrement basé sur les certificats.

Exemple pour GnuTLS :

```
NONE:+VERS-TLS1.2:+ECDHE-RSA:+RSA:+AES-128-GCM:+AES-128-CBC:+AEAD:+SHA256:+SHA1:+CURVE-ALL:+COMP-NULL:+SIG
```

Exemple pour OpenSSL :

```
EECDH+aRSA+AES128:RSA+aRSA+AES128
```

TLSCipherCert13

Spécifiez la chaîne de chiffrement pour OpenSSL 1.1.1 ou une version plus récente dans TLS 1.3. Remplacez les critères de sélection de la suite de chiffrement par défaut pour le chiffrement basé sur des certificats.

TLSCipherPSK

Spécifiez la chaîne de priorité GnuTLS ou la chaîne de chiffrement OpenSSL (TLS 1.2). Remplacez les critères de sélection de la suite de chiffrement par défaut pour le chiffrement basé sur PSK.

Exemple pour GnuTLS :

```
NONE:+VERS-TLS1.2:+ECDHE-PSK:+PSK:+AES-128-GCM:+AES-128-CBC:+AEAD:+SHA256:+SHA1:+CURVE-ALL:+COMP-NULL:+SIG
```

Exemple pour OpenSSL :

```
kECDHEPSK+AES128:kPSK+AES128
```

TLSCipherPSK13

Spécifiez la chaîne de chiffrement pour OpenSSL 1.1.1 ou une version plus récente en TLS 1.3. Remplacez les critères de sélection par défaut de la suite de chiffrement pour le chiffrement basé sur PSK.

Exemple :

```
TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256
```

TLSConnect

Spécifiez le niveau de chiffrement requis pour les connexions du proxy au serveur Zabbix. Utilisé pour un proxy actif, ignoré sur un proxy passif. Une seule valeur peut être spécifiée : - *unencrypted* - connexion sans chiffrement (par défaut) ; - *psk* - connexion à l'aide de TLS et d'une clé prépartagée (PSK) ; - *cert* - connexion à l'aide de TLS et d'un certificat.

Obligatoire : oui pour un proxy actif, si des paramètres de certificat TLS ou de PSK sont définis (même pour une connexion *unencrypted*) ; sinon non

TLSCRLFile

Spécifiez le chemin d'accès complet du fichier contenant les certificats révoqués. Ce paramètre est utilisé pour les communications chiffrées entre les composants Zabbix.

TLSKeyFile

Spécifiez le chemin d'accès complet du fichier contenant la clé privée du proxy, utilisée pour les communications chiffrées entre les composants Zabbix.

TLSListen

Exiger des connexions TLS uniquement sur le socket trapper.

Valeurs prises en charge :

- required - accepter uniquement les connexions TLS

TLSPSKFile

Indiquez le chemin d'accès complet du fichier contenant la clé pré-partagée du proxy, utilisée pour les communications chiffrées avec le serveur Zabbix.

TLSPSKIdentity

Spécifiez la chaîne d'identité de la clé pré-partagée, utilisée pour les communications chiffrées avec le serveur Zabbix.

TLSServerCertIssuer

Spécifiez l'émetteur autorisé du certificat du serveur.

TLSServerCertSubject

Spécifiez le sujet du certificat de serveur autorisé.

TmpDir

Spécifiez le répertoire temporaire.

Par défaut: `/tmp`

TrapperTimeout

Spécifie le délai d'attente en secondes pour :
 - la récupération des données de configuration depuis le serveur Zabbix ;
 - l'exécution d'un script global ou l'exécution d'une commande distante.

Par défaut : 300
 Plage : 1-300

UnavailableDelay

Définissez la fréquence de vérification de la disponibilité de l'hôte pendant la période d'indisponibilité en secondes.

Par défaut: 60
 Plage: 1-3600

UnreachableDelay

Définit la fréquence de vérification de la disponibilité de l'hôte pendant la période d'inaccessibilité, en secondes.

Par défaut: 15
 Plage: 1-3600

UnreachablePeriod

Définit après combien de secondes d'inaccessibilité considérer l'hôte comme indisponible.

Par défaut: 45
 Plage: 1-3600

Utilisateur

Abaisser les privilèges vers un utilisateur spécifique existant sur le système.
Ne prend effet que si l'exécution est effectuée en tant que `root` et que `AllowRoot` est désactivé.

Par défaut: `zabbix`

Vault

Spécifiez le fournisseur du coffre-fort : - *HashiCorp* - HashiCorp KV Secrets Engine version 2 ; - *CyberArk* - CyberArk Central Credential Provider
Doit correspondre au fournisseur du coffre-fort défini dans l'interface.

Par défaut : `HashiCorp`

VaultAppRoleID

Spécifiez l'ID AppRole du coffre HashiCorp.
 Si défini, active l'authentification AppRole. Ne peut pas être utilisé avec VaultToken.
 Pris en charge uniquement pour le coffre HashiCorp.

VaultAppSecretID

Spécifiez l'ID secret AppRole du coffre HashiCorp.
 Pris en charge uniquement pour le coffre HashiCorp.

VaultDBPath

Spécifie l'emplacement permettant de récupérer les identifiants de base de données par clés. Il s'agit d'un chemin ou d'une requête Vault, selon le Vault.

Les clés utilisées pour HashiCorp sont `password` et `username`.

Exemple de chemin avec `VaultPrefix=/v1/secret/data/zabbix/` :

```
database
```

Exemple de chemin sans VaultPrefix :

```
secret/zabbix/database
```

Les clés utilisées pour CyberArk sont `Content` et `UserName`.

Exemple :

```
AppID=zabbix_server&Query=Safe=passwordSafe;Object=zabbix_proxy_database
```

Cette option ne peut être utilisée que si `DBUser` et `DBPassword` ne sont pas spécifiés.

VaultPrefix

Spécifiez un préfixe personnalisé pour le chemin ou la requête du coffre-fort, selon le Vault. Les valeurs par défaut les plus appropriées seront utilisées si ce n'est pas spécifié.
Notez que 'data' est automatiquement ajouté après le point de montage pour HashiCorp si `VaultPrefix` n'est pas spécifié.
Notez que 'data' est automatiquement ajouté après le point de montage pour HashiCorp si `VaultPrefix` n'est pas spécifié.

Exemple de préfixe pour HashiCorp :

```
/v1/secret/data/zabbix/
```

Exemple de préfixe pour Cyberark :

```
/AIMWebService/api/Accounts?
```

VaultTLSCertFile

Spécifiez le nom du fichier de certificat SSL utilisé pour l'authentification du client. Le fichier de certificat doit être au format PEM1.
Si le fichier de certificat contient également la clé privée, laissez le champ du fichier de clé SSL vide.
Le répertoire contenant ce fichier est spécifié par le paramètre de configuration SSLCertLocation.
Cette option peut être omise, mais elle est recommandée pour le coffre CyberArkCCP.

VaultTLSKeyFile

Spécifiez le nom du fichier de clé privée SSL utilisé pour l'authentification du client. Le fichier de clé privée doit être au format PEM1.
Le répertoire contenant ce fichier est spécifié par le paramètre de configuration SSLKeyLocation.
Cette option peut être omise, mais elle est recommandée pour le coffre CyberArkCCP.

VaultToken

Spécifiez le jeton d'authentification HashiCorp Vault qui doit avoir été généré exclusivement pour le proxy Zabbix avec une autorisation en lecture seule sur le chemin spécifié dans le paramètre de configuration facultatif VaultDBPath.
Une erreur se produit si `VaultToken` et la variable d'environnement `VAULT_TOKEN` sont définis en même temps.
 Ne peut pas être utilisé avec VaultAppRoleID.
 Pris en charge uniquement pour HashiCorp Vault.

VaultURL

Spécifiez l'URL du serveur vault. Le répertoire des certificats CA à l'échelle du système sera utilisé si SSLCALocation n'est pas spécifié.

Par défaut: `https://127.0.0.1:8200`

VMwareCacheSize

Définissez la taille de la mémoire partagée pour stocker les données VMware.
Une vérification interne VMware `zabbix[vmware,buffer,` peut être utilisée pour surveiller l'utilisation du cache VMware (voir Vérifications internes).
Notez que la mémoire partagée n'est pas allouée s'il n'existe aucune instance de collecteur VMware configurée au démarrage.

Par défaut: 8M
 Plage: 256K-2G

VMwareFrequency

Définissez la fréquence de collecte des données à partir d'un seul service VMware (en secondes).
Cette fréquence doit être définie sur l'intervalle de mise à jour le plus court de tout élément de surveillance VMware.

Par défaut: 60
 Plage: 10-86400

VMwarePerfFrequency

Définissez la fréquence de récupération des statistiques des compteurs de performance d'un seul service VMware (en secondes).
Cette fréquence doit être définie sur l'intervalle de mise à jour le plus court de tout élément de surveillance VMware qui utilise des compteurs de performance VMware.

Par défaut: 60
 Plage: 10-86400

VMwareTimeout

Définit le nombre maximal de secondes pendant lesquelles un `vmware collector` attendra une réponse du service VMware (vCenter ou hyperviseur ESX).

Par défaut: 10
 Plage: 1-300

WebDriverURL

Spécifiez l'URL de l'interface WebDriver.

Exemple (utilisé avec le serveur autonome Selenium WebDriver) :

```
WebDriverURL=http://localhost:4444
```

agent

Présentation générale L'agent Zabbix est déployé sur une cible de supervision afin de surveiller activement les ressources et les applications locales (disques durs, mémoire, statistiques du processeur, etc.).

L'agent collecte localement les informations opérationnelles et transmet les données au serveur Zabbix pour un traitement ultérieur. En cas de défaillance (par exemple, un disque dur saturé ou un processus de service arrêté), le serveur Zabbix peut alerter activement les administrateurs de la machine concernée ayant signalé la défaillance.

Les agents Zabbix sont très efficaces grâce à l'utilisation d'appels système natifs pour collecter les informations statistiques.

Il est possible d'exécuter plusieurs instances de l'agent sur un hôte.

Voir également :

- Protocoles de vérification passive et active
- Exécution de l'agent en tant que root
- Paramètres de configuration : agent Zabbix
- Paramètres de configuration : agent Zabbix (Windows)
- Agent Zabbix sur Microsoft Windows
- Agent Zabbix sur IBM AIX

Vérifications passives et actives

Les agents Zabbix peuvent effectuer des vérifications passives et actives :

- Vérifications passives - l'agent Zabbix répond à une requête du serveur Zabbix (ou du proxy). Par exemple, le serveur demande des données (par exemple, la charge CPU), et l'agent renvoie le résultat.
- Vérifications actives - l'agent Zabbix collecte et envoie des données sans attendre de requête du serveur Zabbix (ou du proxy). Tout d'abord, il récupère une liste d'éléments de surveillance depuis le serveur (charge CPU, mémoire disponible, etc.), puis collecte les données requises et envoie périodiquement de nouvelles valeurs en retour.

Le type de vérification de l'agent est configuré en sélectionnant le type d'élément de surveillance correspondant. L'agent Zabbix traite les éléments de type *Zabbix agent* ou *Zabbix agent (active)*.

Plates-formes prises en charge Pour les plates-formes prises en charge, consultez la page Requirements.

Agent sur les systèmes de type Unix L'agent Zabbix sur les systèmes de type Unix s'exécute sur l'hôte surveillé.

Installation

L'agent Zabbix peut être installé sur les systèmes basés sur Linux à l'aide de l'une des méthodes suivantes :

- Paquets Zabbix - sélectionnez le composant Agent (après avoir sélectionné votre version de Zabbix, la distribution du système d'exploitation et la version du système d'exploitation), puis suivez les instructions fournies ;
- Sources Zabbix - téléchargez les fichiers sources et compilez l'agent Zabbix en configurant les sources avec l'option `--enable-agent`.

Attention:
En général, les agents Zabbix 32 bits fonctionnent sur les systèmes 64 bits, mais peuvent échouer dans certains cas.

Note:
Des binaires précompilés de l'agent Zabbix sont disponibles au téléchargement pour macOS, IBM AIX, FreeBSD, OpenBSD et Solaris. Des binaires hérités, compatibles avec la version actuelle du serveur/proxy Zabbix, sont disponibles pour NetBSD et HP-UX. Pour installer et exécuter l'agent sur IBM AIX, consultez Agent Zabbix sur IBM AIX.

Si installé en tant que paquet

Zabbix agent s'exécute en tant que processus démon. L'agent peut être démarré en exécutant :

```
systemctl start zabbix-agent
```

Cela fonctionnera sur la plupart des systèmes GNU/Linux. Sur d'autres systèmes, vous devrez peut-être exécuter :

```
/etc/init.d/zabbix-agent start
```

Pour arrêter, redémarrer ou vérifier l'état de Zabbix agent, utilisez les commandes suivantes :

```
systemctl stop zabbix-agent
systemctl restart zabbix-agent
systemctl status zabbix-agent
```

Si installé depuis une archive binaire

Après avoir extrait l'archive, localisez le binaire `zabbix_agentd` et le fichier de configuration `zabbix_agentd.conf`. Leur emplacement dans l'archive dépend de la plateforme (par exemple, `sbin/` et `conf/`, ou `usr/sbin/` et `etc/zabbix/`). Exécutez directement le binaire depuis le répertoire extrait et indiquez le fichier de configuration avec l'option `-c`. Par exemple :

```
/opt/zabbix/sbin/zabbix_agentd -c /opt/zabbix/etc/zabbix/zabbix_agentd.conf
```

Sur les systèmes de type UNIX, l'agent Zabbix est conçu pour s'exécuter avec un utilisateur non root. S'il est démarré avec les privilèges root, il bascule vers l'utilisateur `zabbix` (ou vers l'utilisateur indiqué par le paramètre `[User](<PARAMS>#user)`), sauf si `[AllowRoot=1](<PARAMS>#allowroot)` est défini dans le fichier de configuration. Cet utilisateur doit exister sur le système ; sinon, l'agent ne démarre pas.

Sur IBM AIX, l'agent peut également être exécuté avec le System Resource Controller (SRC) et démarré au démarrage du système ; voir Agent Zabbix sur IBM AIX.

Démarrer manuellement

Vous pouvez démarrer l'agent Zabbix en exécutant directement le binaire `zabbix_agentd`, par exemple :

```
zabbix_agentd
```

Cela fonctionne lorsque le répertoire contenant `zabbix_agentd` est inclus dans `PATH`.

Agent sur les systèmes Windows L'agent Zabbix sur Windows s'exécute en tant que service Windows.

Installation

Zabbix agent peut être installé sous Windows à l'aide de l'une des méthodes suivantes :

- Binaires précompilés de Zabbix agent - téléchargez le package d'installation MSI de Zabbix agent et suivez les instructions de la page Installation de l'agent Windows à partir d'un MSI ;
- Sources Zabbix - téléchargez les fichiers sources et suivez les instructions de la page Compilation de Zabbix agent sous Windows.

Consultez la page Zabbix agent sur Microsoft Windows pour plus de détails sur l'installation de Zabbix agent (à partir d'une archive ZIP) en tant que service Windows.

Options Les paramètres de ligne de commande suivants peuvent être utilisés avec l'agent Zabbix :

Paramètre	Description
Agent UNIX et Windows	
`-c --config <config-file>`	Chemin vers le fichier de configuration. Vous pouvez utiliser cette option pour spécifier un fichier de configuration différent de celui utilisé par défaut.
`-f --foreground`	Exécuter l'agent Zabbix au premier plan (valeur par défaut : `true`).
`-p --print`	Afficher les éléments connus, puis quitter. Notez que pour renvoyer également les résultats des paramètres utilisateur, vous devez spécifier le fichier de configuration (s'il ne se trouve pas à l'emplacement par défaut).
`-t --test <item key>`	Tester l'élément spécifié, puis quitter. Notez que pour renvoyer également les résultats des paramètres utilisateur, vous devez spécifier le fichier de configuration (s'il ne se trouve pas à l'emplacement par défaut).
`-T --test-config`	Valider le fichier de configuration, puis quitter.
`-h --help`	Afficher les informations d'aide.
`-V --version`	Afficher le numéro de version.
Agent UNIX uniquement	
`-R --runtime-control <option>`	Effectuer des fonctions administratives. Voir contrôle à l'exécution.
Agent Windows uniquement	
`-m --multiple-agents`	Utiliser plusieurs instances de l'agent (avec les options `-i`, `-d`, `-s`, `-x`). Pour distinguer les noms de service des instances, chaque nom de service inclura la valeur Hostname du fichier de configuration spécifié.

Paramètre	Description
`-S --startup-type <value>`	Définir le type de démarrage du service de l'agent Zabbix pour Windows. Valeurs autorisées : `automatic` - *(par défaut)* démarrer automatiquement le service au démarrage de Windows ; `delayed` - retarder le démarrage du service jusqu'à ce que les services démarrés automatiquement aient terminé leur démarrage (disponible sous Windows Server 2008/Vista et versions ultérieures) ; `manual` - démarrer le service manuellement (par un utilisateur ou une application) ; `disabled` - désactiver le service afin qu'il ne puisse pas être démarré par un utilisateur ou une application. Vous pouvez utiliser cette option avec l'option `-i`, ou séparément pour modifier le type de démarrage d'un service déjà installé.
`-i --install`	Installer l'agent Zabbix pour Windows en tant que service.
`-d --uninstall`	Désinstaller le service de l'agent Zabbix pour Windows.
`-s --start`	Démarrer le service de l'agent Zabbix pour Windows.
`-x --stop`	Arrêter le service de l'agent Zabbix pour Windows.

Exemples spécifiques d'utilisation des paramètres de ligne de commande :

- Afficher tous les éléments intégrés de l'agent avec leurs valeurs.
- Tester un paramètre utilisateur avec la clé `mysql.ping` définie dans le fichier de configuration spécifié.
- Installer un service Agent Zabbix pour Windows en utilisant le chemin par défaut vers le fichier de configuration.
- Installer un service Agent Zabbix [Hostname] pour Windows en utilisant le fichier de configuration situé dans le même dossier que l'exécutable de l'agent et rendre le nom du service unique en l'étendant avec la valeur `Hostname` du fichier de configuration.
- Modifier le type de démarrage d'un service Agent Zabbix installé pour Windows en utilisant le fichier de configuration situé dans le même dossier que l'exécutable de l'agent.

```
zabbix_agentd --print
zabbix_agentd -t "mysql.ping" -c /etc/zabbix/zabbix_agentd.conf
zabbix_agentd.exe -i
zabbix_agentd.exe -i -m -c zabbix_agentd.conf
zabbix_agentd.exe -c zabbix_agentd.conf -S delayed
```

Contrôle à l'exécution

Les options de contrôle à l'exécution permettent de modifier le niveau de journalisation des processus de l'agent.

Option	Description	Cible
`log_level_increase[=<target>]`	Augmente le niveau de journalisation. Si la cible n'est pas spécifiée, tous les processus sont concernés.	La cible peut être spécifiée comme suit : **type de processus** - tous les processus du type spécifié (par exemple, `listener`) Voir tous les types de processus de l'agent. **type de processus,N** - type et numéro du processus (par exemple, `listener,3`) **pid** - identifiant du processus (1 à 65535). Pour les valeurs supérieures, spécifiez la cible sous la forme « type de processus,N ».
`log_level_decrease[=<target>]`	Diminue le niveau de journalisation. Si la cible n'est pas spécifiée, tous les processus sont concernés.	
`userparameter_reload`	Recharge les valeurs des options `UserParameter` et `Include` à partir du fichier de configuration actuel.	

Exemples :

- Augmentation du niveau de journalisation de tous les processus
- Augmentation du niveau de journalisation du troisième processus `listener`
- Augmentation du niveau de journalisation du processus dont le PID est 1234
- Diminution du niveau de journalisation de tous les processus de vérification active

```
zabbix_agentd -R log_level_increase
zabbix_agentd -R log_level_increase=listener,3
```

```
zabbix_agentd -R log_level_increase=1234
zabbix_agentd -R log_level_decrease="active checks"
```

> **Note:**
> Le contrôle à l'exécution n'est pas pris en charge sur OpenBSD, NetBSD et Windows.

Types de processus de l'agent Les types de processus de l'agent sont les suivants :

- `active checks` - processus d'exécution des vérifications actives ;
- `collector` - processus de collecte des données ;
- `listener` - processus d'écoute des vérifications passives.

Le fichier journal de l'agent peut être utilisé pour observer ces types de processus.

Le fichier journal de l'agent est créé avec des permissions de lecture et d'écriture pour le propriétaire du fichier uniquement. De plus, le fichier est lisible par le groupe du propriétaire. Toutes les autres permissions sont refusées.

Traiter l'utilisateur L'agent Zabbix sur UNIX est conçu pour s'exécuter en tant qu'utilisateur non root. Il s'exécutera avec les privilèges de l'utilisateur non root qui l'a démarré. Vous pouvez donc exécuter l'agent avec n'importe quel utilisateur non root sans problème.

Si vous essayez de l'exécuter en tant que « root », il basculera vers un utilisateur « zabbix » codé en dur, qui doit être présent sur votre système. Vous ne pouvez exécuter l'agent en tant que « root » que si vous modifiez en conséquence le paramètre `AllowRoot` dans le fichier de configuration de l'agent.

Fichier de configuration Le chemin par défaut du fichier de configuration dépend du système d'exploitation et de la méthode d'installation de Zabbix :

- Sous UNIX, le chemin par défaut est :
  - Pour une installation à partir des sources : `/usr/local/etc/zabbix_agentd.conf`, ou celui défini par les variables au moment de la compilation `--sysconfdir` ou `--prefix`.
  - Pour une installation à partir de paquets ou de l'appliance : `/etc/zabbix/zabbix_agentd.conf`.
- Sous Windows, le chemin par défaut est `C:\Program Files\Zabbix Agent\zabbix_agentd.conf`.

Si une seule instance de l'agent est exécutée sur un hôte, elle peut utiliser le fichier de configuration par défaut ou un fichier de configuration spécifié sur la ligne de commande. En cas d'instances multiples, chaque instance de l'agent doit disposer de son propre fichier de configuration (l'une des instances peut utiliser le fichier de configuration par défaut).

Pour plus d'informations sur la configuration de l'agent Zabbix, consultez les options du fichier de configuration de l'agent Zabbix pour Unix ou de l'agent Zabbix pour Windows.

Paramètres régionaux Notez que l'agent nécessite des paramètres régionaux UTF-8 afin que certains éléments textuels de l'agent puissent renvoyer le contenu attendu. La plupart des systèmes Unix modernes ont des paramètres régionaux UTF-8 par défaut, cependant, sur certains systèmes, il peut être nécessaire de les définir explicitement.

Code de sortie L'agent Zabbix renvoie 0 en cas de sortie réussie et 1 en cas d'échec.

Protocoles de vérification passifs et actifs

Cette section fournit des détails sur le protocole des vérifications passives et actives effectuées par Zabbix agent et Zabbix agent 2.

Zabbix utilise un protocole de communication basé sur JSON pour communiquer avec les agents.

Les protocoles de Zabbix agent et de Zabbix agent 2 ont été unifiés depuis Zabbix 7.0. La différence entre les requêtes/réponses de Zabbix agent et de Zabbix agent 2 est exprimée par la valeur de la balise "variant".

Vérifications passives

Une vérification passive est une simple demande de données. Le serveur Zabbix ou le proxy demande certaines données (par exemple, la charge CPU) et l'agent Zabbix renvoie le résultat au serveur.

Les vérifications passives sont exécutées de manière asynchrone - il n'est pas nécessaire de recevoir la réponse à une requête avant que d'autres vérifications ne soient lancées. La résolution DNS est également asynchrone.

Le collecteur de l'agent tentera de se connecter à toutes les adresses renvoyées par la recherche DNS. Cela garantit que si une adresse IP est inaccessible, le collecteur essaiera l'adresse disponible suivante, augmentant ainsi la probabilité d'une connexion réussie. Cette amélioration s'applique à la fois au serveur Zabbix et au proxy.

La concurrence maximale des vérifications asynchrones est de 1000 (définie par le paramètre MaxConcurrentChecksPerPoller).

Le nombre de collecteurs d'agent asynchrones est défini par le paramètre StartAgentPollers.

Requête du serveur

Pour la définition de l'en-tête et de la longueur des données, veuillez vous référer aux détails du protocole.

```
{
  "request": "passive checks",
  "data": [
    {
      "key": "agent.version",
      "timeout": 3
    }
  ]
}
```

Field		Type	Mandatory	Value
request		*string*	yes	`"passive checks"`
data		*array of object*	yes	Élément de vérification passive.
	key	*string*	yes	Clé de l'élément avec les macros développées.
	timeout	*number*	yes	Délai d'attente de communication.

Réponse de l'agent

```
{
  "version": "8.0.0",
  "variant": 2,
  "data": [
    {
      "value": "8.0.0"
    }
  ]
}
```

Field		Type	Mandatory	Value
version		*string*	yes	Le numéro de version de l'agent.
variant		*number*	yes	La variante de l'agent (*1* - Zabbix agent, *2* - Zabbix agent 2).
data		*array of object*	yes	Contient le résultat de la vérification.
	value	*string*	no	La valeur de l'élément si la vérification a réussi.
	error	*string*	no	Le message d'erreur si la vérification n'a pas réussi.

Par exemple, pour les éléments pris en charge :

1. Le serveur ouvre une connexion TCP
2. Le serveur envoie **<HEADER><DATALEN>{"request":"passive checks","data":[{"key":"agent.ping","timeout":3}]}**
3. L'agent lit la requête et répond avec **<HEADER><DATALEN>{"version":"8.0.0","variant":2,"data":[{"value":1}]}**
4. Le serveur traite les données pour obtenir la valeur, '1' dans notre cas
5. La connexion TCP est fermée

Pour les éléments non pris en charge :

1. Le serveur ouvre une connexion TCP
2. Le serveur envoie **<HEADER><DATALEN>{"request":"passive checks","data":[{"key":"vfs.fs.size[/nono]","timeout":3}]**
3. L'agent lit la requête et répond avec **<HEADER><DATALEN>{"version":"8.0.0","variant":2,"data":[{"error":"Unsupported item key."}]}**
4. Le serveur traite les données, change l'état de l'élément en non pris en charge avec le message d'erreur spécifié
5. La connexion TCP est fermée

Basculement vers l'ancien protocole

Pour garantir que le serveur ou le proxy Zabbix puisse fonctionner avec des agents des versions antérieures à 7.2, qui utilisent le protocole en texte clair, un basculement vers l'ancien protocole est implémenté.

Les vérifications passives sont effectuées à l'aide du protocole JSON (7.0 et versions ultérieures) après un redémarrage ou lorsque la configuration de l'interface est modifiée.
Si aucune réponse JSON valide n'est reçue (l'agent a envoyé "ZBX_NOTSUPPORTED"), Zabbix mettra en cache l'interface comme utilisant l'ancien protocole et **réessaiera** la vérification en envoyant uniquement la clé de l'élément.

Notez que chaque heure, le serveur/proxy Zabbix tentera à nouveau d'utiliser le nouveau protocole avec toutes les interfaces, en revenant à l'ancien protocole si nécessaire.

Vérifications actives

Les vérifications actives nécessitent un traitement plus complexe. L'agent doit d'abord récupérer auprès du serveur/proxy une liste d'éléments et/ou de commandes à distance à traiter indépendamment.

Les serveurs/proxys auprès desquels récupérer les vérifications actives sont indiqués dans le paramètre `ServerActive` du fichier de configuration de l'agent. La fréquence de demande de ces vérifications est définie par le paramètre `RefreshActiveChecks` dans le même fichier de configuration. Cependant, si l'actualisation des vérifications actives échoue, une nouvelle tentative est effectuée après 60 secondes, valeur codée en dur.

Note:
Depuis Zabbix 6.4, l'agent (en mode actif) ne reçoit plus du serveur/proxy une copie complète de la configuration toutes les deux minutes (par défaut). À la place, afin de réduire le trafic réseau et l'utilisation des ressources, une synchronisation incrémentielle de la configuration est effectuée toutes les 5 secondes (par défaut), à l'issue de laquelle le serveur/proxy fournit une copie complète de la configuration **uniquement** si l'agent ne l'a pas encore reçue, ou si quelque chose a changé dans la configuration de l'hôte, les macros globales ou les expressions régulières globales.

L'agent envoie ensuite périodiquement les nouvelles valeurs au serveur(s). Si l'agent a reçu des commandes à distance à exécuter, le résultat de l'exécution sera également envoyé. Notez que l'exécution de commandes à distance sur un agent actif est prise en charge depuis Zabbix agent 7.0.

Note:
Si un agent se trouve derrière le pare-feu, vous pouvez envisager d'utiliser uniquement des vérifications actives, car dans ce cas, vous n'auriez pas besoin de modifier le pare-feu pour autoriser les connexions entrantes initiales.

Obtention de la liste des éléments

Requête de l'agent

La requête de vérifications actives est utilisée pour obtenir les vérifications actives à traiter par l'agent.
Cette requête est envoyée par l'agent au démarrage, puis à intervalles définis par RefreshActiveChecks.

```
{
  "request": "active checks",
  "host": "Zabbix server",
  "host_metadata": "mysql,nginx",
  "interface": "zabbix.server.lan",
  "ip": "159.168.1.1",
  "port": 12050,
  "version": "8.0.0",
  "variant": 2,
  "config_revision": 1,
  "session": "e3dcbd9ace2c9694e1d7bbd030eeef6e"
}
```

Field	Type	Mandatory	Value
request	*string*	yes	`active checks`
host	*string*	yes	Nom de l'hôte.
host_metadata	*string*	no	Le paramètre de configuration HostMetadata ou la valeur de l'élément de surveillance HostMetadataItem.
interface	*string*	no	Le paramètre de configuration HostInterface ou la valeur de l'élément de surveillance HostInterfaceItem.
ip	*string*	no	La première adresse IP du paramètre de configuration ListenIP si elle est définie.

Field	Type	Mandatory	Value
port	*number*	no	La valeur du paramètre de configuration ListenPort si elle est définie et qu'il ne s'agit pas du port d'écoute par défaut de l'agent.
version	*string*	yes	Le numéro de version de l'agent.
variant	*number*	yes	La variante de l'agent (*1* - Zabbix agent, *2* - Zabbix agent 2).
config_revision	*number*	no	Identifiant de configuration pour la synchronisation incrémentielle de la configuration.
session	*string*	no	Identifiant de session pour la synchronisation incrémentielle de la configuration.

Réponse du serveur

La réponse aux vérifications actives est envoyée par le serveur à l'agent après le traitement de la requête de vérifications actives.

```
{
  "response": "success",
  "config_revision": 2,
  "data": [
    {
      "key": "system.uptime",
      "itemid": 1234,
      "delay": "10s",
      "lastlogsize": 0,
      "mtime": 0
    },
    {
      "key": "agent.version",
      "itemid": 5678,
      "delay": "10m",
      "lastlogsize": 0,
      "mtime": 0,
      "timeout": "30s"
    }
  ],
  "commands": [
    {
      "command": "df -h --output=source,size / | awk 'NR>1 {print $2}'",
      "id": 1324,
      "wait": 1
    }
  ]
}
```

Field	Type	Mandatory	Value
response	*string*	yes	`success` | `failed`
info	*string*	no	Informations sur l'erreur en cas d'échec.
data	*array of objects*	no	Éléments de vérification active. Omis si la configuration de l'hôte est inchangée.
key	*string*	no	Clé de l'élément avec macros développées.
itemid	*number*	no	Identifiant de l'élément.
delay	*string*	no	Intervalle de mise à jour de l'élément. Les intervalles flexibles/de planification sont pris en charge par Zabbix agent et Zabbix agent 2 depuis Zabbix 7.0.
lastlogsize	*number*	no	lastlogsize de l'élément.
mtime	*number*	no	mtime de l'élément.
timeout	*string*	no	Délai d'attente de l'élément.
refresh_unsupported	*number*	no	Intervalle de rafraîchissement des éléments non pris en charge.
regexp	*array of objects*	no	Expressions régulières globales.
name	*string*	no	Nom de l'expression régulière globale.
expression	*string*	no	Expression régulière globale.

Field	Type	Mandatory	Value
expression_type	*number*	no	Type d'expression régulière globale.
exp_delimiter	*string*	no	Délimiteur de l'expression régulière globale.
case_sensitive	*number*	no	Paramètre de sensibilité à la casse de l'expression régulière globale.
commands	*array of objects*	no	Commandes distantes à exécuter. Incluses si l'exécution de commandes distantes a été déclenchée par une opération d'action ou par l'exécution manuelle d'un script. Notez que l'exécution de commandes distantes sur un agent actif est prise en charge depuis Zabbix agent 7.0. Les anciens agents actifs ignoreront toutes les commandes distantes incluses dans la réponse du serveur aux vérifications actives.
command	*string*	no	Commande distante.
id	*number*	no	Identifiant de la commande distante.
wait	*number*	no	Mode d'exécution de la commande distante ("0" (nowait) pour les commandes provenant des opérations d'action ; "1" (wait) pour les commandes provenant de l'exécution manuelle d'un script).
timeout	*number*	no	Délai d'attente d'exécution de la commande distante dans la configuration du serveur/proxy.
config_revision	*number*	no	Identifiant de configuration pour la synchronisation incrémentielle de la configuration. Omis si la configuration de l'hôte est inchangée. Incrémenté si la configuration de l'hôte est modifiée.

Le serveur doit répondre avec success.

Par exemple :

1. L'agent ouvre une connexion TCP
2. L'agent demande la liste des vérifications
3. Le serveur répond avec une liste d'éléments et de commandes distantes à exécuter
4. L'agent analyse la réponse
5. La connexion TCP est fermée
6. L'agent commence la collecte périodique des données et exécute les commandes distantes (pris en charge depuis Zabbix agent 7.0)

Attention:
Notez que des données de configuration (sensibles) peuvent devenir accessibles à des parties ayant accès au port trapper du serveur Zabbix lors de l'utilisation d'une vérification active. Cela est possible car n'importe qui peut se faire passer pour un agent actif et demander les données de configuration des éléments ; l'authentification n'a pas lieu, sauf si vous utilisez des options de chiffrement.

Envoi des données collectées

L'agent envoie

La requête de données de l'agent contient les valeurs des éléments collectées et les valeurs des commandes distantes exécutées (le cas échéant).

```
{
  "request": "agent data",
  "data": [
    {
      "id": 1,
      "itemid": 5678,
      "value": "7.0.0",
      "clock": 1712830783,
      "ns": 76808644
    },
    {
      "id": 2,
      "itemid": 1234,
      "value": "69672",
      "clock": 1712830783,
      "ns": 77053975
    }
  ],
```

```
  "commands": [
    {
      "id": 1324,
      "value": "16G"
    }
  ],
  "session": "8495cd52070e6ca52b371f29c8574165",
  "host": "Zabbix server",
  "version": "8.0.0",
  "variant": 2
}
```

Field	Type	Mandatory	Value
request	*string*	yes	`agent data`
data	*array of objects*	yes	Valeurs des éléments.
id	*number*	yes	L'identifiant de la valeur (compteur incrémental utilisé pour vérifier les valeurs en double en cas de problèmes réseau).
itemid	*number*	yes	L'identifiant de l'élément.
value	*string*	no	La valeur de l'élément.
lastlogsize	*number*	no	Le lastlogsize de l'élément.
mtime	*number*	no	Le mtime de l'élément.
state	*number*	no	L'état de l'élément.
source	*string*	no	La source du journal d'événements de la valeur.
eventid	*number*	no	L'eventid du journal d'événements de la valeur.
severity	*number*	no	La gravité du journal d'événements de la valeur.
timestamp	*number*	no	L'horodatage du journal d'événements de la valeur.
clock	*number*	yes	L'horodatage de la valeur (secondes depuis l'Epoch).
ns	*number*	yes	Les nanosecondes de l'horodatage de la valeur.
commands	*array of objects*	no	Résultat de l'exécution des commandes distantes. Notez que l'exécution de commandes distantes sur un agent actif est prise en charge depuis Zabbix agent 7.0. Les agents actifs plus anciens ignoreront toute commande distante incluse dans la réponse du serveur aux vérifications actives.
id	*number*	no	Identifiant de la commande distante.
value	*string*	no	Résultat de l'exécution de la commande distante si l'exécution a réussi.
error	*string*	no	Message d'erreur de l'exécution de la commande distante si l'exécution a échoué.
session	*string*	yes	Identifiant de session unique généré à chaque démarrage de l'agent (32 chiffres hexadécimaux).
host	*string*	yes	Nom de l'hôte.
version	*string*	yes	Numéro de version de l'agent.
variant	*number*	yes	Variante de l'agent (*1* - Zabbix agent, *2* - Zabbix agent 2).

Un ID virtuel est attribué à chaque valeur. L'ID de valeur est un simple compteur croissant, unique au sein d'une session de données (identifiée par le jeton de session). Cet ID est utilisé pour ignorer les valeurs en double qui peuvent être envoyées dans des environnements où la connectivité est médiocre.

Réponse du serveur

La réponse aux données de l'agent est envoyée par le serveur à l'agent après le traitement de la requête de données de l'agent.

```
{
  "response": "success",
  "info": "processed: 2; failed: 0; total: 2; seconds spent: 0.003534"
}
```

Field	Type	Mandatory	Value
response	*string*	yes	`success` | `failed`
info	*string*	yes	Résultats du traitement des éléments.

Attention:
Si l'envoi de certaines valeurs échoue sur le serveur (par exemple, parce que l'hôte ou l'élément a été désactivé ou supprimé), l'agent ne réessaiera pas d'envoyer ces valeurs.

Par exemple :

1. L'agent ouvre une connexion TCP
2. L'agent envoie une liste de valeurs
3. Le serveur traite les données et renvoie l'état
4. La connexion TCP est fermée

Attention:
Le message d'erreur sera tronqué à 2048 symboles côté serveur.

Message de pulsation

L'agent envoie

Le message de pulsation est envoyé par un agent actif au serveur/proxy Zabbix toutes les `HeartbeatFrequency` secondes (configuré dans le fichier de configuration de l'agent Zabbix ou de Zabbix agent 2).

Il est utilisé pour surveiller la disponibilité des vérifications actives.

```
{
  "request": "active check heartbeat",
  "host": "Zabbix server",
  "heartbeat_freq": 60,
  "version": "8.0.0",
  "variant": 2
}
```

Champ	Type	Obligatoire	Valeur
request	*string*	oui	`active check heartbeat`
host	*string*	oui	Le nom de l'hôte.
heartbeat_freq	*number*	oui	La fréquence de pulsation de l'agent (paramètre de configuration HeartbeatFrequency).
version	*string*	oui	Le numéro de version de l'agent.
variant	*number*	oui	La variante de l'agent (*1* - agent Zabbix, *2* - Zabbix agent 2).

Réponse de redirection

Lorsqu'un hôte a été réaffecté, le serveur peut demander à l'agent de rediriger sa pulsation (ainsi que les vérifications actives suivantes) vers un autre proxy ou une autre instance du serveur.

```
  {
    "response": "failed",
    "redirect": {
      "revision": 2,
      "address": "192.0.2.1:10055"
    }
  }
```

Champ		Type	Obligatoire	Valeur
response		*string*	oui	`success` | `failed`
redirect		*object*	oui	Instructions de redirection.
	revision	*number*	oui	Identifiant de révision de la configuration.
	address	*string*	oui	Adresse du serveur/proxy cible.

Ancien protocole XML

Note:
Zabbix acceptera jusqu'à 16 Mo de données XML encodées en Base64, mais une valeur décodée individuelle ne doit pas dépasser 64 Ko ; sinon, elle sera tronquée à 64 Ko lors du décodage.

Exécution de l'agent en tant que root

Depuis Zabbix **5.0.0**, le fichier de service systemd pour l'agent Zabbix dans les packages officiels inclut explicitement des directives pour `User` et `Group`. Les deux sont définies sur `zabbix`.

Il n'est plus possible de configurer l'utilisateur sous lequel l'agent Zabbix s'exécute via le fichier `zabbix_agentd.conf`, car l'agent ignorera cette configuration et s'exécutera sous l'utilisateur spécifié dans le fichier de service systemd. Pour exécuter l'agent Zabbix en tant que root, vous devez appliquer les modifications décrites ci-dessous.

Agent Zabbix

Pour remplacer l'utilisateur et le groupe par défaut de l'agent Zabbix, exécutez :

```
systemctl edit zabbix-agent
```

Ajoutez ensuite le contenu suivant :

```
[Service]
User=root
Group=root
```

Rechargez les daemons et redémarrez le service zabbix-agent :

```
systemctl daemon-reload
systemctl restart zabbix-agent
```

Pour **l'agent Zabbix**, cela réactive la fonctionnalité permettant de configurer l'utilisateur dans le fichier de configuration de l'agent. Vous devez maintenant définir les paramètres de configuration `User=root` et `AllowRoot=1` dans le fichier de configuration.

Zabbix agent 2

Pour remplacer l'utilisateur et le groupe par défaut de Zabbix agent 2, exécutez :

```
systemctl edit zabbix-agent2
```

Ajoutez ensuite le contenu suivant :

```
[Service]
User=root
Group=root
```

Rechargez les démons et redémarrez le service zabbix-agent2 :

```
systemctl daemon-reload
systemctl restart zabbix-agent2
```

Pour **Zabbix agent2**, cela détermine entièrement l'utilisateur sous lequel il s'exécute. Aucune modification supplémentaire n'est requise.

Paramètres de configuration: Zabbix agent

Vue d'ensemble

Les paramètres de configuration permettent de personnaliser le comportement de l'agent Zabbix.

Indiquez les valeurs des paramètres dans le fichier de configuration de l'agent Zabbix (`zabbix_agentd.conf`). Tous les paramètres sont **facultatifs**, sauf indication explicite contraire.

Cette page est structurée comme suit :

- Vue d'ensemble des paramètres (pour afficher tous les détails d'un paramètre, cliquez sur son nom)
- Détails des paramètres

Paramètre	Description
Alias	Définir un alias pour une clé d'élément.
AllowKey	Autoriser l'exécution des clés d'élément correspondant à un modèle.
AllowKeyRegexp	Autoriser l'exécution des clés d'élément correspondant à un modèle basé sur une expression régulière.
AllowRoot	Autoriser l'agent à s'exécuter en tant que `root`.
BufferSend	Définir la fréquence d'envoi des valeurs de la mémoire tampon au serveur/proxy Zabbix.
BufferSize	Définir le nombre maximal de valeurs dans la mémoire tampon.
DebugLevel	Définir le niveau de débogage.

Paramètre	Description
DenyKey	Interdire l'exécution des clés d'élément correspondant à un modèle.
DenyKeyRegexp	Interdire l'exécution des clés d'élément correspondant à un modèle basé sur une expression régulière.
EnableRemoteCommands	Autoriser les commandes distantes provenant du serveur Zabbix. Ce paramètre est obsolète ; utilisez `AllowKey`, `AllowKeyRegexp`, `DenyKey` ou `DenyKeyRegexp` à la place.
HeartbeatFrequency	Définir la fréquence des messages de pulsation, en secondes.
HostInterface	Indiquer l'interface de l'hôte.
HostInterfaceItem	Indiquer un élément permettant d'obtenir l'interface de l'hôte.
HostMetadata	Indiquer les métadonnées de l'hôte.
HostMetadataItem	Indiquer un élément d'agent Zabbix permettant d'obtenir les métadonnées de l'hôte.
Hostname	Indiquer le nom d'hôte.
HostnameItem	Indiquer un élément d'agent Zabbix permettant d'obtenir le nom d'hôte.
Include	Indiquer un répertoire ou des fichiers individuels à inclure dans le fichier de configuration.
ListenBacklog	Définir le nombre maximal de connexions en attente dans la file TCP.
ListenIP	Indiquer l'adresse IP d'écoute de l'agent pour les connexions provenant du serveur/proxy.
ListenPort	Indiquer le port d'écoute de l'agent.
LoadModule	Indiquer le module à charger au démarrage de l'agent.
LoadModulePath	Indiquer le chemin complet vers l'emplacement des modules de l'agent.
LogFile	Indiquer le fichier journal de l'agent.
LogFileSize	Définir la taille maximale du fichier journal.
LogRemoteCommands	Activer la journalisation des commandes shell exécutées sous forme d'avertissements.
LogType	Indiquer le type de sortie du journal.
MaxLinesPerSecond	Définir le nombre maximal de nouvelles lignes par seconde à envoyer au serveur/proxy Zabbix lors du traitement des contrôles actifs `log` ou `logrt`.
PidFile	Indiquer le fichier PID.
RefreshActiveChecks	Indiquer la fréquence d'actualisation de la liste des contrôles actifs.
Server	Indiquer les adresses IP ou les noms DNS des serveurs/proxys Zabbix.
ServerActive	Indiquer l'adresse du serveur/proxy Zabbix ou la configuration du cluster depuis laquelle obtenir les contrôles actifs.
SourceIP	Indiquer l'adresse IP source.
StartAgents	Définir le nombre d'instances pré-créées de `zabbix_agentd` pour le traitement des contrôles passifs.
Timeout	Indiquer le délai d'attente (en secondes) pour établir la connexion et échanger des données avec le proxy ou le serveur Zabbix.
TLSAccept	Indiquer le niveau de chiffrement requis pour les connexions entrantes.
TLSCAFile	Indiquer le chemin complet du fichier contenant les certificats de l'autorité de certification (CA) de niveau supérieur pour la vérification du certificat du pair, utilisé pour les communications chiffrées entre les composants Zabbix.
TLSCertFile	Indiquer le chemin complet du fichier contenant le certificat ou la chaîne de certificats de l'agent, utilisé pour les communications chiffrées entre les composants Zabbix.
TLSCipherAll	Indiquer la chaîne de priorité GnuTLS ou la chaîne de chiffrement OpenSSL (TLS 1.2). Remplacer les critères de sélection par défaut de la suite de chiffrement pour le chiffrement basé sur des certificats et des clés pré-partagées (PSK).
TLSCipherAll13	Indiquer la chaîne de chiffrement pour OpenSSL 1.1.1 ou version ultérieure avec TLS 1.3. Remplacer les critères de sélection par défaut de la suite de chiffrement pour le chiffrement basé sur des certificats et des clés pré-partagées (PSK).
TLSCipherCert	Indiquer la chaîne de priorité GnuTLS ou la chaîne de chiffrement OpenSSL (TLS 1.2). Remplacer les critères de sélection par défaut de la suite de chiffrement pour le chiffrement basé sur des certificats.
TLSCipherCert13	Indiquer la chaîne de chiffrement pour OpenSSL 1.1.1 ou version ultérieure avec TLS 1.3. Remplacer les critères de sélection par défaut de la suite de chiffrement pour le chiffrement basé sur des certificats.
TLSCipherPSK	Indiquer la chaîne de priorité GnuTLS ou la chaîne de chiffrement OpenSSL (TLS 1.2). Remplacer les critères de sélection par défaut de la suite de chiffrement pour le chiffrement basé sur des clés pré-partagées (PSK).
TLSCipherPSK13	Indiquer la chaîne de chiffrement pour OpenSSL 1.1.1 ou version ultérieure avec TLS 1.3. Remplacer les critères de sélection par défaut de la suite de chiffrement pour le chiffrement basé sur des clés pré-partagées (PSK).
TLSConnect	Indiquer le niveau de chiffrement requis pour les connexions au serveur ou au proxy Zabbix.
TLSCRLFile	Indiquer le chemin complet du fichier contenant les certificats révoqués. Ce paramètre est utilisé pour les communications chiffrées entre les composants Zabbix.

Paramètre	Description
TLSKeyFile	Indiquer le chemin complet du fichier contenant la clé privée de l'agent, utilisée pour les communications chiffrées entre les composants Zabbix.
TLSPSKFile	Indiquer le chemin complet du fichier contenant la clé pré-partagée de l'agent, utilisée pour les communications chiffrées avec le serveur ou le proxy Zabbix.
TLSPSKIdentity	Indiquer la chaîne d'identité de la clé pré-partagée, utilisée pour les communications chiffrées avec le serveur ou le proxy Zabbix.
TLSServerCertIssuer	Indiquer l'émetteur autorisé du certificat du serveur (proxy).
TLSServerCertSubject	Indiquer le sujet autorisé du certificat du serveur (proxy).
UnsafeUserParameters	Autoriser le passage de tous les caractères dans les arguments des paramètres définis par l'utilisateur.
User	Abandonner les privilèges pour utiliser un utilisateur spécifique existant sur le système.
UserParameter	Indiquer un paramètre défini par l'utilisateur à surveiller.
UserParameterDir	Indiquer le chemin de recherche par défaut des commandes UserParameter.

Notez que :

- Les valeurs par défaut correspondent aux valeurs par défaut du démon, et non aux valeurs des fichiers de configuration fournis.
- Les valeurs prennent en charge les variables d'environnement.
- Zabbix prend uniquement en charge les fichiers de configuration encodés en UTF-8 sans BOM.
- Les commentaires commençant par # sont pris en charge uniquement au début de la ligne.

Détails du paramètre

Alias

Définissez un alias pour une clé d'élément. Il peut être utilisé pour remplacer une clé d'élément longue et complexe par une clé plus courte et plus simple.
 Plusieurs paramètres `Alias` peuvent être présents. Plusieurs paramètres avec la même clé `Alias` ne sont pas autorisés.
 Différentes clés `Alias` peuvent faire référence à la même clé d'élément.
 Les alias peuvent être utilisés dans le paramètre `HostMetadataItem`, mais pas dans le paramètre `HostnameItem`.

Exemple 1 : récupération de l'ID de l'utilisateur `zabbix`.

```
Alias=zabbix.userid:vfs.file.regexp[/etc/passwd,"^zabbix:.:([0-9]+)",,,,\1]
```

La clé abrégée `zabbix.userid` peut maintenant être utilisée pour récupérer les données.

Exemple 2 : obtention de l'utilisation du processeur avec des paramètres par défaut et personnalisés.

```
Alias=cpu.util:system.cpu.util
Alias=cpu.util[*]:system.cpu.util[*]
```

Cela vous permet d'utiliser la clé `cpu.util` pour obtenir le pourcentage d'utilisation du processeur avec les paramètres par défaut, ainsi que `cpu.util[all, idle, avg15]` pour obtenir des données spécifiques sur l'utilisation du processeur.

Exemple 3 : exécution de plusieurs règles de découverte de bas niveau traitant les mêmes éléments de découverte.

```
Alias=vfs.fs.discovery[*]:vfs.fs.discovery
```

Il est maintenant possible de configurer plusieurs règles de découverte utilisant `vfs.fs.discovery` avec des paramètres différents pour chaque règle, par exemple `vfs.fs.discovery[foo]`, `vfs.fs.discovery[bar]`, etc.

AllowKey

Autorise l'exécution des clés d'élément qui correspondent à un modèle. Le modèle de clé est une expression générique qui prend en charge le caractère * pour faire correspondre n'importe quel nombre de caractères.
 Plusieurs règles de correspondance de clés peuvent être définies en combinaison avec `DenyKey`, `AllowKeyRegexp` et `DenyKeyRegexp`. Les paramètres sont traités un par un selon leur ordre d'apparition. L'ordre détermine quelle règle correspondante aura la priorité. Voir aussi : Restriction des vérifications de l'agent.

AllowKeyRegexp

Autorise l'exécution des clés d'éléments correspondant à un modèle. Le modèle de clé est une expression régulière.
 Plusieurs règles de correspondance de clés peuvent être définies en combinaison avec `DenyKeyRegexp`, `AllowKey` et `DenyKey`. Les paramètres sont traités un par un, selon leur ordre d'apparition. Cet ordre détermine quelle règle correspondante sera prioritaire. Voir également : Restriction des vérifications de l'agent.

Exemple :

```
AllowKeyRegexp=^system\.run\[(?:/usr)?/bin/hostname(?: -[fsd])?\]$
```

AllowRoot

Autoriser l'agent à s'exécuter en tant que `root`. Si cette option est désactivée et que l'agent est démarré par `root`, l'agent tentera plutôt de basculer vers l'utilisateur `zabbix`. Sans effet si l'agent est démarré sous un utilisateur standard.

Par défaut: 0
 Valeurs: 0 - ne pas autoriser; 1 - autoriser

BufferSend

Définissez la fréquence d'envoi des valeurs du tampon vers le serveur/proxy Zabbix (en secondes).

Par défaut: 5
 Plage: 1-3600

BufferSize

Définit le nombre maximal de valeurs dans le tampon mémoire.
 L'agent enverra toutes les données collectées au serveur Zabbix ou au proxy si le tampon est plein.

Par défaut: 100
 Plage: 2-65535

DebugLevel

Définissez le niveau de débogage : - *0* - informations de base sur le démarrage et l'arrêt des processus Zabbix ; - *1* - informations critiques ; - *2* - informations d'erreur ; - *3* - avertissements ; - *4* - pour le débogage (génère beaucoup d'informations) ; - *5* - débogage étendu (génère encore plus d'informations).

Par défaut : 3
 Plage : 0-5

DenyKey

Interdit l'exécution des clés d'élément correspondant à un modèle. Le modèle de clé est une expression générique qui prend en charge le caractère * pour correspondre à un nombre quelconque de caractères.
 Plusieurs règles de correspondance de clés peuvent être définies en combinaison avec `AllowKey`, `AllowKeyRegexp` et `DenyKeyRegexp`. Les paramètres sont traités un par un, selon leur ordre d'apparition. L'ordre détermine la règle correspondante prioritaire. Voir également : Restriction des vérifications de l'agent.

Par défaut : `DenyKey=system.run[*]`

DenyKeyRegexp

Interdire l'exécution des clés d'élément correspondant à un modèle. Le modèle de clé est une expression régulière.
 Plusieurs règles de correspondance de clés peuvent être définies en combinaison avec `AllowKeyRegexp`, `AllowKey` et `DenyKey`. Les paramètres sont traités un par un, selon leur ordre d'apparition. L'ordre détermine quelle règle correspondante sera prioritaire.
 Tous les éléments `system.run[]` sont interdits par défaut (`DenyKey=system.run[*]`), sauf si une autre règle `system.run[*]` est spécifiée.

Voir également : Restriction des vérifications de l'agent.

EnableRemoteCommands

Autoriser les commandes distantes depuis le serveur Zabbix.
 Ce paramètre est **obsolète** ; utilisez plutôt `AllowKey=system.run[*]`, `AllowKeyRegexp=^system\.run\[.*\]$`, `DenyKey=system.run[*]` ou `DenyKeyRegexp=^system\.run\[.*\]$`.
 Il s'agit d'un alias interne pour les paramètres `AllowKey`, `AllowKeyRegexp`, `DenyKey` et `DenyKeyRegexp`, selon la valeur : - 0 - `DenyKey=system.run[*]` (`DenyKeyRegexp=^system\.run\[.*\]$`) - 1 - `AllowKey=system.run[*]` (`AllowKeyRegexp=^system\.run\[.*\]$`)

Valeur par défaut : 0
 Valeurs : 0 - ne pas autoriser, 1 - autoriser

HeartbeatFrequency

Définissez la fréquence des messages de heartbeat en secondes. Utilisé pour surveiller la disponibilité des vérifications actives.
0 - messages de heartbeat désactivés.

Par défaut: 60
 Plage: 0-3600

HostInterface

Spécifiez l'interface de l'hôte (adresse IP ou nom DNS) utilisée lors de l'autoregistration de l'hôte. Cette valeur sera utilisée pour renseigner l'interface dans le nouvel hôte créé, et permet de configurer explicitement une adresse IP ou DNS. Pour plus de détails, voir Utiliser DNS comme interface par défaut.

Si elle n'est pas définie, la valeur sera obtenue à partir de `HostInterfaceItem`.

L'agent générera une erreur et ne démarrera pas si la valeur dépasse la limite de 255 caractères.

Plage : 0-255 caractères

HostInterfaceItem

Spécifiez un élément pour déterminer l'interface de l'hôte (adresse IP ou nom DNS) lors de l'enregistrement automatique de l'hôte. Cette valeur est utilisée uniquement si `HostInterface` n'est pas défini.
Pour plus d'informations, consultez Utilisation du DNS comme interface par défaut.

Lors d'une demande d'enregistrement automatique, l'agent consignera un message d'avertissement si la valeur renvoyée par l'élément spécifié dépasse la limite de 255 caractères.

L'élément `system.run[]` est pris en charge quels que soient les paramètres `AllowKey`, `AllowKeyRegexp`, `DenyKey` ou `DenyKeyRegexp`.

HostMetadata

Spécifiez les métadonnées permettant d'identifier ou de distinguer le hôte lors de l'enregistrement automatique du hôte (agent actif). `HostMetadata` permet de distinguer les hôtes au-delà du nom d'hôte.

Si cette valeur n'est pas définie, elle sera obtenue à partir de `HostMetadataItem`.

L'agent génère une erreur et ne démarre pas si la valeur spécifiée dépasse la limite de 2034 octets ou s'il s'agit d'une chaîne qui n'est pas au format UTF-8. Lorsqu'un paramètre attend une adresse IP ou un nom DNS, les valeurs qui sont valides au format UTF-8, mais qui ne sont pas des adresses IP ou des noms DNS valides, sont également rejetées et signalées comme non valides.

Les métadonnées multilignes ne sont pas prises en charge : la sortie est tronquée au premier saut de ligne.

Plage : 0-2034 octets

HostMetadataItem

Spécifiez un élément de l'agent Zabbix pour obtenir les métadonnées de l'hôte. Cette option est utilisée uniquement lorsque `HostMetadata` n'est pas défini.

La valeur de `HostMetadataItem` est récupérée à chaque tentative d'enregistrement automatique et utilisée uniquement lors du processus d'enregistrement automatique de l'hôte (agent actif). `HostMetadataItem` permet de distinguer les hôtes au-delà du nom d'hôte.

Les paramètres utilisateur et les alias sont pris en charge. L'élément `system.run[]` est pris en charge quels que soient les paramètres `AllowKey`, `AllowKeyRegexp`, `DenyKey` ou `DenyKeyRegexp`.

Lors d'une demande d'enregistrement automatique, l'agent consignera un avertissement si la valeur renvoyée par l'élément spécifié dépasse la limite de 65535 points de code UTF-8. La valeur renvoyée par l'élément doit être une chaîne UTF-8 ; sinon, elle sera ignorée. Lorsqu'un paramètre attend une adresse IP ou un nom DNS, les valeurs qui sont valides en UTF-8 mais qui ne sont pas des adresses IP ou des noms DNS valides seront également rejetées et signalées comme non valides.

Les métadonnées sur plusieurs lignes ne sont pas prises en charge : la sortie sera tronquée au premier saut de ligne.

Nom d'hôte

Spécifiez des noms d'hôte uniques, sensibles à la casse. Une liste délimitée par des virgules.
 Requis pour les vérifications actives et doit correspondre aux noms d'hôte tels qu'ils sont configurés sur le serveur.
 La valeur est obtenue à partir de `HostnameItem` si elle n'est pas définie.
Caractères autorisés: alphanumériques, '.', ' ', '_' et '-'. Longueur maximale: 128 caractères par nom d'hôte, 2048 caractères pour l'ensemble de la ligne.

Par défaut: défini par `HostnameItem`

HostnameItem

Spécifiez un élément de l'agent Zabbix permettant d'obtenir le nom d'hôte.
 Cette option est utilisée uniquement lorsque Hostname n'est pas défini.
 Les paramètres utilisateur et les alias ne sont pas pris en charge, mais l'élément `system.run[]` est pris en charge quelles que soient les valeurs de `AllowKey`, `AllowKeyRegexp`, `DenyKey` ou `DenyKeyRegexp`.

Valeur par défaut : `system.hostname`

Inclure

Spécifiez un répertoire ou des fichiers individuels à inclure dans le fichier de configuration.
 Lors de l'installation de Zabbix, ce paramètre est défini sur `/usr/local/etc`, sauf si vous le modifiez lors de la compilation.

Règles générales pour le paramètre `Include` :

- Le chemin peut être relatif à l'emplacement du fichier `zabbix_agentd.conf`.
- Vous pouvez spécifier plusieurs paramètres `Include` dans un même fichier de configuration.
- Zabbix traite le paramètre `Include` dès qu'il le lit, donc envisagez de le placer à la fin du fichier de configuration.

Règles pour les fichiers inclus :

- Chaque fichier doit être lisible.
- Chaque fichier doit avoir une syntaxe correcte. Sinon, le composant ne démarrera pas.

Règles pour l'inclusion de répertoires ou de modèles de fichiers :

- Ne supposez pas d'ordre d'inclusion particulier (par exemple, les fichiers ne sont pas inclus par ordre alphabétique).
- Ne définissez pas le même paramètre dans plusieurs fichiers inclus (par exemple, pour remplacer un paramètre général par un paramètre spécifique), car vous ne pouvez pas prévoir quelle valeur sera utilisée.
- Faites attention aux copies de sauvegarde que certains éditeurs de texte créent automatiquement (par exemple, la modification de `include/my_specific.conf` peut créer `include/my_specific.conf.BAK`), car les deux fichiers seront inclus.

Pour inclure uniquement les fichiers pertinents d'un répertoire, utilisez le caractère générique (`*`) pour la correspondance de modèle.

Exemple :

```
Include=/absolute/path/to/config/files/*.conf
```

ListenBacklog

Définit le nombre maximal de connexions en attente dans la file d'attente TCP.
 La valeur par défaut est une constante codée en dur, qui dépend du système.
 La valeur maximale prise en charge dépend également du système ; des valeurs trop élevées peuvent être tronquées silencieusement à la « valeur maximale spécifiée par l'implémentation ».

Par défaut: `SOMAXCONN`
 Plage: 0 - INT_MAX

ListenIP

Spécifiez l'adresse IP d'écoute de l'agent pour les connexions provenant du serveur/proxy. Liste séparée par des virgules.

Par défaut: `0.0.0.0`

ListenPort

Spécifiez le port d'écoute de l'agent pour les connexions provenant du serveur/proxy.

Par défaut: 10050
 Plage: 1024-32767

LoadModule

Spécifie le module à charger au démarrage de l'agent.
 Les modules sont utilisés pour étendre les fonctionnalités de l'agent.
 Le module doit se trouver dans le répertoire spécifié par `LoadModulePath` ou le chemin doit précéder le nom du module. Si le chemin précédent est absolu (commence par `/`), alors `LoadModulePath` est ignoré.
 Formats : - `LoadModule=<module.so>` - `LoadModule=<path/module.so>` - `LoadModule=</abs_path/module.so>`

Il est permis d'inclure plusieurs paramètres `LoadModule`.

LoadModulePath

Spécifiez le chemin complet vers l'emplacement des modules de l'agent.
 La valeur par défaut dépend des options de compilation.

LogFile

Spécifiez le fichier journal de l'agent.

Obligatoire : Oui, si `LogType` est défini sur `file` ; sinon non

LogFileSize

Définissez la taille maximale du fichier journal (en Mo).
0 - désactive la rotation automatique des journaux.
Remarque : si la limite de taille du fichier journal est atteinte et que la rotation du fichier échoue, quelle qu'en soit la raison, le fichier journal existant est tronqué et recréé.

Par défaut : 1
 Plage : 0-1024

LogRemoteCommands

Active la journalisation des commandes shell exécutées en tant qu'avertissements.
 Les commandes ne seront journalisées que si elles sont exécutées à distance.
 Aucune entrée de journal ne sera créée si `system.run[]` est lancé localement par les paramètres `HostMetadataItem`, `HostInterfaceItem` ou `HostnameItem`.

Par défaut: 0
 Valeurs: 0 - désactivé, 1 - activé

LogType

Spécifiez le type de sortie du journal : - *file* - écrire le journal dans le fichier spécifié par le paramètre LogFile ; - *system* - écrire le journal dans syslog ; - *console* - écrire le journal dans la sortie standard.

Par défaut : file

MaxLinesPerSecond

Définit le nombre maximal de nouvelles lignes par seconde à envoyer au serveur/proxy Zabbix lors du traitement des vérifications actives log ou logrt.
 La valeur fournie sera remplacée par le paramètre maxlines, indiqué dans la clé d'élément log ou logrt.
Remarque : Zabbix traitera 10 fois plus de nouvelles lignes que la valeur définie dans MaxLinesPerSecond afin de rechercher la chaîne requise dans les éléments de journal.

Par défaut : 20
 Plage : 1-1000

PidFile

Spécifiez le fichier PID.

Par défaut: /tmp/zabbix_agentd.pid

RefreshActiveChecks

Spécifiez la fréquence d'actualisation de la liste des vérifications actives, en secondes.
 Notez qu'après un échec d'actualisation des vérifications actives, la prochaine actualisation sera tentée dans 60 secondes.

Par défaut: 5
 Plage: 1-86400

Serveur

Spécifiez les adresses IP, éventuellement en notation CIDR, ou les noms DNS des serveurs/proxys Zabbix. Une liste séparée par des virgules.
 Les connexions entrantes seront acceptées uniquement depuis les hôtes répertoriés ici. Si la prise en charge d'IPv6 est activée, alors 127.0.0.1, ::127.0.0.1, ::ffff:127.0.0.1 sont traités de la même manière et ::/0 autorisera n'importe quelle adresse IPv4 ou IPv6. 0.0.0.0/0 peut être utilisé pour autoriser n'importe quelle adresse IPv4. Notez que les "adresses IPv6 compatibles IPv4" (préfixe 0000::/96) sont prises en charge mais obsolètes selon RFC4291. Les espaces sont autorisés.

Exemple :

```
Server=127.0.0.1,192.168.1.0/24,::1,2001:db8::/32,zabbix.example.com
```

Obligatoire : oui, si StartAgents n'est pas explicitement défini sur 0

ServerActive

Spécifiez l'adresse du serveur/proxy Zabbix ou la configuration du cluster pour obtenir les vérifications actives.
 L'adresse du serveur/proxy est une adresse IP ou un nom DNS, avec éventuellement un port séparé par deux-points.
La configuration du cluster est constituée d'une ou plusieurs adresses de membres du groupe de serveurs ou de proxy séparées par des points-virgules. Plusieurs serveurs/clusters Zabbix et proxies Zabbix peuvent être spécifiés, séparés par des virgules. Sauf en cas d'utilisation de groupes de proxy, il ne faut pas spécifier plus d'un proxy Zabbix pour chaque serveur/cluster Zabbix. Si un proxy Zabbix est spécifié, alors le serveur/cluster Zabbix correspondant à ce proxy ne doit pas être spécifié.
Plusieurs adresses séparées par des virgules peuvent être fournies pour utiliser plusieurs serveurs Zabbix indépendants en parallèle. Les espaces sont autorisés.
Si le port n'est pas spécifié, le port par défaut est utilisé.
Les adresses IPv6 doivent être placées entre crochets si le port de cet hôte est spécifié. Si le port n'est pas spécifié, les crochets pour les adresses IPv6 sont facultatifs.
Si ce paramètre n'est pas spécifié, les vérifications actives sont désactivées.

Exemple pour un proxy Zabbix:

```
ServerActive=127.0.0.1:10051
```

Exemple pour un groupe de proxy Zabbix:

```
ServerActive=proxy1.example.com;proxy2.example.com;proxy3.example.com;proxy4.example.com;proxy5.example.co
```

Exemple pour plusieurs serveurs:

```
ServerActive=127.0.0.1:20051,zabbix.domain,[::1]:30051,::1,[12fc::1]
```

Exemple pour la haute disponibilité:

```
ServerActive=zabbix.cluster.node1;zabbix.cluster.node2:20051;zabbix.cluster.node3
```

Exemple pour la haute disponibilité avec deux clusters et un serveur:

```
ServerActive=zabbix.cluster.node1;zabbix.cluster.node2:20051,zabbix.cluster2.node1;zabbix.cluster2.node2,z
```

SourceIP

Spécifiez l'adresse IP source pour :

- les connexions sortantes vers le serveur Zabbix ou le proxy Zabbix ;
- l'établissement de connexions lors de l'exécution de certains éléments (`web.page.get`, `net.tcp.port`, etc.).

StartAgents

Définit le nombre d'instances pré-forkées de `zabbix_agentd` pour le traitement des vérifications passives.
 Si la valeur est 0, les vérifications passives sont désactivées et l'agent n'écoutera sur aucun port TCP.

Valeur par défaut: 10
 Plage: 0-100

Délai d'attente

Spécifiez la durée d'attente, en secondes, pour établir la connexion et échanger des données avec le proxy ou le serveur Zabbix.

Ce paramètre définit la durée de diverses opérations de communication, notamment :

- l'attente d'une réponse du serveur Zabbix ;
- l'envoi de requêtes au serveur Zabbix, y compris les requêtes de configuration des éléments et les requêtes de données des éléments dans les vérifications actives ;
- la récupération des données de journal via logfile ;
- l'envoi de messages de heartbeat ;
- la durée maximale des vérifications `vfs.*` ;
- l'utilisation par les modules de l'agent Zabbix ;
- le rôle de solution de repli dans les scénarios où un serveur ou un proxy antérieur à la version 7.0 envoie des vérifications sans délais d'attente.

Ce délai d'attente **ne sera pas** utilisé pour les vérifications d'agent qui disposent de paramètres de délai d'attente configurables dans l'interface (au niveau global, du proxy ou de chaque élément).

Par défaut : 3
 Plage : 1-30

TLSAccept

Spécifiez le niveau de chiffrement requis pour les connexions entrantes.
 Utilisé pour les vérifications passives.
 Plusieurs valeurs peuvent être spécifiées, séparées par des virgules : - *unencrypted* - accepter les connexions sans chiffrement (par défaut) ; - *psk* - accepter les connexions avec TLS et une clé prépartagée (PSK) ; - *cert* - accepter les connexions avec TLS et un certificat.

Obligatoire : oui, si les paramètres du certificat TLS ou du PSK sont définis (même pour une connexion *unencrypted*) ; sinon non

TLSCAFile

Spécifiez le chemin d'accès complet du fichier contenant les certificats de l'autorité de certification racine pour la vérification du certificat du pair, utilisé pour les communications chiffrées entre les composants Zabbix.

TLSCertFile

Spécifiez le chemin d'accès complet du fichier contenant le certificat de l'agent ou la chaîne de certificats, utilisé pour les communications chiffrées avec les composants Zabbix.

TLSCipherAll

Spécifiez la chaîne de priorité GnuTLS ou la chaîne de suites de chiffrement OpenSSL (TLS 1.2). Remplace les critères de sélection par défaut des suites de chiffrement pour le chiffrement basé sur des certificats et sur PSK.

Exemple :

```
TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256
```

TLSCipherAll13

Spécifiez la chaîne de chiffrement pour OpenSSL 1.1.1 ou une version plus récente dans TLS 1.3. Remplacez les critères de sélection par défaut de la suite de chiffrement pour le chiffrement basé sur les certificats et sur PSK.

Exemple pour GnuTLS :

```
NONE:+VERS-TLS1.2:+ECDHE-RSA:+RSA:+ECDHE-PSK:+PSK:+AES-128-GCM:+AES-128-CBC:+AEAD:+SHA256:+SHA1:+CURVE-ALL
```

Exemple pour OpenSSL :

```
EECDH+aRSA+AES128:RSA+aRSA+AES128:kECDHEPSK+AES128:kPSK+AES128
```

TLSCipherCert

Spécifiez la chaîne de priorité GnuTLS ou la chaîne de chiffrement OpenSSL (TLS 1.2). Remplacez les critères de sélection de la suite de chiffrement par défaut pour le chiffrement basé sur les certificats.

Exemple pour GnuTLS :

```
NONE:+VERS-TLS1.2:+ECDHE-RSA:+RSA:+AES-128-GCM:+AES-128-CBC:+AEAD:+SHA256:+SHA1:+CURVE-ALL:+COMP-NULL:+SIG
```

Exemple pour OpenSSL :

```
EECDH+aRSA+AES128:RSA+aRSA+AES128
```

TLSCipherCert13

Spécifiez la chaîne de chiffrement pour OpenSSL 1.1.1 ou une version plus récente dans TLS 1.3. Remplacez les critères de sélection par défaut de la suite de chiffrement pour le chiffrement basé sur des certificats.

TLSCipherPSK

Spécifiez la chaîne de priorité GnuTLS ou la chaîne de chiffrement OpenSSL (TLS 1.2). Remplacez les critères de sélection de la suite de chiffrement par défaut pour le chiffrement basé sur PSK.

Exemple pour GnuTLS :

```
NONE:+VERS-TLS1.2:+ECDHE-PSK:+PSK:+AES-128-GCM:+AES-128-CBC:+AEAD:+SHA256:+SHA1:+CURVE-ALL:+COMP-NULL:+SIG
```

Exemple pour OpenSSL :

```
kECDHEPSK+AES128:kPSK+AES128
```

TLSCipherPSK13

Spécifiez la chaîne de chiffrement pour OpenSSL 1.1.1 ou une version plus récente en TLS 1.3. Remplacez les critères de sélection par défaut de la suite de chiffrement pour le chiffrement basé sur PSK.

Exemple :

```
TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256
```

TLSConnect

Spécifiez le niveau de chiffrement requis pour les connexions de l'agent au serveur Zabbix ou au proxy.
 Utilisé pour les vérifications actives.
 Une seule valeur peut être spécifiée : - *unencrypted* - connexion sans chiffrement (par défaut) ; - *psk* - connexion à l'aide de TLS et d'une clé prépartagée (PSK) ; - *cert* - connexion à l'aide de TLS et d'un certificat.

Obligatoire : oui, si des paramètres de certificat TLS ou de PSK sont définis (même pour une connexion *unencrypted*) ; sinon non

TLSCRLFile

Spécifiez le chemin d'accès complet du fichier contenant les certificats révoqués. Ce paramètre est utilisé pour les communications chiffrées entre les composants Zabbix.

TLSKeyFile

Indiquez le chemin d'accès complet du fichier contenant la clé privée de l'agent, utilisée pour les communications chiffrées entre les composants Zabbix.

TLSPSKFile

Spécifiez le chemin d'accès complet du fichier contenant la clé pré-partagée de l'agent, utilisée pour les communications chiffrées avec le serveur Zabbix ou le proxy.

TLSPSKIdentity

Spécifiez la chaîne d'identité de clé pré-partagée, utilisée pour les communications chiffrées avec le serveur Zabbix ou le proxy.

TLSServerCertIssuer

Spécifiez l'émetteur de certificat autorisé du serveur (proxy).

TLSServerCertSubject

Spécifiez le sujet du certificat autorisé du serveur (proxy).

UnsafeUserParameters

Autoriser tous les caractères à être transmis dans les arguments des paramètres définis par l'utilisateur. Les caractères suivants ne sont pas autorisés : \ ' " ` * ? [ ] { } ~ $! & ; ( ) < > | # @ %
De plus, les caractères de nouvelle ligne ne sont pas autorisés.

Par défaut : 0
 Valeurs : 0 - ne pas autoriser, 1 - autoriser

Utilisateur

Abaisser les privilèges vers un utilisateur spécifique existant sur le système.
Ne prend effet que si l'exécution se fait en tant que `root` et que `AllowRoot` est désactivé.

Par défaut: `zabbix`

UserParameter

Spécifiez un paramètre défini par l'utilisateur à surveiller. Il peut y avoir plusieurs paramètres définis par l'utilisateur.
Format : `UserParameter=<key>,<shell command>`
Notez que la commande shell ne doit pas renvoyer une chaîne vide ni uniquement un EOL. Les commandes shell peuvent utiliser des chemins relatifs si le paramètre `UserParameterDir` est spécifié.

Exemple :

```
UserParameter=system.test,who|wc -l
UserParameter=check_cpu,./custom_script.sh
```

UserParameterDir

Spécifie le chemin de recherche par défaut pour les commandes `UserParameter`. Si cette option est utilisée, l'agent changera son répertoire de travail vers celui indiqué ici avant d'exécuter une commande. Ainsi, les commandes `UserParameter` peuvent utiliser un préfixe relatif `./` au lieu d'un chemin complet.
Une seule entrée est autorisée.

Exemple :

```
UserParameterDir=/opt/myscripts
```

Voir aussi

1. Différences dans la configuration de l'agent Zabbix pour les vérifications actives et passives à partir de la version 2.0.0

Paramètres de configuration: Zabbix agent (Windows)

Vue d'ensemble

Les paramètres de configuration permettent de personnaliser le comportement de l'agent Zabbix sous Windows.

Indiquez les valeurs des paramètres dans le fichier de configuration de l'agent (`zabbix_agentd.conf`). Tous les paramètres sont **facultatifs**, sauf indication explicite contraire.

Cette page est structurée comme suit :

- Vue d'ensemble des paramètres (pour afficher tous les détails d'un paramètre, cliquez sur son nom).
- Détails des paramètres.

Paramètre	Description
Alias	Définir un alias pour une clé d'élément.
AllowKey	Autoriser l'exécution des clés d'élément correspondant à un modèle.
AllowKeyRegexp	Autoriser l'exécution des clés d'élément correspondant à un modèle basé sur une expression régulière.
BufferSend	Définir la fréquence d'envoi des valeurs de la mémoire tampon au serveur/proxy Zabbix.
BufferSize	Définir le nombre maximal de valeurs dans la mémoire tampon.
DebugLevel	Définir le niveau de débogage.
DenyKey	Interdire l'exécution des clés d'élément correspondant à un modèle.
DenyKeyRegexp	Interdire l'exécution des clés d'élément correspondant à un modèle basé sur une expression régulière.
EnableRemoteCommands	Autoriser les commandes distantes provenant du serveur Zabbix. Ce paramètre est obsolète ; utilisez `AllowKey`, `AllowKeyRegexp`, `DenyKey` ou `DenyKeyRegexp` à la place.
HeartbeatFrequency	Définir la fréquence des messages de pulsation, en secondes.
HostInterface	Indiquer l'interface du hôte.
HostInterfaceItem	Indiquer un élément permettant d'obtenir l'interface du hôte.
HostMetadata	Indiquer les métadonnées du hôte.
HostMetadataItem	Indiquer un élément de l'agent Zabbix permettant d'obtenir les métadonnées du hôte.
Hostname	Indiquer le nom d'hôte.
HostnameItem	Indiquer un élément de l'agent Zabbix permettant d'obtenir le nom d'hôte.
Include	Indiquer un répertoire ou des fichiers individuels à inclure dans le fichier de configuration.
ListenBacklog	Définir le nombre maximal de connexions en attente dans la file TCP.

Paramètre	Description
ListenIP	Indiquer l'adresse IP d'écoute de l'agent pour les connexions provenant du serveur/proxy.
ListenPort	Indiquer le port d'écoute de l'agent.
LogFile	Indiquer le fichier journal de l'agent.
LogFileSize	Définir la taille maximale du fichier journal.
LogRemoteCommands	Activer la journalisation des commandes shell exécutées sous forme d'avertissements.
LogType	Indiquer le type de sortie du journal.
MaxLinesPerSecond	Définir le nombre maximal de nouvelles lignes par seconde à envoyer au serveur/proxy Zabbix lors du traitement des contrôles actifs `log`, `logrt` ou `eventlog`.
PerfCounter	Définir un nouveau paramètre <parameter_name> correspondant à la valeur moyenne du compteur de performances système <perf_counter_path> pour la période spécifiée <period> (en secondes).
PerfCounterEn	Définir un nouveau paramètre <parameter_name> correspondant à la valeur moyenne du compteur de performances système <perf_counter_path> pour la période spécifiée <period> (en secondes). Contrairement à PerfCounter, les chemins des compteurs de performances doivent être en anglais.
RefreshActiveChecks	Indiquer la fréquence de mise à jour de la liste des contrôles actifs.
Server	Indiquer les adresses IP ou les noms DNS des serveurs/proxys Zabbix.
ServerActive	Indiquer l'adresse du serveur/proxy Zabbix ou la configuration du cluster à partir de laquelle obtenir les contrôles actifs.
SourceIP	Indiquer l'adresse IP source.
StartAgents	Définir le nombre d'instances précréées de `zabbix_agentd` pour le traitement des contrôles passifs.
Timeout	Indiquer la durée d'attente (en secondes) pour établir la connexion et échanger des données avec le serveur ou le proxy Zabbix.
TLSAccept	Indiquer le niveau de chiffrement requis pour les connexions entrantes.
TLSCAFile	Indiquer le chemin complet du fichier contenant les certificats des autorités de certification de niveau supérieur pour la vérification du certificat du pair, utilisés pour les communications chiffrées entre les composants Zabbix.
TLSCertFile	Indiquer le chemin complet du fichier contenant le certificat ou la chaîne de certificats de l'agent, utilisés pour les communications chiffrées entre les composants Zabbix.
TLSConnect	Indiquer le niveau de chiffrement requis pour les connexions au serveur ou au proxy Zabbix.
TLSCRLFile	Indiquer le chemin complet du fichier contenant les certificats révoqués. Ce paramètre est utilisé pour les communications chiffrées entre les composants Zabbix.
TLSKeyFile	Indiquer le chemin complet du fichier contenant la clé privée de l'agent, utilisée pour les communications chiffrées entre les composants Zabbix.
TLSPSKFile	Indiquer le chemin complet du fichier contenant la clé prépartagée de l'agent, utilisée pour les communications chiffrées avec le serveur ou le proxy Zabbix.
TLSPSKIdentity	Indiquer la chaîne d'identité de la clé prépartagée, utilisée pour les communications chiffrées avec le serveur ou le proxy Zabbix.
TLSServerCertIssuer	Indiquer l'émetteur autorisé du certificat du serveur (proxy).
TLSServerCertSubject	Indiquer le sujet autorisé du certificat du serveur (proxy).
UnsafeUserParameters	Autoriser le passage de tous les caractères dans les arguments des paramètres définis par l'utilisateur.
UserParameter	Indiquer un paramètre défini par l'utilisateur à surveiller.
UserParameterDir	Indiquer le chemin de recherche par défaut des commandes UserParameter.

Notez que :

- Les valeurs par défaut correspondent aux valeurs par défaut du démon, et non aux valeurs des fichiers de configuration fournis.
- Les valeurs prennent en charge les variables d'environnement.
- Zabbix prend uniquement en charge les fichiers de configuration encodés en UTF-8 sans BOM.
- Les commentaires commençant par # sont pris en charge uniquement au début de la ligne.

Détails du paramètre

Alias

Définissez un alias pour une clé d'élément. Il peut être utilisé pour remplacer une clé d'élément longue et complexe par une clé plus courte et plus simple.
 Plusieurs paramètres `Alias` peuvent être présents. Plusieurs paramètres avec la même clé `Alias` ne sont pas autorisés.
 Différentes clés `Alias` peuvent faire référence à la même clé d'élément.
 Les alias peuvent être utilisés dans `HostMetadataItem`, mais pas dans les paramètres `HostnameItem` ou `PerfCounter`.

Exemple 1 : récupération de l'utilisation du fichier d'échange en pourcentage depuis le serveur.

```
Alias=pg_usage:perf_counter[\Paging File(_Total)\% Usage]
```

La clé abrégée pg_usage peut maintenant être utilisée pour récupérer les données.

Exemple 2 : obtention de la charge du processeur avec des paramètres par défaut et personnalisés.

```
Alias=cpu.load:system.cpu.load
Alias=cpu.load[*]:system.cpu.load[*]
```

Cela vous permet d'utiliser la clé cpu.load pour obtenir la charge du processeur avec les paramètres par défaut, ainsi que cpu.load[percpu,avg15] pour obtenir des données spécifiques sur la charge du processeur.

Exemple 3 : exécution de plusieurs règles de découverte de bas niveau traitant les mêmes éléments de découverte.

```
Alias=vfs.fs.discovery[*]:vfs.fs.discovery
```

Il est maintenant possible de configurer plusieurs règles de découverte à l'aide de vfs.fs.discovery, avec des paramètres différents pour chaque règle, par exemple vfs.fs.discovery[foo], vfs.fs.discovery[bar], etc.

AllowKey

Autorise l'exécution des clés d'élément qui correspondent à un modèle. Le modèle de clé est une expression générique qui prend en charge le caractère * pour correspondre à n'importe quel nombre de caractères.
 Plusieurs règles de correspondance de clés peuvent être définies en combinaison avec DenyKey, AllowKeyRegexp et DenyKeyRegexp. Les paramètres sont traités un par un selon leur ordre d'apparition. L'ordre détermine quelle règle correspondante aura la priorité. Voir aussi : Restricting agent checks.

AllowKeyRegexp

Autorise l'exécution des clés d'éléments correspondant à un modèle. Le modèle de clé est une expression régulière.
 Plusieurs règles de correspondance de clés peuvent être définies en combinaison avec DenyKeyRegexp, AllowKey et DenyKey. Les paramètres sont traités un par un, selon leur ordre d'apparition. Cet ordre détermine quelle règle correspondante sera prioritaire. Voir également : Restriction des vérifications de l'agent.

Exemple :

```
AllowKeyRegexp=^system\.run\[(?:/usr)?/bin/hostname(?: -[fsd])?\]$
```

BufferSend

Définissez la fréquence d'envoi des valeurs du buffer vers le serveur/proxy Zabbix (en secondes).

Par défaut: 5
 Plage: 1-3600

BufferSize

Définit le nombre maximal de valeurs dans le tampon mémoire.
 L'agent enverra toutes les données collectées au serveur Zabbix ou au proxy si le tampon est plein.

Par défaut: 100
 Plage: 2-65535

DebugLevel

Définissez le niveau de débogage : - *0* - informations de base sur le démarrage et l'arrêt des processus Zabbix ; - *1* - informations critiques ; - *2* - informations d'erreur ; - *3* - avertissements ; - *4* - pour le débogage (génère beaucoup d'informations) ; - *5* - débogage étendu (génère encore plus d'informations).

Par défaut : 3
 Plage : 0-5

DenyKey

Interdit l'exécution des clés d'élément correspondant à un modèle. Le modèle de clé est une expression générique qui prend en charge le caractère * pour correspondre à un nombre quelconque de caractères.
 Plusieurs règles de correspondance de clés peuvent être définies en combinaison avec AllowKey, AllowKeyRegexp et DenyKeyRegexp. Les paramètres sont traités un par un, selon leur ordre d'apparition. Cet ordre détermine quelle règle correspondante est prioritaire. Voir également : Restriction des vérifications de l'agent.

Par défaut : DenyKey=system.run[*]

DenyKeyRegexp

Interdit l'exécution des clés d'élément correspondant à un modèle. Le modèle de clé est une expression régulière.
 Plusieurs règles de correspondance de clés peuvent être définies en combinaison avec AllowKeyRegexp, AllowKey et DenyKey. Les

paramètres sont traités un par un, selon leur ordre d'apparition. Cet ordre détermine quelle règle correspondante est prioritaire.
 Tous les éléments `system.run[]` sont interdits par défaut (`DenyKey=system.run[*]`), sauf si une autre règle `system.run[*]` est spécifiée.

Voir aussi : Restriction des vérifications de l'agent.

EnableRemoteCommands

Autoriser les commandes distantes depuis le serveur Zabbix.
 Ce paramètre est **obsolète** ; utilisez plutôt `AllowKey=system.run[*]`, `AllowKeyRegexp=^system\.run\[.*\]$`, `DenyKey=system.run[*]` ou `DenyKeyRegexp=^system\.run\[.*\]$`.
 Il s'agit d'un alias interne pour les paramètres `AllowKey`, `AllowKeyRegexp`, `DenyKey` et `DenyKeyRegexp`, selon la valeur : - 0 - `DenyKey=system.run[*]` (`DenyKeyRegexp=^system\.run\[.*\]$`) - 1 - `AllowKey=system.run[*]` (`AllowKeyRegexp=^system\.run\[.*\]$`)

Valeur par défaut : 0
 Valeurs : 0 - ne pas autoriser, 1 - autoriser

HeartbeatFrequency

Définissez la fréquence des messages de heartbeat en secondes. Utilisé pour surveiller la disponibilité des vérifications actives.
0 - messages de heartbeat désactivés.

Par défaut: 60
 Plage: 0-3600

HostInterface

Spécifiez l'interface de l'hôte (adresse IP ou nom DNS) utilisée lors de l'autoregistration de l'hôte. Cette valeur sera utilisée pour renseigner l'interface dans le nouvel hôte créé, et permet de configurer explicitement une adresse IP ou DNS. Pour plus de détails, voir Using DNS as default interface.

Si elle n'est pas définie, la valeur sera récupérée depuis HostInterfaceItem.

L'agent renverra une erreur et ne démarrera pas si la valeur dépasse la limite de 255 caractères.

Plage: 0-255 caractères

HostInterfaceItem

Spécifiez un élément pour déterminer l'interface du hôte (adresse IP ou nom DNS) lors de l'enregistrement automatique du hôte. Cette valeur est utilisée uniquement si `HostInterface` n'est pas défini. Pour plus de détails, consultez Utilisation du DNS comme interface par défaut.

Lors d'une demande d'enregistrement automatique, l'agent consignera un message d'avertissement si la valeur renvoyée par l'élément spécifié dépasse la limite de 255 caractères.

L'élément `system.run[]` est pris en charge quels que soient les paramètres `AllowKey`, `AllowKeyRegexp`, `DenyKey` ou `DenyKeyRegexp`.

HostMetadata

Spécifiez les métadonnées utilisées pour identifier ou distinguer le hôte lors de l'enregistrement automatique du hôte (agent actif). `HostMetadata` permet de distinguer les hôtes au-delà du nom d'hôte.

Si elle n'est pas définie, la valeur sera obtenue à partir de `HostMetadataItem`.

L'agent générera une erreur et ne démarrera pas si la valeur spécifiée dépasse la limite de 2034 octets ou s'il s'agit d'une chaîne qui n'est pas au format UTF-8. Lorsqu'un paramètre attend une adresse IP ou un nom DNS, les valeurs qui sont valides au format UTF-8, mais qui ne sont pas des adresses IP ou des noms DNS valides, seront également rejetées et signalées comme non valides.

Les métadonnées multilignes ne sont pas prises en charge : la sortie sera tronquée au premier saut de ligne.

Plage : 0-2034 octets

HostMetadataItem

Spécifiez un élément de l'agent Zabbix pour obtenir les métadonnées de l'hôte. Cette option est utilisée uniquement lorsque `HostMetadata` n'est pas défini.

La valeur de `HostMetadataItem` est récupérée à chaque tentative d'enregistrement automatique et utilisée uniquement lors du processus d'enregistrement automatique de l'hôte (agent actif). `HostMetadataItem` permet de distinguer les hôtes au-delà du nom d'hôte.

Les paramètres utilisateur, les compteurs de performance et les alias sont pris en charge. L'élément `system.run[]` est pris en charge quels que soient les paramètres `AllowKey`, `AllowKeyRegexp`, `DenyKey` ou `DenyKeyRegexp`.

Lors d'une demande d'enregistrement automatique, l'agent consigne un avertissement si la valeur renvoyée par l'élément spécifié dépasse la limite de 65535 points de code UTF-8. La valeur renvoyée par l'élément doit être une chaîne UTF-8 ; sinon, elle

est ignorée. Lorsqu'un paramètre attend une adresse IP ou un nom DNS, les valeurs qui sont valides en UTF-8 mais qui ne correspondent pas à des adresses IP ou à des noms DNS valides sont également rejetées et signalées comme non valides.

Les métadonnées multilignes ne sont pas prises en charge : la sortie est tronquée au premier saut de ligne.

Nom d'hôte

Spécifiez des noms d'hôte uniques, sensibles à la casse. Une liste délimitée par des virgules.
 Requis pour les vérifications actives et doit correspondre aux noms d'hôte tels qu'ils sont configurés sur le serveur. La valeur est obtenue à partir de `HostnameItem` si elle n'est pas définie.
Caractères autorisés: alphanumériques, '.', ' ', '_' et '-'. Longueur maximale: 128 caractères par nom d'hôte, 2048 caractères pour l'ensemble de la ligne.

Par défaut: défini par HostnameItem

HostnameItem

Spécifiez un élément de l'agent Zabbix permettant d'obtenir le nom d'hôte.
 Cette option est utilisée uniquement lorsque `Hostname` n'est pas défini.
 Les paramètres utilisateur, les compteurs de performance et les alias ne sont pas pris en charge, mais l'élément `system.run[]` est pris en charge quelles que soient les valeurs de `AllowKey`, `AllowKeyRegexp`, `DenyKey` ou `DenyKeyRegexp`.
Voir également une description plus détaillée.

Valeur par défaut : `system.hostname`

Include

Spécifiez un répertoire ou des fichiers individuels à inclure dans le fichier de configuration (situé par défaut dans `C:\Program Files\Zabbix Agent` si l'agent Zabbix est installé à l'aide des packages d'installation MSI pour Windows ; situé dans le dossier spécifié lors de l'installation si l'agent Zabbix est installé sous forme d'archive zip).

Règles générales pour le paramètre `Include` :

- Le chemin peut être relatif à l'emplacement du fichier `zabbix_agentd.conf`.
- Vous pouvez spécifier plusieurs paramètres `Include` dans un même fichier de configuration.
- Zabbix traite le paramètre `Include` dès sa lecture. Il est donc recommandé de le placer à la fin du fichier de configuration.

Règles applicables aux fichiers inclus :

- Chaque fichier doit être lisible.
- Chaque fichier doit avoir une syntaxe correcte. Dans le cas contraire, le composant ne démarrera pas.

Règles applicables à l'inclusion de répertoires ou de modèles de fichiers :

- Ne présumez pas d'un ordre d'inclusion particulier (par exemple, les fichiers ne sont pas inclus dans l'ordre alphabétique).
- Ne définissez pas le même paramètre dans plusieurs fichiers inclus (par exemple, pour remplacer un paramètre général par un paramètre spécifique), car vous ne pouvez pas prévoir quelle valeur sera utilisée.
- Faites attention aux copies de sauvegarde que certains éditeurs de texte créent automatiquement (par exemple, la modification de `include/my_specific.conf` peut créer `include/my_specific.conf.BAK`), car les deux fichiers seront inclus.

Pour inclure uniquement les fichiers pertinents d'un répertoire, utilisez le caractère générique (`*`) pour la correspondance de modèles.

Exemple :

```
Include=C:\Program Files\Zabbix Agent\zabbix_agentd.d\*.conf
```

ListenBacklog

Définit le nombre maximal de connexions en attente dans la file d'attente TCP.
 La valeur par défaut est une constante codée en dur, qui dépend du système.
 La valeur maximale prise en charge dépend également du système ; des valeurs trop élevées peuvent être tronquées silencieusement à la « valeur maximale spécifiée par l'implémentation ».

Par défaut: `SOMAXCONN`
 Plage: 0 - INT_MAX

ListenIP

Spécifiez l'adresse IP d'écoute de l'agent pour les connexions provenant du serveur/proxy. Liste séparée par des virgules.

Par défaut: `0.0.0.0`

ListenPort

Spécifiez le port d'écoute de l'agent pour les connexions provenant du serveur/proxy.

Par défaut: 10050
 Plage: 1024-32767

LogFile

Spécifiez le fichier journal de l'agent.

Par défaut: `c:\zabbix_agentd.log`
 Obligatoire: Oui, si `LogType` est défini sur `file`; sinon non

LogFileSize

Définissez la taille maximale du fichier journal (en Mo).
0 - désactive la rotation automatique des journaux.
Remarque : si la limite de taille du fichier journal est atteinte et que la rotation du fichier échoue, quelle qu'en soit la raison, le fichier journal existant est tronqué et recréé.

Par défaut : 1
 Plage : 0-1024

LogRemoteCommands

Active l'enregistrement des commandes shell exécutées en tant qu'avertissements.
 Les commandes ne seront consignées que si elles sont exécutées à distance.
 Aucune entrée de journal ne sera créée si `system.run[]` est lancé localement par les paramètres `HostMetadataItem`, `HostInterfaceItem` ou `HostnameItem`.

Par défaut: 0
 Valeurs: 0 - désactivé, 1 - activé

LogType

Spécifiez le type de sortie du journal : - *file* - écrire le journal dans le fichier spécifié par le paramètre `LogFile` ; - *system* - écrire le journal dans le journal des événements Windows ; - *console* - écrire le journal sur la sortie standard.

Par défaut : `file`

MaxLinesPerSecond

Définit le nombre maximal de nouvelles lignes par seconde à envoyer au serveur/proxy Zabbix lors du traitement des vérifications actives `log`, `logrt` ou `eventlog`.
 La valeur fournie sera remplacée par le paramètre `maxlines`, indiqué dans la clé d'élément `log`, `logrt` ou `eventlog`.
Note : Zabbix traitera 10 fois plus de nouvelles lignes que la valeur définie dans `MaxLinesPerSecond` afin de rechercher la chaîne requise dans les éléments de journal.

Par défaut : 20
 Plage : 1-1000

PerfCounter

Définissez un nouveau paramètre <parameter_name> qui correspond à la valeur moyenne du compteur de performance système <perf_counter_path> pour la période spécifiée <period> (en secondes).
Syntaxe: <parameter_name>,"<perf_counter_path>",<period>

Par exemple, si vous souhaitez recevoir le nombre moyen d'interruptions du processeur par seconde pour la dernière minute, vous pouvez définir un nouveau paramètre "interrupts" comme suit :

```
PerfCounter = interrupts,"\Processor(0)\Interrupts/sec",60
```

Veuillez noter les guillemets doubles autour du chemin du compteur de performance. Le nom du paramètre (interrupts) doit être utilisé comme clé d'élément lors de la création d'un élément. Les échantillons utilisés pour calculer la valeur moyenne seront prélevés chaque seconde.
Vous pouvez exécuter `typeperf -qx` pour obtenir la liste de tous les compteurs de performance disponibles dans Windows.

PerfCounterEn

Définissez un nouveau paramètre <parameter_name> qui correspond à la valeur moyenne du compteur de performance système <perf_counter_path> pour la période spécifiée <period> (en secondes). Par rapport à PerfCounter, les chemins de compteur de performance doivent être en anglais. Pris en charge uniquement sur **Windows Server 2008/Vista** et versions ultérieures.
Syntaxe: <parameter_name>,"<perf_counter_path>",<period>

Par exemple, si vous souhaitez recevoir le nombre moyen d'interruptions du processeur par seconde pour la dernière minute, vous pouvez définir un nouveau paramètre "interrupts" comme suit:

```
PerfCounterEn = interrupts,"\Processor(0)\Interrupts/sec",60
```

Veuillez noter les guillemets doubles autour du chemin du compteur de performance. Le nom du paramètre (interrupts) doit être utilisé comme clé de l'élément lors de la création d'un élément. Les échantillons utilisés pour calculer la valeur moyenne seront prélevés chaque seconde.
Vous pouvez trouver la liste des chaînes en anglais en consultant la clé de registre suivante: `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Perflib\009`.

RefreshActiveChecks

Spécifiez la fréquence d'actualisation de la liste des vérifications actives, en secondes.
 Notez qu'après un échec d'actualisation des vérifications actives, la prochaine actualisation sera tentée dans 60 secondes.

Par défaut: 5
 Plage: 1-86400

Serveur

Spécifiez les adresses IP, éventuellement en notation CIDR, ou les noms DNS des serveurs/proxys Zabbix. Liste séparée par des virgules.
 Les connexions entrantes ne seront acceptées que depuis les hôtes listés ici. Si la prise en charge d'IPv6 est activée, alors `127.0.0.1`, `::127.0.0.1`, `::ffff:127.0.0.1` sont traités de la même manière et `::/0` autorisera toute adresse IPv4 ou IPv6. `0.0.0.0/0` peut être utilisé pour autoriser toute adresse IPv4. Notez que les "adresses IPv6 compatibles IPv4" (`préfixe 0000::/96`) sont prises en charge mais obsolètes selon RFC4291. Les espaces sont autorisés.

Exemple :

```
Server=127.0.0.1,192.168.1.0/24,::1,2001:db8::/32,zabbix.example.com
```

Obligatoire : oui, si StartAgents n'est pas explicitement défini sur 0

ServerActive

Spécifie l'adresse du serveur/proxy Zabbix ou la configuration de cluster à partir de laquelle obtenir les vérifications actives.
 L'adresse du serveur/proxy est une adresse IP ou un nom DNS, avec éventuellement un port séparé par deux-points.
La configuration de cluster est constituée d'une ou plusieurs adresses de membres de groupe de serveur ou de proxy séparées par un point-virgule. Plusieurs serveurs/clusters Zabbix et proxies Zabbix peuvent être spécifiés, séparés par une virgule. Sauf en cas d'utilisation de groupes de proxy, il ne faut pas spécifier plus d'un proxy Zabbix pour chaque serveur/cluster Zabbix. Si un proxy Zabbix est spécifié, alors le serveur/cluster Zabbix correspondant à ce proxy ne doit pas être spécifié.
Plusieurs adresses séparées par des virgules peuvent être fournies pour utiliser plusieurs serveurs Zabbix indépendants en parallèle. Les espaces sont autorisés.
Si le port n'est pas spécifié, le port par défaut est utilisé.
Les adresses IPv6 doivent être placées entre crochets si le port de cet hôte est spécifié. Si le port n'est pas spécifié, les crochets pour les adresses IPv6 sont facultatifs.
Si ce paramètre n'est pas spécifié, les vérifications actives sont désactivées.

Exemple pour un proxy Zabbix :

```
ServerActive=127.0.0.1:10051
```

Exemple pour un groupe de proxy Zabbix :

```
ServerActive=proxy1.example.com;proxy2.example.com;proxy3.example.com;proxy4.example.com;proxy5.example.co
```

Exemple pour plusieurs serveurs :

```
ServerActive=127.0.0.1:20051,zabbix.domain,[::1]:30051,::1,[12fc::1]
```

Exemple pour la haute disponibilité :

```
ServerActive=zabbix.cluster.node1;zabbix.cluster.node2:20051;zabbix.cluster.node3
```

Exemple pour la haute disponibilité avec deux clusters et un serveur :

```
ServerActive=zabbix.cluster.node1;zabbix.cluster.node2:20051,zabbix.cluster2.node1;zabbix.cluster2.node2,z
```

Plage : Le nombre de serveurs actifs listés dans `ServerActive` plus le nombre d'instances pré-forkées pour les vérifications passives spécifiées dans `StartAgents` doit être inférieur à 64.

SourceIP

Spécifiez l'adresse IP source pour :

- les connexions sortantes vers le serveur Zabbix ou le proxy Zabbix ;
- l'établissement de connexions lors de l'exécution de certains éléments (`web.page.get`, `net.tcp.port`, etc.).

StartAgents

Définit le nombre d'instances pré-forkées de `zabbix_agentd` pour le traitement des vérifications passives.
 Si la valeur est définie sur 0, les vérifications passives sont désactivées et l'agent n'écoutera sur aucun port TCP.

Par défaut: 10
 Plage: 0-100 (*)

Timeout

Spécifiez la durée d'attente (en secondes) pour établir la connexion et échanger des données avec le proxy ou le serveur Zabbix.

Ce paramètre définit la durée de diverses opérations de communication, notamment :

- l'attente d'une réponse du serveur Zabbix ;
- l'envoi de requêtes au serveur Zabbix, y compris les requêtes de configuration des éléments et les données d'élément dans les vérifications actives ;
- la récupération des données de journal via la surveillance des fichiers journaux ou du journal des événements Windows ;
- l'envoi de messages de heartbeat ;
- la durée maximale des vérifications `vfs.*` ;

- l'utilisation par les modules de l'agent Zabbix ;
- le rôle de solution de repli dans les scénarios où un serveur ou un proxy antérieur à la version 7.0 envoie des vérifications sans délais d'attente.

Ce délai d'attente ne sera **pas** utilisé pour les vérifications d'agent qui disposent de paramètres de délai d'attente configurables dans l'interface (au niveau global, du proxy ou de chaque élément).

Par défaut : 3
 Plage : 1-30

TLSAccept

Spécifiez le niveau de chiffrement requis pour les connexions entrantes.
 Utilisé pour les vérifications passives.
 Plusieurs valeurs peuvent être spécifiées, séparées par des virgules : - *unencrypted* - accepter les connexions sans chiffrement (par défaut) ; - *psk* - accepter les connexions avec TLS et une clé prépartagée (PSK) ; - *cert* - accepter les connexions avec TLS et un certificat.

Obligatoire : oui, si les paramètres du certificat TLS ou de la PSK sont définis (même pour une connexion *unencrypted*) ; sinon non

TLSCAFile

Spécifiez le chemin complet du fichier contenant le ou les certificats de l'autorité de certification racine pour la vérification du certificat du pair, utilisé pour les communications chiffrées entre les composants Zabbix.

TLSCertFile

Spécifiez le chemin d'accès complet du fichier contenant le certificat de l'agent ou la chaîne de certificats, utilisé pour les communications chiffrées avec les composants Zabbix.

TLSConnect

Spécifiez le niveau de chiffrement requis pour les connexions de l'agent au serveur Zabbix ou au proxy.
 Utilisé pour les vérifications actives.
 Une seule valeur peut être spécifiée : - *unencrypted* - se connecter sans chiffrement (par défaut) ; - *psk* - se connecter en utilisant TLS et une clé prépartagée (PSK) ; - *cert* - se connecter en utilisant TLS et un certificat.

Obligatoire : oui, si des paramètres de certificat TLS ou de PSK sont définis (même pour une connexion *unencrypted*) ; sinon non

TLSCRLFile

Spécifiez le chemin d'accès complet du fichier contenant les certificats révoqués. Ce paramètre est utilisé pour les communications chiffrées entre les composants Zabbix.

TLSKeyFile

Spécifiez le chemin d'accès complet du fichier contenant la clé privée de l'agent, utilisée pour les communications chiffrées entre les composants Zabbix.

TLSPSKFile

Spécifiez le chemin d'accès complet du fichier contenant la clé prépartagée de l'agent, utilisée pour les communications chiffrées avec le serveur Zabbix ou le proxy.

TLSPSKIdentity

Spécifiez la chaîne d'identité de la clé pré-partagée, utilisée pour les communications chiffrées avec le serveur ou le proxy Zabbix.

TLSServerCertIssuer

Spécifiez l'émetteur du certificat du serveur autorisé (proxy).

TLSServerCertSubject

Spécifiez le sujet de certificat du serveur (proxy) autorisé.

UnsafeUserParameters

Autoriser tous les caractères à être transmis dans les arguments des paramètres définis par l'utilisateur. Les caractères suivants ne sont pas autorisés : \ ' " ` * ? [ ] { } ~ $! & ; ( ) < > | # @ %
De plus, les caractères de nouvelle ligne ne sont pas autorisés.

Par défaut : 0
 Valeurs : 0 - ne pas autoriser, 1 - autoriser

UserParameter

Spécifiez un paramètre défini par l'utilisateur à surveiller. Il peut y avoir plusieurs paramètres définis par l'utilisateur.
Format: `UserParameter=<key>,<shell command>`
Notez que la commande shell ne doit pas renvoyer une chaîne vide ni uniquement un EOL. Les commandes shell peuvent utiliser des chemins relatifs si le paramètre `UserParameterDir` est spécifié.

Exemple:

```
UserParameter=system.test,dir /b | find /c /v ""
UserParameter=check_cpu,.\custom\_script.bat
```

UserParameterDir

Spécifie le chemin de recherche par défaut pour les commandes `UserParameter`. Si cette option est utilisée, l'agent changera son répertoire de travail vers celui indiqué ici avant d'exécuter une commande. Ainsi, les commandes `UserParameter` peuvent utiliser un préfixe relatif `./` au lieu d'un chemin complet. Une seule entrée est autorisée.

Exemple :

```
UserParameterDir=C:\opt\myscripts
```

Zabbix agent sur Microsoft Windows

Configuration de l'agent

Les deux générations d'agents Zabbix s'exécutent en tant que service Windows. Pour Zabbix agent 2, remplacez `agentd` par `agent2` dans les instructions ci-dessous.

Vous pouvez exécuter une seule instance de l'agent Zabbix ou plusieurs instances de l'agent sur un hôte Microsoft Windows. Une seule instance peut utiliser l'un des éléments suivants :

- le fichier de configuration par défaut, situé dans le même répertoire que le binaire de l'agent ;
- un fichier de configuration spécifié sur la ligne de commande.

En cas d'instances multiples, chaque instance de l'agent doit disposer de son propre fichier de configuration (l'une des instances peut utiliser le fichier de configuration par défaut).

Un exemple de fichier de configuration est disponible dans l'archive des sources Zabbix sous les noms suivants :

- `conf/zabbix_agentd.conf` pour l'agent Zabbix ;
- `conf/zabbix_agent2.conf` pour agent2 Zabbix.

Si vous souhaitez installer l'agent Zabbix/agent 2 pour Windows en tant que service à partir d'une archive sans spécifier explicitement le fichier de configuration, avant d'installer l'agent :

- `conf/zabbix_agentd.conf` doit être copié manuellement dans le répertoire où `zabbix_agentd.exe` sera installé ;
- `conf/zabbix_agent2.conf` et le répertoire `conf/zabbix_agent2.d` doivent être copiés manuellement dans le répertoire où `zabbix_agent2.exe` sera installé.

Consultez les options du fichier de configuration de l'agent Zabbix ou du fichier de configuration de l'agent 2 Zabbix pour plus d'informations sur la configuration de l'agent Zabbix pour Windows.

Paramètre Hostname

Pour effectuer des vérifications actives sur un hôte, l'agent Zabbix doit avoir le nom d'hôte défini. De plus, la valeur du nom d'hôte définie du côté de l'agent doit correspondre exactement au *Nom d'hôte* configuré pour l'hôte dans l'interface.

La valeur du nom d'hôte du côté de l'agent peut être définie par le paramètre `Hostname` ou `HostnameItem` dans le fichier de configuration de l'agent - sinon, les valeurs par défaut sont utilisées si aucun de ces paramètres n'est spécifié.

La valeur par défaut du paramètre `HostnameItem` est la valeur renvoyée par la clé d'agent `system.hostname`. Sous Windows, cette clé renvoie le résultat de la fonction `gethostname()`, qui interroge les fournisseurs d'espaces de noms afin de déterminer le nom d'hôte local. Si aucun fournisseur d'espace de noms ne répond, le nom NetBIOS est renvoyé.

La valeur par défaut de `Hostname` est la valeur renvoyée par le paramètre `HostnameItem`. Ainsi, si aucun de ces deux paramètres n'est spécifié, le nom d'hôte réel sera le nom NetBIOS de l'hôte ; l'agent Zabbix utilisera le nom d'hôte NetBIOS pour récupérer la liste des vérifications actives auprès du serveur Zabbix et lui envoyer les résultats.

La clé `system.hostname` prend en charge deux paramètres facultatifs : *type* et *transform*.

Type détermine le type de nom que l'élément doit renvoyer :

- *netbios* (par défaut) - renvoie le nom d'hôte NetBIOS, limité à 15 caractères et composé uniquement de lettres MAJUSCULES ;
- *host* - sensible à la casse, renvoie le nom d'hôte Windows complet et réel (sans domaine) ;
- *shorthost* - renvoie la partie du nom d'hôte située avant le premier point. La chaîne complète est renvoyée si le nom ne contient pas de point.
- *fqdn* - renvoie le nom de domaine complet (sans le point final).

Transform permet de spécifier une règle de transformation supplémentaire pour le nom d'hôte :

- *none* (par défaut) - conserve la casse d'origine ;
- *lower* - convertit le texte en minuscules.

Ainsi, pour simplifier la configuration du fichier `zabbix_agentd.conf` et la rendre uniforme, différentes approches peuvent être utilisées :

1. Laisser les paramètres `Hostname` ou `HostnameItem` non définis ; l'agent Zabbix utilisera alors le nom d'hôte NetBIOS comme nom d'hôte.
2. Laisser le paramètre `Hostname` non défini et définir `HostnameItem` comme suit :
   - `HostnameItem=system.hostname[host]` - pour que l'agent Zabbix utilise le nom d'hôte Windows complet et réel (sensible à la casse) comme nom d'hôte.
   - `HostnameItem=system.hostname[shorthost,lower]` - pour que l'agent Zabbix utilise uniquement la partie du nom d'hôte située avant le premier point, convertie en minuscules.
   - `HostnameItem=system.hostname[fqdn]` - pour que l'agent Zabbix utilise le nom de domaine complet comme nom d'hôte.

Le nom d'hôte est également utilisé comme partie du nom du service Windows, qui sert à installer, démarrer, arrêter et désinstaller le service Windows. Par exemple, si le fichier de configuration de l'agent Zabbix spécifie `Hostname=Windows_db_server`, l'agent sera installé en tant que service Windows `Zabbix Agent [Windows_db_server]`. Par conséquent, pour avoir un nom de service Windows différent pour chaque instance de l'agent Zabbix, chaque instance doit utiliser un nom d'hôte différent.

Installation de l'agent en tant que service Windows

Avant d'installer l'agent, copiez manuellement `conf/zabbix_agentd.conf` dans le répertoire où `zabbix_agentd.exe` sera installé.

Pour installer une instance unique de l'agent Zabbix avec le fichier de configuration par défaut :

```
zabbix_agentd.exe --install
```

> **Attention:**
> Sur un système 64 bits, une version 64 bits de l'agent Zabbix est requise pour que toutes les vérifications liées à l'exécution de processus 64 bits fonctionnent correctement.

Si vous souhaitez utiliser un fichier de configuration autre que celui par défaut, vous devez utiliser la commande suivante pour l'installation du service :

```
zabbix_agentd.exe --config <your_configuration_file> --install
```

Un chemin complet vers le fichier de configuration doit être spécifié.

Plusieurs instances de l'agent Zabbix peuvent être installées en tant que services de cette manière :

```
zabbix_agentd.exe --config <configuration_file_for_instance_1> --install --multiple-agents
zabbix_agentd.exe --config <configuration_file_for_instance_2> --install --multiple-agents
...
zabbix_agentd.exe --config <configuration_file_for_instance_N> --install --multiple-agents
```

Le service installé devrait maintenant être visible dans le Panneau de configuration.

Démarrage de l'agent

Pour démarrer le service de l'agent, vous pouvez utiliser le Panneau de configuration ou le faire depuis la ligne de commande.

Pour démarrer une instance unique de l'agent Zabbix avec le fichier de configuration par défaut :

```
zabbix_agentd.exe --start
```

Pour démarrer une instance unique de l'agent Zabbix avec un autre fichier de configuration :

```
zabbix_agentd.exe --config <your_configuration_file> --start
```

Pour démarrer l'une des plusieurs instances de l'agent Zabbix :

```
zabbix_agentd.exe --config <configuration_file_for_this_instance> --start --multiple-agents
```

Arrêt de l'agent

Pour arrêter le service de l'agent, vous pouvez utiliser le Panneau de configuration ou le faire depuis la ligne de commande.

Pour arrêter une instance unique de Zabbix agent démarrée avec le fichier de configuration par défaut :

```
zabbix_agentd.exe --stop
```

Pour arrêter une instance unique de Zabbix agent démarrée avec un autre fichier de configuration :

```
zabbix_agentd.exe --config <your_configuration_file> --stop
```

Pour arrêter l'une des plusieurs instances de Zabbix agent :

```
zabbix_agentd.exe --config <configuration_file_for_this_instance> --stop --multiple-agents
```

Désinstallation du service Windows de l'agent

Pour désinstaller une seule instance de Zabbix agent à l'aide du fichier de configuration par défaut :

```
zabbix_agentd.exe --uninstall
```

Pour désinstaller une seule instance de Zabbix agent à l'aide d'un fichier de configuration non par défaut :

```
zabbix_agentd.exe --config <your_configuration_file> --uninstall
```

Pour désinstaller plusieurs instances de Zabbix agent à partir des services Windows :

```
zabbix_agentd.exe --config <configuration_file_for_instance_1> --uninstall --multiple-agents
zabbix_agentd.exe --config <configuration_file_for_instance_2> --uninstall --multiple-agents
...
zabbix_agentd.exe --config <configuration_file_for_instance_N> --uninstall --multiple-agents
```

Limitations

L'agent Zabbix pour Windows ne prend pas en charge les configurations Windows non standard dans lesquelles les CPU sont répartis de manière non uniforme entre les nœuds NUMA. Si les CPU logiques sont répartis de manière non uniforme, les métriques de performance du CPU peuvent ne pas être disponibles pour certains CPU. Par exemple, s'il y a 72 CPU logiques avec 2 nœuds NUMA, chaque nœud doit avoir 36 CPU.

Agent Zabbix sur IBM AIX

Vue d'ensemble

Les binaires précompilés de l'agent Zabbix pour IBM AIX sont disponibles au téléchargement sous forme d'archive `.tar.gz`. Cette page décrit comment installer l'agent à partir de l'archive, l'exécuter manuellement ou avec le System Resource Controller (SRC), le démarrer au démarrage du système et le supprimer du SRC.

Installation de l'agent

La commande `tar` d'AIX ne prend pas en charge l'option `-z`. Décompressez donc l'archive avec `gunzip`. Utilisez l'option `-o` de `tar` afin que les fichiers extraits appartiennent à l'utilisateur qui les extrait, et non aux identifiants numériques de l'utilisateur et du groupe stockés dans l'archive. Par exemple, pour installer l'agent dans `/opt/zabbix` (à exécuter en tant que root ; `/opt/zabbix` ne doit pas encore exister) :

```
gunzip -c zabbix_agent-7.4.7-aix-7.2-powerpc-openssl.tar.gz | tar -xof -
mv zabbix_agent-7.4.7-aix-7.2-powerpc-openssl /opt/zabbix
```

Dans l'archive AIX, le binaire de l'agent se trouve dans `usr/sbin/` et le fichier de configuration dans `etc/zabbix/`. Avant de démarrer l'agent, ajustez `/opt/zabbix/etc/zabbix/zabbix_agentd.conf` si nécessaire (au minimum les paramètres `Server`, `ServerActive` et `Hostname`).

S'il est démarré en tant que root, l'agent bascule vers l'utilisateur `zabbix` (ou vers l'utilisateur indiqué par le paramètre `User`), sauf si `AllowRoot=1` est défini dans le fichier de configuration. Cet utilisateur doit exister sur le système ; sinon, l'agent ne démarre pas. Si l'utilisateur `zabbix` n'existe pas encore, créez-le avec un groupe du même nom :

```
mkgroup zabbix
mkuser pgrp=zabbix login=false rlogin=false zabbix
```

Les attributs `login=false` et `rlogin=false` empêchent l'utilisation du compte pour se connecter ; l'agent peut néanmoins s'exécuter avec cet utilisateur.

Démarrer l'agent manuellement

L'agent peut être démarré manuellement :

```
/opt/zabbix/usr/sbin/zabbix_agentd -c /opt/zabbix/etc/zabbix/zabbix_agentd.conf
```

Exécution de l'agent avec SRC

Enregistrement de l'agent

L'agent peut également être enregistré auprès du System Resource Controller (SRC). Lorsqu'il est démarré via le SRC, l'agent doit s'exécuter en mode premier plan (option `-f`). Dans le cas contraire, le SRC perd le suivi du processus de l'agent et ne peut pas l'arrêter. Par exemple (à exécuter en tant que root) :

```
mkssys \
  -s zabbix_agentd \
  -p /opt/zabbix/usr/sbin/zabbix_agentd \
  -a "-c /opt/zabbix/etc/zabbix/zabbix_agentd.conf -f" \
  -u zabbix \
  -G tcpip \
  -S -n 15 -f 9 \
  -R
```

Dans cet exemple :

- `-s` définit le nom du sous-système utilisé avec les commandes SRC ;
- `-p` définit le chemin complet vers le binaire de l'agent ;
- `-a` définit les arguments de ligne de commande de l'agent sous la forme d'une chaîne entre guillemets ; le SRC divise cette chaîne en arguments distincts lorsqu'il démarre l'agent (une seule option `-a` est acceptée) ;
- `-u` définit l'utilisateur sous lequel l'agent s'exécute ; le nom d'utilisateur est converti en identifiant utilisateur numérique lors de la création du sous-système. L'utilisateur doit donc déjà exister ;
- `-G` ajoute l'agent au groupe de sous-systèmes `tcpip`, afin que les commandes de groupe s'appliquent également à celui-ci (par exemple, `stopsrc -g tcpip` arrête également l'agent) ;
- `-S -n 15 -f 9` configure le SRC pour contrôler l'agent au moyen de signaux : `stopsrc` envoie `SIGTERM` (15) pour un arrêt normal et `SIGKILL` (9) pour un arrêt forcé (`stopsrc -f`) ;
- `-R` configure le SRC pour redémarrer l'agent s'il se termine anormalement (par exemple, si l'un de ses processus se termine de manière inattendue) ; l'agent n'est pas redémarré après l'exécution de `stopsrc`.

Comme l'identifiant utilisateur est enregistré dans la définition du sous-système, mettez à jour cette définition si l'utilisateur `zabbix` est recréé avec un autre identifiant utilisateur :

```
chssys -s zabbix_agentd -u zabbix
```

Démarrage et arrêt de l'agent

L'agent peut être démarré, interrogé et arrêté avec `startsrc`, `lssrc` et `stopsrc` :

```
startsrc -s zabbix_agentd
lssrc -s zabbix_agentd
stopsrc -s zabbix_agentd
```

Évitez l'arrêt forcé (`stopsrc -f`). Il tue uniquement le processus principal de l'agent ; les autres processus de l'agent continuent de s'exécuter et gardent le port d'écoute ouvert. L'agent ne peut donc pas être redémarré tant qu'ils n'ont pas été arrêtés manuellement (par exemple avec `kill -9`).

Démarrage de l'agent au démarrage du système

L'enregistrement de l'agent auprès de SRC ne le fait pas démarrer au démarrage du système. Pour démarrer automatiquement l'agent au démarrage, ajoutez une entrée à `/etc/inittab` :

```
mkitab "zabbix_agentd:2:once:/usr/bin/startsrc -s zabbix_agentd >/dev/console 2>&1"
```

Suppression de l'agent de SRC

Pour supprimer l'agent du démarrage du système et de SRC :

```
stopsrc -s zabbix_agentd
rmitab zabbix_agentd
rmssys -s zabbix_agentd
```

Agent 2

Vue d'ensemble Zabbix agent 2 est une nouvelle génération de agent Zabbix, écrit en Go (avec du code C réutilisé de Zabbix agent). Il est conçu pour :

- Réduire le nombre de connexions TCP ;
- Fournir une meilleure concurrence des vérifications ;

- Être facilement extensible grâce à des plugins qui fournissent des vérifications simples avec un code minimal et prennent en charge des vérifications complexes composées de scripts de longue durée et de collecte autonome de données avec rapport périodique ;
- Fonctionner comme un remplacement de Zabbix agent, en prenant en charge toutes les fonctionnalités précédentes.

Toutes les métriques dans Zabbix agent 2 sont collectées par des plugins.

Ces fonctionnalités sont spécifiques aux plugins :

- Intervalles planifiés et flexibles pour les vérifications passives et actives
- Gestion de la file des tâches en fonction de la planification et de la concurrence des tâches
- Délais d'attente au niveau du plugin
- Vérification de compatibilité de Zabbix agent 2 et de ses plugins au démarrage

Il existe deux types de plugins :

- Plugins intégrés
- Plugins chargeables

Vous pouvez également créer vos propres plugins.

Les plugins créés sur mesure n'ont pas besoin d'être intégrés à agent 2. Vous pouvez les ajouter en tant que plugins chargeables, ce qui simplifie ainsi la création de plugins pour de nouvelles métriques.

Les plugins chargeables peuvent être ajoutés à l'exécution sans recompiler Zabbix agent 2.

Voir aussi :

- Comparaison entre agent et agent 2
- Paramètres de configuration : Zabbix agent 2
- Paramètres de configuration : Zabbix agent 2 (Windows)
- Paramètres de configuration : Plugins
- Protocole des plugins de Zabbix agent 2

Vérifications passives et actives

Zabbix agent 2 prend en charge les vérifications passives et actives, tout comme Zabbix agent. De plus, les vérifications actives de Zabbix agent 2 prennent en charge les intervalles flexibles/planifiés et la concurrence des vérifications au sein d'un même serveur actif.

> **Note:**
> Par défaut, après un redémarrage, Zabbix agent 2 planifie la première collecte de données pour les vérifications actives à un moment conditionnellement aléatoire situé dans l'intervalle de mise à jour de l'élément, afin d'éviter les pics d'utilisation des ressources. Pour effectuer immédiatement après le redémarrage de l'agent les vérifications actives qui ne disposent pas d'un intervalle de mise à jour *Scheduling*, définissez le paramètre `ForceActiveChecksOnStart` (au niveau global) ou le paramètre du plugin `Plugins.<Plugin name>.System.ForceActiveChecksOnStart` (qui ne s'applique qu'aux vérifications d'un plugin spécifique) dans le fichier de configuration. Si le paramètre au niveau du plugin est défini, il remplace le paramètre global.

Concurrence des vérifications

Les vérifications provenant de différents plugins peuvent être exécutées simultanément. Le nombre de vérifications simultanées au sein d'un même plugin est limité par le paramètre de capacité du plugin. Chaque plugin peut avoir un paramètre de capacité codé en dur (1000 étant la valeur par défaut) qui peut être réduit à l'aide du paramètre `Plugins.<PluginName>.System.Capacity=N` dans la configuration `Plugins` parameter.

Plates-formes prises en charge Pour les plates-formes prises en charge, consultez la page Requirements.

Agent 2 sur les systèmes de type Unix Zabbix agent 2 sur les systèmes de type Unix s'exécute sur l'hôte surveillé.

Installation

Zabbix agent 2 peut être installé sur des systèmes basés sur Linux à l'aide de l'une des méthodes suivantes :

- Paquets Zabbix - sélectionnez le composant Agent 2 (après avoir sélectionné votre version de Zabbix, la distribution de l'OS et la version de l'OS) et suivez les instructions ;
- Sources Zabbix - téléchargez les fichiers source et compilez l'agent en configurant avec l'option `--enable-agent2`.

Note:
Les capacités de surveillance de Zabbix agent 2 peuvent être étendues avec des plugins chargeables, disponibles séparément. Pour plus de détails, voir Plugins chargeables.

Si installé en tant que paquet

Zabbix agent 2 s'exécute en tant que processus au premier plan et s'appuie sur un gestionnaire de services externe (par exemple, systemd) pour l'exécution en arrière-plan. Zabbix agent 2 ne dispose pas d'une prise en charge intégrée de la mise en arrière-plan en tant que démon sous Linux.

L'agent peut être démarré en exécutant :

```
systemctl start zabbix-agent2
```

Pour arrêter, redémarrer ou vérifier l'état de Zabbix agent 2, utilisez les commandes suivantes :

```
systemctl stop zabbix-agent2
systemctl restart zabbix-agent2
systemctl status zabbix-agent2
```

Démarrer manuellement

Vous pouvez démarrer l'agent Zabbix en localisant le binaire `zabbix_agent2` et en l'exécutant directement ; par exemple :

```
zabbix_agent2
```

Agent 2 sur les systèmes Windows Zabbix agent 2 s'exécute en tant que processus autonome ; toutefois, il peut également être exécuté en tant que service Windows.

Installation

Zabbix agent 2 peut être installé sur Windows en utilisant l'une des méthodes suivantes :

- Binaires Zabbix agent précompilés - téléchargez le package d'installation MSI de l'agent et suivez les instructions de la page Installation de l'agent Windows à partir de MSI ;
- Sources Zabbix - téléchargez les fichiers sources et suivez les instructions de la page Compilation de Zabbix agent 2 sur Windows.

Note:
Les capacités de supervision de Zabbix agent 2 peuvent être étendues à l'aide de plugins chargeables, disponibles séparément. Pour plus de détails, consultez Plugins chargeables.

Consultez la page Zabbix agent sur Microsoft Windows pour plus de détails sur l'installation de Zabbix agent 2 (à partir d'une archive ZIP) en tant que service Windows.

Options Les paramètres de ligne de commande suivants peuvent être utilisés avec Zabbix agent 2 :

Paramètre	Description
Agent UNIX et Windows	
`-c --config <config-file>`	Chemin vers le fichier de configuration. Vous pouvez utiliser cette option pour spécifier un fichier de configuration différent de celui utilisé par défaut.
`-f --foreground`	Exécuter Zabbix agent au premier plan (valeur par défaut : true).
`-p --print`	Afficher les éléments connus, puis quitter. Notez que pour renvoyer également les résultats des paramètres utilisateur, vous devez spécifier le fichier de configuration (s'il ne se trouve pas à l'emplacement par défaut).
`-t --test <item key>`	Tester l'élément spécifié, puis quitter. Notez que pour renvoyer également les résultats des paramètres utilisateur, vous devez spécifier le fichier de configuration (s'il ne se trouve pas à l'emplacement par défaut).
`-T --test-config`	Valider le fichier de configuration, puis quitter.
`-h --help`	Afficher les informations d'aide, puis quitter.
`-v --verbose`	Afficher les informations de débogage. Utilisez cette option avec les options `-p` et `-t`.
`-V --version`	Afficher la version de l'agent et les informations de licence.

Paramètre	Description
-R --runtime-control <option>	Effectuer des fonctions administratives. Consultez la section contrôle à l'exécution.
Agent Windows uniquement	
-m	Utiliser plusieurs instances de l'agent (avec les options -i, -d, -s, -x).
--multiple-agents	Pour distinguer les noms de service des instances, chaque nom de service inclura la valeur Hostname du fichier de configuration spécifié.
-S --startup-type <value>	Définir le type de démarrage du service Zabbix agent pour Windows. Valeurs autorisées : automatic - *(par défaut)* démarrer automatiquement le service au démarrage de Windows ; delayed - retarder le démarrage du service jusqu'à ce que les services démarrés automatiquement aient terminé leur démarrage ; manual - démarrer le service manuellement (par un utilisateur ou une application) ; disabled - désactiver le service afin qu'il ne puisse pas être démarré par un utilisateur ou une application. Vous pouvez utiliser cette option avec l'option -i, ou séparément pour modifier le type de démarrage d'un service déjà installé.
-i --install	Installer Zabbix agent pour Windows en tant que service.
-d --uninstall	Désinstaller le service Zabbix agent pour Windows.
-s --start	Démarrer le service Zabbix agent pour Windows.
-x --stop	Arrêter le service Zabbix agent pour Windows.

Exemples spécifiques d'utilisation des paramètres de ligne de commande :

- Afficher tous les éléments intégrés de l'agent avec leurs valeurs.
- Tester un paramètre utilisateur avec la clé mysql.ping définie dans le fichier de configuration spécifié.
- Installer un service Zabbix Agent pour Windows en utilisant le chemin par défaut vers le fichier de configuration.
- Modifier le type de démarrage d'un service Zabbix Agent pour Windows installé en utilisant le fichier de configuration situé dans le même dossier que l'exécutable de l'agent.

```
zabbix_agent2 --print
zabbix_agent2 -t "mysql.ping" -c /etc/zabbix/zabbix_agent2.conf
zabbix_agent2.exe -i
zabbix_agent2.exe -c zabbix_agent2.conf -S delayed
```

Contrôle de l'exécution

Le contrôle de l'exécution fournit plusieurs options de contrôle à distance.

Option	Description
log_level_increase	Augmenter le niveau de journalisation.
log_level_decrease	Diminuer le niveau de journalisation.
metrics	Répertorier les métriques disponibles.
periodic_prof_enable	Activer le profilage périodique. Toutes les métriques collectent les données au moment de l'exécution et les écrivent, à l'exception des données du processeur. Les données périodiques du processeur sont collectées au fil du temps dans un fichier temporaire masqué. Lorsque l'intervalle se termine, le fichier est publié en tant que profil du processeur et le nombre configuré de fichiers terminés est conservé. Le fichier temporaire n'est pas comptabilisé dans ProfilerMaxFilesPerProfile.
periodic_prof_disable	Désactiver le profilage périodique.
periodic_prof_execute	Écrire les profils après un échantillonnage de 5 secondes. Lorsque le profilage périodique est activé, cette commande termine l'intervalle actuel du processeur, écrit un profil distinct du processeur de 5 secondes et démarre un nouveau fichier périodique du processeur. Les profils explicitement demandés sont soumis à une rotation conformément à ProfilerMaxFilesPerProfile lorsque la commande se termine. Pendant la collecte des profils par cette commande, toutes les commandes d'exécution du profileur sont rejetées avec le message « profiler is busy ».
periodic_prof_set_interval	Définir l'intervalle de profilage.
version	Afficher la version de l'agent.

Option	Description
userparameter_reload	Recharger les valeurs des options UserParameter et Include à partir du fichier de configuration actuel.
help	Afficher les informations d'aide sur le contrôle de l'exécution.

Exemples :

- Augmenter le niveau de journalisation de l'agent 2.
- Afficher les options de contrôle de l'exécution.

```
zabbix_agent2 -R log_level_increase
zabbix_agent2 -R help
```

Fichier de configuration Le chemin par défaut du fichier de configuration dépend du système d'exploitation et de la méthode d'installation de Zabbix :

- Sous UNIX, la valeur par défaut est :
  - Pour une installation à partir des sources : /usr/local/etc/zabbix_agent2.conf, ou selon les variables définies lors de la compilation --sysconfdir ou --prefix.
  - Pour une installation à partir de paquets : /etc/zabbix/zabbix_agent2.conf.
- Sous Windows, la valeur par défaut est C:\Program Files\Zabbix Agent 2\zabbix_agent2.conf.

Si une seule instance de l'agent s'exécute sur un hôte, elle peut utiliser le fichier de configuration par défaut ou un fichier de configuration spécifié sur la ligne de commande. En cas d'instances multiples, chaque instance de l'agent doit disposer de son propre fichier de configuration (l'une des instances peut utiliser le fichier de configuration par défaut).

Les paramètres de configuration de Zabbix agent 2 correspondent pour la plupart à ceux de Zabbix agent, à quelques exceptions près :

Paramètres		Description
Paramètres de configuration spécifiques à Zabbix agent 2		
	ControlSocket	Chemin du socket de contrôle d'exécution. Zabbix agent 2 utilise un socket de contrôle pour les commandes d'exécution.
	EnablePersistentBuffer, PersistentBufferFile, PersistentBufferPeriod	Ces paramètres servent à configurer le stockage persistant de Zabbix agent 2 pour les éléments actifs.
	ForceActiveChecksOnStart	Détermine si l'agent doit effectuer les vérifications actives immédiatement après un redémarrage ou les répartir uniformément dans le temps.
	Plugins	Les plugins peuvent avoir leurs propres paramètres, au format Plugins.<Plugin name>.<Parameter>=<value>. Un paramètre de plugin courant est System.Capacity, qui définit la limite du nombre de vérifications pouvant être exécutées simultanément.
	StatusPort	Port sur lequel Zabbix agent 2 écoute les requêtes d'état HTTP et l'affichage d'une liste des plugins configurés ainsi que de certains paramètres internes.

Paramètres		Description
Paramètres de configuration spécifiques à Zabbix agent non pris en charge par Zabbix agent 2		
	`AllowRoot`, `User`	Non pris en charge par Zabbix agent 2, car celui-ci ne prend pas en charge la démonisation.
	`EnableRemoteCommands`	Obsolète ; les paramètres `AllowKey`, `AllowKeyRegexp`, `DenyKey` et `DenyKeyRegexp` sont utilisés à la place dans les deux agents.
	`ListenBacklog`	Utilisé dans Zabbix agent pour définir le nombre maximal de connexions en attente dans la file TCP. Non pris en charge par Zabbix agent 2.
	`LoadModule`, `LoadModulePath`	Les modules chargeables ne sont pas pris en charge par Zabbix agent 2.
	`MaxLinesPerSecond`, `LogRemoteCommands`	Dans Zabbix agent 2, `Plugins.Log.MaxLinesPerSecond` et `Plugins.SystemRun.LogRemoteCommands` sont utilisés à la place.
	`StartAgents`	Ce paramètre est utilisé dans Zabbix agent pour augmenter la simultanéité des vérifications passives ou les désactiver. Dans Zabbix agent 2, la simultanéité est configurée au niveau du plugin et peut être limitée par un paramètre de capacité. Les vérifications passives sont désactivées dans Zabbix Agent 2 si le paramètre `Server` n'est pas spécifié.

Pour plus de détails, consultez les options du fichier de configuration de Zabbix agent2.

Codes de sortie Zabbix agent 2 peut également être compilé avec des versions plus anciennes d'OpenSSL (1.0.1, 1.0.2).

Dans ce cas, Zabbix fournit des mutex pour le verrouillage dans OpenSSL. Si le verrouillage ou le déverrouillage d'un mutex échoue, un message d'erreur est affiché sur le flux d'erreur standard (STDERR) et Agent 2 se termine avec le code de retour 2 ou 3, respectivement.

Comparaison entre agent et agent 2

Cette section décrit les différences entre Zabbix agent et Zabbix agent 2.

Parameter	Zabbix agent	Zabbix agent 2
Langage de programmation	C	Go avec certaines parties en C
Démonisation	oui	uniquement via systemd (oui sous Windows)
Extensions prises en charge	Modules chargeables personnalisés loadable modules en C.	Plugins personnalisés en Go.
Exigences		

Parameter	Zabbix agent	Zabbix agent 2
Plateformes prises en charge	Linux, IBM AIX, FreeBSD, NetBSD, OpenBSD, HP-UX, Mac OS X, Solaris: 9, 10, 11, Windows: toutes les versions bureau et serveur depuis XP	Linux, Windows: toutes les versions bureau et serveur, sur lesquelles une version Go prise en charge peut être installée.
Bibliothèques cryptographiques prises en charge	GnuTLS 3.1.18 et versions plus récentes. OpenSSL 1.0.1, 1.0.2, 1.1.0, 1.1.1, 3.0.x. LibreSSL - testé avec les versions 2.7.4, 2.8.2 (certaines limitations s'appliquent, voir la page Encryption pour plus de détails).	Linux: OpenSSL 1.0.1 et versions ultérieures. MS Windows: OpenSSL 1.1.1 ou versions ultérieures. La bibliothèque OpenSSL doit avoir la prise en charge PSK activée. LibreSSL n'est pas pris en charge.
Processus de surveillance		
Processus	Un processus de vérification active distinct pour chaque enregistrement serveur/proxy.	Un seul processus avec des threads créés automatiquement. Le nombre maximal de threads est déterminé par la variable d'environnement `GOMAXPROCS`.
Mesures	**UNIX**: voir la liste des éléments pris en charge. **Windows**: voir la liste des éléments supplémentaires spécifiques à Windows.	**UNIX**: Toutes les mesures prises en charge par Zabbix agent. De plus, agent 2 fournit une solution de surveillance native Zabbix pour: Docker, Memcached, MySQL, PostgreSQL, Redis, systemd et d'autres cibles de surveillance - voir la liste complète des éléments spécifiques à agent 2. **Windows**: Toutes les mesures prises en charge par Zabbix agent, ainsi que les vérifications `net.tcp.service*` pour HTTPS, LDAP. De plus, agent 2 fournit une solution de surveillance native Zabbix pour: PostgreSQL, Redis.
Concurrence	Les vérifications actives pour un seul serveur sont exécutées séquentiellement.	Les vérifications provenant de différents plugins ou plusieurs vérifications au sein d'un même plugin peuvent être exécutées simultanément.
Trappes tierces	non	oui
Fonctionnalités supplémentaires		
Stockage persistant	non	oui
Fichiers persistants pour les mesures `log*[]`	oui (uniquement sur Unix)	non
Téléversement des données de journal	Peut être effectué pendant la collecte des journaux afin de libérer le tampon.	La collecte des journaux s'arrête lorsque le tampon est plein, par conséquent le paramètre `BufferSize` doit être au moins égal à `MaxLinesPerSecond` x 2.
Change d'utilisateur à l'exécution	oui (systèmes de type Unix uniquement)	non (contrôlé par systemd)
Suites de chiffrement configurables par l'utilisateur	oui	non

Voir aussi:

- *Description des processus Zabbix*: Zabbix agent, Zabbix agent 2
- *Paramètres de configuration*: Zabbix agent UNIX / Windows, Zabbix agent 2 UNIX / Windows

Plugins intégrés

Ces plugins sont intégrés à Zabbix agent 2 et sont disponibles immédiatement. Cliquez sur le nom du plugin pour accéder au dépôt du plugin avec des informations supplémentaires.

Nom du plugin	Description	Clés d'élément prises en charge	Commentaires
Agent	Métriques de l'agent Zabbix utilisé.	agent.hostname, agent.ping, agent.version	Les clés prises en charge ont les mêmes paramètres que les clés de Zabbix agent.
CPU	Surveillance du CPU système (nombre de CPU/cœurs CPU, CPU découverts, pourcentage d'utilisation).	system.cpu.discovery, system.cpu.num, system.cpu.util	Les clés prises en charge ont les mêmes paramètres que les clés de Zabbix agent.
Docker	Surveillance des conteneurs Docker.	docker.container_info, docker.container_stats, docker.containers, docker.containers.discovery, docker.data_usage, docker.images, docker.images.discovery, docker.info, docker.ping	Voir aussi : Paramètres de configuration
File	Collecte des métriques de fichiers.	vfs.file.cksum, vfs.file.contents, vfs.file.exists, vfs.file.md5sum, vfs.file.regexp, vfs.file.regmatch, vfs.file.size, vfs.file.time	Les clés prises en charge ont les mêmes paramètres que les clés de Zabbix agent.
Kernel	Surveillance du noyau.	kernel.maxfiles, kernel.maxproc	Les clés prises en charge ont les mêmes paramètres que les clés de Zabbix agent.
Log	Surveillance des fichiers journaux.	log, log.count, logrt, logrt.count	Les clés prises en charge ont les mêmes paramètres que les clés de Zabbix agent. Voir aussi : Paramètres de configuration du plugin (Unix/Windows)
Memcached	Surveillance du serveur Memcached.	memcached.ping, memcached.stats	
Modbus	Lit les données Modbus.	modbus.get	Les clés prises en charge ont les mêmes paramètres que les clés de Zabbix agent.
MQTT	Reçoit les valeurs publiées des topics MQTT.	mqtt.get	Pour configurer une connexion chiffrée au broker MQTT, spécifiez les paramètres TLS dans le fichier de configuration de l'agent sous forme de paramètres de session nommée ou de valeur par défaut. Actuellement, les paramètres TLS ne peuvent pas être transmis comme paramètres de clé d'élément.
MySQL	Surveillance de MySQL et de ses dérivés.	mysql.custom.query, mysql.db.discovery, mysql.db.size, mysql.get_status_variables, mysql.ping, mysql.replication.discovery, mysql.replication.get_slave_status, mysql.version	Pour configurer une connexion chiffrée à la base de données, spécifiez les paramètres TLS dans le fichier de configuration de l'agent sous forme de paramètres de session nommée ou de valeur par défaut. Actuellement, les paramètres TLS ne peuvent pas être transmis comme paramètres de clé d'élément.
NetIf	Surveillance des interfaces réseau.	net.if.collisions, net.if.discovery, net.if.get, net.if.in, net.if.out, net.if.total	Les clés prises en charge ont les mêmes paramètres que les clés de Zabbix agent.

Nom du plugin	Description	Clés d'élément prises en charge	Commentaires
Oracle	Surveillance d'Oracle Database.	oracle.diskgroups.stats, oracle.diskgroups.discovery, oracle.archive.info, oracle.archive.discovery, oracle.cdb.info, oracle.custom.query, oracle.datafiles.stats, oracle.db.discovery, oracle.fra.stats, oracle.instance.info, oracle.pdb.info, oracle.pdb.discovery, oracle.pga.stats, oracle.ping, oracle.proc.stats, oracle.redolog.info, oracle.sga.stats, oracle.sessions.stats, oracle.sys.metrics, oracle.sys.params, oracle.ts.stats, oracle.ts.discovery, oracle.user.info, oracle.version	Installez Oracle Instant Client avant d'utiliser le plugin.
Proc	Pourcentage d'utilisation CPU des processus.	proc.cpu.util	La clé prise en charge a les mêmes paramètres que la clé de Zabbix agent.
Redis	Surveillance du serveur Redis.	redis.config, redis.info, redis.ping, redis.slowlog.count	Pour configurer une connexion chiffrée à Redis, spécifiez les paramètres TLS dans le fichier de configuration de l'agent sous forme de paramètres de session nommée ou de valeur par défaut. Les paramètres TLS ne peuvent pas être transmis comme paramètres de clé d'élément. Notez qu'une configuration TLS incorrecte ou autrement invalide peut empêcher le démarrage de Zabbix agent 2 ; vérifiez donc les fichiers de certificats, les autorisations et les chemins avant d'activer TLS.

Smart	Surveillance S.M.A.R.T.	smart.attribute.discovery, smart.disk.discovery, smart.disk.get	La version minimale requise de smartctl est 7.1.
			Des droits sudo/root sur smartctl sont requis pour l'utilisateur exécutant Zabbix agent 2.
			Le plugin utilise uniquement les commandes suivantes :
			`/usr/sbin/smartctl -a *`
			`/usr/sbin/smartctl --scan *`
			`/usr/sbin/smartctl -j -V`
			
			Les clés prises en charge peuvent être utilisées avec Zabbix agent 2 uniquement sous Linux/Windows, à la fois comme vérification passive et active.
			Voir aussi :
			Paramètres de configuration

SW	Liste des paquets installés.	system.sw.packages, system.sw.packages.get	Les clés prises en charge ont les mêmes paramètres que la clé de Zabbix agent.

Nom du plugin	Description	Clés d'élément prises en charge	Commentaires
Swap	Taille de l'espace d'échange en octets/pourcentage.	system.swap.size	La clé prise en charge a les mêmes paramètres que la clé de Zabbix agent.
SystemRun	Exécute la commande spécifiée.	system.run	La clé prise en charge a les mêmes paramètres que la clé de Zabbix agent. Voir aussi : Paramètres de configuration du plugin (Unix/Windows)
Systemd	Surveillance des services systemd.	systemd.unit.discovery, systemd.unit.get, systemd.unit.info	
TCP	Vérification de la disponibilité des connexions TCP.	net.tcp.port	La clé prise en charge a les mêmes paramètres que la clé de Zabbix agent.
UDP	Surveillance de la disponibilité et des performances des services UDP.	net.udp.service, net.udp.service.perf	Les clés prises en charge ont les mêmes paramètres que les clés de Zabbix agent.
Uname	Récupération d'informations sur le système.	system.hostname, system.sw.arch, system.uname	Les clés prises en charge ont les mêmes paramètres que les clés de Zabbix agent.
Uptime	Collecte des métriques de disponibilité du système.	system.uptime	La clé prise en charge a les mêmes paramètres que la clé de Zabbix agent.
VFSDev	Collecte des métriques VFS.	vfs.dev.discovery, vfs.dev.get, vfs.dev.read, vfs.dev.write	Les clés prises en charge ont les mêmes paramètres que les clés de Zabbix agent.
WebCertificate	Surveillance des certificats de sites web TLS/SSL.	web.certificate.get	
WebPage	Surveillance des pages web.	web.page.get, web.page.perf, web.page.regexp	Les clés prises en charge ont les mêmes paramètres que les clés de Zabbix agent.
ZabbixAsync	Collecte asynchrone des métriques.	net.tcp.listen, net.udp.listen, sensor, system.boottime, system.cpu.intr, system.cpu.load, system.cpu.switches, system.hw.cpu, system.hw.macaddr, system.localtime, system.sw.os, system.swap.in, system.swap.out, vfs.fs.discovery	Les clés prises en charge ont les mêmes paramètres que les clés de Zabbix agent.
ZabbixStats	Métriques internes du serveur/proxy Zabbix ou nombre d'éléments en retard dans une file d'attente.	zabbix.stats	Les clés prises en charge ont les mêmes paramètres que les clés de Zabbix agent.

Nom du plugin	Description	Clés d'élément prises en charge	Commentaires
ZabbixSync	Collecte synchrone des métriques.	net.dns, net.dns.record, net.tcp.service, net.tcp.service.perf, proc.mem, proc.num, system.hw.chassis, system.hw.devices, system.sw.packages, system.users.num, vfs.dir.count, vfs.dir.size, vfs.fs.get, vfs.fs.inode, vfs.fs.size, vm.memory.size.	Les clés prises en charge ont les mêmes paramètres que les clés de Zabbix agent.

Plugins chargeables

Les plugins chargeables ne sont pas disponibles par défaut pour Zabbix agent 2 et doivent être installés séparément :

- Sous Linux, vous pouvez utiliser des packages (par exemple, installer Ember+ sur Ubuntu avec `apt install zabbix-agent2-plugin-ember-plus`) ou compiler les plugins
- Sous Windows, vous pouvez installer depuis un fichier MSI ou compiler les plugins

Attention:
Avant d'installer un plugin, veuillez consulter son fichier README. Il peut contenir des exigences et des instructions d'installation spécifiques.

Cliquez sur le nom du plugin pour accéder à son dépôt, qui contient le fichier README du plugin avec des informations supplémentaires.

Nom du plugin	Description	Clés d'élément prises en charge	Commentaires
Ceph	Surveillance de Ceph.	ceph.df.details, ceph.osd.stats, ceph.osd.discovery, ceph.osd.dump, ceph.ping, ceph.pool.discovery, ceph.status	Voir également les paramètres de configuration du plugin Ceph.
Ember+	Surveillance d'Ember+.	ember.get	Voir également les paramètres de configuration du plugin Ember+.

Nom du plugin	Description	Clés d'élément prises en charge	Commentaires
MongoDB	Surveillance des serveurs et clusters MongoDB (base de données distribuée orientée documents).	mongodb.cfg.discovery, mongodb.collection.stats, mongodb.collections.discovery, mongodb.collections.usage, mongodb.connpool.stats, mongodb.db.stats, mongodb.db.discovery, mongodb.jumbo_chunks.count, mongodb.oplog.stats, mongodb.ping, mongodb.rs.config, mongodb.rs.status, mongodb.server.status, mongodb.sh.discovery, mongodb.version	Pour configurer des connexions chiffrées à la base de données, spécifiez les paramètres TLS dans le fichier de configuration de l'agent en tant que paramètres d'une session nommée. Actuellement, les paramètres TLS ne peuvent pas être transmis en tant que paramètres de clé d'élément. Voir également les paramètres de configuration du plugin MongoDB.
MSSQL	Surveillance d'une base de données MSSQL.	mssql.availability.group.get, mssql.custom.query, mssql.db.get, mssql.job.status.get, mssql.last.backup.get, mssql.local.db.get, mssql.mirroring.get, mssql.nonlocal.db.get, mssql.perfcounter.get, mssql.ping, mssql.quorum.get, mssql.quorum.member.get, mssql.replica.get, mssql.version	Pour configurer une connexion chiffrée à la base de données, spécifiez les paramètres TLS dans le fichier de configuration de l'agent en tant que paramètres d'une session nommée ou par défaut. Actuellement, les paramètres TLS ne peuvent pas être transmis en tant que paramètres de clé d'élément. Voir également les paramètres de configuration du plugin MSSQL.
NVIDIA GPU	Surveillance des GPU NVIDIA.	nvml.device.count, nvml.device.decoder.utilization, nvml.device.ecc.mode, nvml.device.encoder.stats.get, nvml.device.encoder.utilization, nvml.device.energy.consumption, nvml.device.errors.memory, nvml.device.errors.register, nvml.device.fan.speed.avg, nvml.device.get, nvml.device.graphics.frequency, nvml.device.memory.bar1.get, nvml.device.memory.fb.get, nvml.device.memory.frequency, nvml.device.pci.utilization, nvml.device.performance.state, nvml.device.power.limit, nvml.device.power.usage, nvml.device.serial, nvml.device.sm.frequency, nvml.device.temperature, nvml.device.utilization, nvml.device.video.frequency, nvml.system.driver.version, nvml.version	Voir également les paramètres de configuration du plugin NVIDIA GPU.

Nom du plugin	Description	Clés d'élément prises en charge	Commentaires
PostgreSQL	Surveillance de PostgreSQL et de ses forks.	pgsql.autovacuum.count, pgsql.archive, pgsql.bgwriter, pgsql.cache.hit, pgsql.connections, pgsql.custom.query, pgsql.dbstat, pgsql.dbstat.sum, pgsql.db.age, pgsql.db.bloating_tables, pgsql.db.discovery, pgsql.db.size, pgsql.locks, pgsql.oldest.xid, pgsql.ping, pgsql.queries, pgsql.replication.count, pgsql.replication.process, pgsql.replication.process.discovery, pgsql.replication.recovery_role, pgsql.replication.status, pgsql.replication_lag.b, pgsql.replication_lag.sec, pgsql.uptime, pgsql.version, pgsql.wal.stat	Pour configurer des connexions chiffrées à la base de données, spécifiez les paramètres TLS dans le fichier de configuration de l'agent en tant que paramètres d'une session nommée ou par défaut. Actuellement, les paramètres TLS ne peuvent pas être transmis en tant que paramètres de clé d'élément. Voir également les paramètres de configuration du plugin PostgreSQL.

Note:
Plugins chargeables, lorsqu'ils sont lancés avec :
 - `-V --version` - afficher la version du plugin et les informations de licence ;
 - `-h --help` - afficher les informations d'aide.
 - `-t, --test <item key>` - lancer le plugin à des fins de test (la configuration du plugin est ignorée).

Paramètres de configuration: Zabbix agent 2

Vue d'ensemble

Les paramètres de configuration permettent de personnaliser le comportement de Zabbix agent 2.

Zabbix agent 2 est une nouvelle génération d'agent Zabbix et peut être utilisé à la place de Zabbix agent.

Indiquez les valeurs des paramètres dans le fichier de configuration de l'agent 2 de Zabbix (`zabbix_agent2.conf`). Tous les paramètres sont **facultatifs**, sauf indication explicite contraire.

Cette page est structurée comme suit :

- Vue d'ensemble des paramètres (pour afficher tous les détails d'un paramètre, cliquez sur son nom)
- Détails des paramètres

Paramètre	Description
Alias	Définir un alias pour une clé d'élément.
AllowKey	Autoriser l'exécution des clés d'élément correspondant à un modèle.
AllowKeyRegexp	Autoriser l'exécution des clés d'élément correspondant à un modèle basé sur une expression régulière.
BufferSend	Définir la fréquence d'envoi des valeurs de la mémoire tampon au serveur/proxy Zabbix.
BufferSize	Définir le nombre maximal de valeurs dans la mémoire tampon.
ControlSocket	Indiquer le socket de contrôle utilisé pour envoyer des commandes d'exécution avec l'option « -R ».
DebugLevel	Définir le niveau de débogage.
DenyKey	Interdire l'exécution des clés d'élément correspondant à un modèle.

Paramètre	Description
DenyKeyRegexp	Interdire l'exécution des clés d'élément correspondant à un modèle basé sur une expression régulière.
EnablePersistentBuffer	Activer l'utilisation d'un stockage local persistant pour les éléments actifs.
EnableProfiler	Activer le profilage Go périodique au démarrage de l'agent.
ForceActiveChecksOnStart	Effectuer immédiatement les vérifications actives après le redémarrage pour la première configuration reçue.
HeartbeatFrequency	Définir la fréquence des messages de pulsation, en secondes.
HostInterface	Indiquer l'interface de l'hôte.
HostInterfaceItem	Indiquer un élément permettant d'obtenir l'interface de l'hôte.
HostMetadata	Indiquer les métadonnées de l'hôte.
HostMetadataItem	Indiquer un élément de Zabbix agent permettant d'obtenir les métadonnées de l'hôte.
Hostname	Indiquer le nom d'hôte.
HostnameItem	Indiquer un élément de Zabbix agent permettant d'obtenir le nom d'hôte.
Include	Indiquer un répertoire ou des fichiers individuels à inclure dans le fichier de configuration.
ListenIP	Indiquer l'adresse IP d'écoute de l'agent pour les connexions provenant du serveur/proxy.
ListenPort	Indiquer le port d'écoute de l'agent.
LogFile	Indiquer le fichier journal de l'agent.
LogFileSize	Définir la taille maximale du fichier journal.
LogType	Indiquer le type de sortie des journaux.
PersistentBufferFile	Indiquer le fichier utilisé pour conserver la base de données SQLite.
PersistentBufferPeriod	Définir la période de conservation des données lorsqu'il n'existe aucune connexion au serveur ou au proxy Zabbix.
PidFile	Indiquer le fichier PID.
Plugins.<PluginName>.SystemCapacity	Définir la limite de vérifications par plugin pouvant être exécutées simultanément.
Plugins.Log.MaxLinesPerSecond	Définir le nombre maximal de nouvelles lignes par seconde à envoyer au serveur/proxy Zabbix lors du traitement des vérifications actives `log` ou `logrt`.
Plugins.SystemRun.LogRemoteCommands	Activer la journalisation des commandes shell exécutées en tant qu'avertissements.
PluginSocket	Indiquer le chemin du socket UNIX utilisé pour les communications avec les plugins chargeables.
PluginTimeout	Définir le délai d'expiration des connexions aux plugins chargeables.
ProfilerDir	Répertoire dans lequel les données de profilage seront écrites.
ProfilerMaxFilesPerProfile	Nombre maximal de fichiers à conserver pour chaque type de profil.
ProfilerInterval	Intervalle d'écriture des données de profilage.
RefreshActiveChecks	Indiquer la fréquence d'actualisation de la liste des vérifications actives.
Server	Indiquer les adresses IP ou les noms DNS des serveurs/proxys Zabbix.
ServerActive	Indiquer l'adresse du serveur/proxy Zabbix ou la configuration du cluster depuis laquelle obtenir les vérifications actives.
SourceIP	Indiquer l'adresse IP source.
StatusPort	Indiquer le port d'écoute des requêtes d'état HTTP (`http://localhost:<port>/status`).
Timeout	Indiquer le délai d'attente (en secondes) pour établir la connexion et échanger des données avec le serveur ou le proxy Zabbix.
TLSAccept	Indiquer le niveau de chiffrement requis pour les connexions entrantes.
TLSCAFile	Indiquer le chemin complet du fichier contenant les certificats de l'autorité de certification (CA) de niveau supérieur utilisés pour vérifier le certificat du pair, dans le cadre des communications chiffrées entre les composants Zabbix.
TLSCertFile	Indiquer le chemin complet du fichier contenant le certificat de l'agent ou la chaîne de certificats, utilisé pour les communications chiffrées entre les composants Zabbix.
TLSCipherAll	Indiquer la chaîne de chiffrement OpenSSL (TLS 1.2). Remplacer les critères de sélection par défaut de la suite de chiffrement pour le chiffrement basé sur un certificat et sur une clé prépartagée (PSK).
TLSCipherAll13	Indiquer la chaîne de chiffrement OpenSSL (TLS 1.3) pour OpenSSL 1.1.1 ou version ultérieure. Remplacer les critères de sélection par défaut de la suite de chiffrement pour le chiffrement basé sur un certificat et sur une clé prépartagée (PSK).
TLSCipherCert	Indiquer la chaîne de chiffrement OpenSSL (TLS 1.2). Remplacer les critères de sélection par défaut de la suite de chiffrement pour le chiffrement basé sur un certificat.
TLSCipherCert13	Indiquer la chaîne de chiffrement OpenSSL (TLS 1.3) pour OpenSSL 1.1.1 ou version ultérieure. Remplacer les critères de sélection par défaut de la suite de chiffrement pour le chiffrement basé sur un certificat.
TLSCipherPSK	Indiquer la chaîne de chiffrement OpenSSL (TLS 1.2). Remplacer les critères de sélection par défaut de la suite de chiffrement pour le chiffrement basé sur une clé prépartagée (PSK).
TLSCipherPSK13	Indiquer la chaîne de chiffrement OpenSSL (TLS 1.3) pour OpenSSL 1.1.1 ou version ultérieure. Remplacer les critères de sélection par défaut de la suite de chiffrement pour le chiffrement basé sur une clé prépartagée (PSK).

Paramètre	Description
TLSConnect	Indiquer le niveau de chiffrement requis pour les connexions au serveur ou au proxy Zabbix.
TLSCRLFile	Indiquer le chemin complet du fichier contenant les certificats révoqués. Ce paramètre est utilisé pour les communications chiffrées entre les composants Zabbix.
TLSKeyFile	Indiquer le chemin complet du fichier contenant la clé privée de l'agent, utilisée pour les communications chiffrées entre les composants Zabbix.
TLSPSKFile	Indiquer le chemin complet du fichier contenant la clé prépartagée de l'agent, utilisée pour les communications chiffrées avec le serveur ou le proxy Zabbix.
TLSPSKIdentity	Indiquer la chaîne d'identité de la clé prépartagée, utilisée pour les communications chiffrées avec le serveur ou le proxy Zabbix.
TLSServerCertIssuer	Indiquer l'émetteur autorisé du certificat du serveur (proxy).
TLSServerCertSubject	Indiquer le sujet autorisé du certificat du serveur (proxy).
UnsafeUserParameters	Autoriser le passage de tous les caractères dans les arguments des paramètres définis par l'utilisateur.
UserParameter	Indiquer un paramètre défini par l'utilisateur à surveiller.
UserParameterDir	Indiquer le chemin de recherche par défaut des commandes UserParameter.

Notez que :

- Les valeurs par défaut reflètent les valeurs par défaut du processus, et non celles des fichiers de configuration fournis.
- Les valeurs prennent en charge les variables d'environnement.
- Zabbix prend uniquement en charge les fichiers de configuration encodés en UTF-8 sans BOM.
- Les commentaires commençant par # sont pris en charge uniquement au début de la ligne.

Détails du paramètre

Alias

Définissez un alias pour une clé d'élément. Il peut être utilisé pour remplacer une clé d'élément longue et complexe par une clé plus courte et plus simple.
 Plusieurs paramètres `Alias` peuvent être présents. Plusieurs paramètres avec la même clé `Alias` ne sont pas autorisés.
 Différentes clés `Alias` peuvent faire référence à la même clé d'élément.
 Les alias peuvent être utilisés dans le paramètre `HostMetadataItem`, mais pas dans le paramètre `HostnameItem`.

Exemple 1 : récupération de l'ID de l'utilisateur `zabbix`.

```
Alias=zabbix.userid:vfs.file.regexp[/etc/passwd,"^zabbix:.:([0-9]+)",,,,\1]
```

La clé abrégée `zabbix.userid` peut maintenant être utilisée pour récupérer les données.

Exemple 2 : obtention de l'utilisation du processeur avec les paramètres par défaut et personnalisés.

```
Alias=cpu.util:system.cpu.util
Alias=cpu.util[*]:system.cpu.util[*]
```

Cela vous permet d'utiliser la clé `cpu.util` pour obtenir le pourcentage d'utilisation du processeur avec les paramètres par défaut, ainsi que d'utiliser `cpu.util[all, idle, avg15]` pour obtenir des données spécifiques sur l'utilisation du processeur.

Exemple 3 : exécution de plusieurs règles de découverte de bas niveau traitant les mêmes éléments de découverte.

```
Alias=vfs.fs.discovery[*]:vfs.fs.discovery
```

Il est maintenant possible de configurer plusieurs règles de découverte à l'aide de `vfs.fs.discovery`, avec des paramètres différents pour chaque règle, par exemple `vfs.fs.discovery[foo]`, `vfs.fs.discovery[bar]`, etc.

AllowKey

Autoriser l'exécution des clés d'élément qui correspondent à un modèle. Le modèle de clé est une expression générique qui prend en charge le caractère * pour correspondre à n'importe quel nombre de caractères.
Plusieurs règles de correspondance de clés peuvent être définies en combinaison avec `DenyKey`. Plusieurs règles de correspondance de clés peuvent être définies en combinaison avec `DenyKey`, `AllowKeyRegexp` et `DenyKeyRegexp`. Les paramètres sont traités un par un selon leur ordre d'apparition. L'ordre détermine quelle règle correspondante aura la priorité. Voir aussi : Restriction des vérifications de l'agent.

AllowKeyRegexp

Autorise l'exécution des clés d'éléments correspondant à un motif. Le motif de clé est une expression régulière.
 Plusieurs règles de correspondance de clés peuvent être définies en combinaison avec `DenyKeyRegexp`, `AllowKey` et `DenyKey`. Les paramètres sont traités un par un, selon leur ordre d'apparition. L'ordre détermine quelle règle correspondante est prioritaire. Voir également : Restriction des vérifications de l'agent.

Exemple :

```
AllowKeyRegexp=^system\.run\[(?:/usr)?/bin/hostname(?: -[fsd])?\]$
```

BufferSend

Définissez la fréquence d'envoi des valeurs du tampon vers le serveur/proxy Zabbix (en secondes).
 Notez que si le tampon est plein, les données seront envoyées plus tôt.

Par défaut: 5
 Plage: 1-3600

BufferSize

Définit le nombre maximal de valeurs dans le tampon mémoire.
 L'agent enverra toutes les données collectées au serveur Zabbix ou au proxy si le tampon est plein. Ce paramètre ne doit être utilisé que si le tampon persistant est désactivé (`EnablePersistentBuffer=0`).

Par défaut: 1000
 Plage: 2-65535

ControlSocket

Spécifiez le socket de contrôle pour envoyer des commandes d'exécution avec l'option `-R`.

Par défaut: `/tmp/agent.sock`

DebugLevel

Définissez le niveau de débogage : - *0* - informations de base sur le démarrage et l'arrêt des processus Zabbix ; - *1* - informations critiques ; - *2* - informations d'erreur ; - *3* - avertissements ; - *4* - pour le débogage (génère beaucoup d'informations) ; - *5* - débogage étendu (génère encore plus d'informations).

Par défaut : 3
 Plage : 0-5

DenyKey

Interdit l'exécution des clés d'élément correspondant à un modèle.
 Le modèle de clé est une expression générique qui prend en charge le caractère * pour correspondre à un nombre quelconque de caractères.
Plusieurs règles de correspondance de clés peuvent être définies en combinaison avec `AllowKey`. Plusieurs règles de correspondance de clés peuvent être définies en combinaison avec `AllowKey`, `AllowKeyRegexp` et `DenyKeyRegexp`. Les paramètres sont traités un par un, selon leur ordre d'apparition. L'ordre détermine quelle règle correspondante sera prioritaire. Voir également : Restriction des vérifications de l'agent.

Par défaut : `DenyKey=system.run[*]`

DenyKeyRegexp

Interdit l'exécution des clés d'élément correspondant à un modèle.
Le modèle de clé est une expression régulière.
 Plusieurs règles de correspondance de clés peuvent être définies en combinaison avec `AllowKeyRegexp`, `AllowKey` et `DenyKey`. Les paramètres sont traités un par un, selon leur ordre d'apparition. Cet ordre détermine quelle règle correspondante sera prioritaire.
 Tous les éléments `system.run[]` sont interdits par défaut (`DenyKey=system.run[*]`), sauf si une autre règle `system.run[*]` est spécifiée.

Voir également : Restriction des vérifications de l'agent.

Exemple :

```
DenyKeyRegexp=^system\.run\[.*(?:rm -rf|poweroff|reboot).*\]$
```

EnablePersistentBuffer

Activez l'utilisation du stockage persistant local pour les éléments actifs.
 Si le stockage persistant est désactivé, le tampon mémoire sera utilisé.

Par défaut: 0
 Valeurs: 0 - désactivé, 1 - activé

EnableProfiler

Active le profilage périodique de Go au démarrage de l'agent. Le profilage peut également être activé ou désactivé à l'exécution à l'aide de l'option `-R`. Pendant que `periodic_prof_execute` collecte les profils, toutes les commandes d'exécution du profileur sont rejetées car il est occupé.

Par défaut : 0
 Valeurs : 0 - désactivé, 1 - activé

ForceActiveChecksOnStart

Effectue les vérifications actives immédiatement après le redémarrage pour la première configuration reçue.
 Également disponible en tant que paramètre de configuration par plugin, par exemple : `Plugins.Uptime.System.ForceActiveChecksOnStart=1`

Par défaut : 0
 Valeurs : 0 - désactivé, 1 - activé

HeartbeatFrequency

Définit la fréquence des messages de heartbeat en secondes.
 Utilisé pour surveiller la disponibilité des vérifications actives.
0 - messages de heartbeat désactivés.

Par défaut: 60
 Plage: 0-3600

HostInterface

Spécifiez l'interface de l'hôte (adresse IP ou nom DNS) utilisée lors de l'autoregistration.
 Cette valeur sera utilisée pour renseigner l'interface dans le nouvel hôte créé, et permet de configurer explicitement une adresse IP ou DNS.
 Pour plus de détails, voir Using DNS as default interface.

Si elle n'est pas définie, la valeur sera récupérée à partir de `HostInterfaceItem`.

L'agent renverra une erreur et ne démarrera pas si la valeur dépasse la limite de 255 caractères.

Plage: 0-255 caractères

HostInterfaceItem

Spécifiez un élément pour déterminer l'interface de l'hôte (adresse IP ou nom DNS) lors de l'enregistrement automatique de l'hôte.
 Cette valeur est utilisée uniquement si `HostInterface` n'est pas défini.
 Pour plus de détails, consultez Utilisation du DNS comme interface par défaut.

Lors d'une demande d'enregistrement automatique, l'agent consignera un message d'avertissement si la valeur renvoyée par l'élément spécifié dépasse la limite de 255 caractères.

L'élément `system.run[]` est pris en charge quels que soient les paramètres `AllowKey`, `AllowKeyRegexp`, `DenyKey` ou `DenyKeyRegexp`.

HostMetadata

Spécifiez les métadonnées utilisées pour identifier ou distinguer le hôte lors de l'enregistrement automatique du hôte.
 `HostMetadata` permet de distinguer les hôtes au-delà du nom d'hôte.

Si elle n'est pas définie, la valeur sera obtenue à partir de `HostMetadataItem`.

L'agent générera une erreur et ne démarrera pas si la valeur spécifiée dépasse la limite de 2034 octets ou s'il s'agit d'une chaîne qui n'est pas au format UTF-8. Lorsqu'un paramètre attend une adresse IP ou un nom DNS, les valeurs qui sont valides au format UTF-8, mais qui ne sont pas des adresses IP ou des noms DNS valides, seront également rejetées et signalées comme non valides.

Les métadonnées multilignes ne sont pas prises en charge : la sortie sera tronquée au premier saut de ligne.

Plage : 0-2034 octets

HostMetadataItem

Spécifiez un élément permettant d'obtenir les métadonnées de l'hôte.
 Cette option est utilisée uniquement lorsque `HostMetadata` n'est pas défini.

La valeur de `HostMetadataItem` est récupérée à chaque tentative d'enregistrement automatique et utilisée uniquement lors du processus d'enregistrement automatique de l'hôte. `HostMetadataItem` permet de distinguer les hôtes au-delà du nom d'hôte.

Les paramètres utilisateur et les alias sont pris en charge. L'élément `system.run[]` est pris en charge quels que soient les paramètres `AllowKey`, `AllowKeyRegexp`, `DenyKey` ou `DenyKeyRegexp`.

Lors d'une demande d'enregistrement automatique, l'agent consignera un avertissement si la valeur renvoyée par l'élément spécifié dépasse la limite de 65535 points de code UTF-8. La valeur renvoyée par l'élément doit être une chaîne UTF-8 ; sinon, elle sera ignorée. Lorsqu'un paramètre attend une adresse IP ou un nom DNS, les valeurs qui sont valides en UTF-8 mais qui ne correspondent pas à des adresses IP ou à des noms DNS valides seront également rejetées et signalées comme non valides.

Les métadonnées multilignes ne sont pas prises en charge : la sortie sera tronquée au premier saut de ligne.

Nom d'hôte

Spécifiez des noms d'hôte uniques, sensibles à la casse. Une liste délimitée par des virgules.
 Requis pour les vérifications actives et doit correspondre aux noms d'hôte tels qu'ils sont configurés sur le serveur.
 La valeur est obtenue à partir de `HostnameItem` si elle n'est pas définie.
 Caractères autorisés: alphanumériques, '.', ' ', '_' et '-'. Longueur maximale: 128 caractères par nom d'hôte, 2048 caractères pour l'ensemble de la ligne.

Par défaut: défini par `HostnameItem`

HostnameItem

Spécifiez un élément permettant d'obtenir le nom d'hôte.
 Cette option est utilisée uniquement lorsque Hostname n'est pas défini.
 Les paramètres utilisateur ou les alias ne sont pas pris en charge, mais l'élément system.run[] est pris en charge quelles que soient les valeurs de AllowKey, AllowKeyRegexp, DenyKey ou DenyKeyRegexp.

Valeur par défaut : system.hostname

Inclure

Spécifiez un répertoire ou des fichiers individuels à inclure dans le fichier de configuration.
 Lors de l'installation de Zabbix, ce paramètre est défini sur /usr/local/etc, sauf si vous le modifiez au moment de la compilation.

Règles générales pour le paramètre Include :

- Le chemin peut être relatif à l'emplacement du fichier zabbix_agent2.conf.
- Vous pouvez spécifier plusieurs paramètres Include dans un même fichier de configuration.
- Zabbix traite le paramètre Include dès qu'il le lit, donc pensez à le placer à la fin du fichier de configuration.

Règles pour les fichiers inclus :

- Chaque fichier doit être lisible.
- Chaque fichier doit avoir une syntaxe correcte. Sinon, le composant ne démarrera pas.

Règles pour l'inclusion de répertoires ou de modèles de fichiers :

- Ne supposez aucun ordre d'inclusion (par exemple, les fichiers ne sont pas inclus par ordre alphabétique).
- Ne définissez pas le même paramètre dans plusieurs fichiers inclus (par exemple, pour remplacer un réglage général par un réglage spécifique), car vous ne pouvez pas prévoir quelle valeur sera utilisée.
- Faites attention aux copies de sauvegarde que certains éditeurs de texte créent automatiquement (par exemple, la modification de include/my_specific.conf peut créer include/my_specific.conf.BAK), car les deux fichiers seront inclus.

Pour inclure uniquement les fichiers pertinents d'un répertoire, utilisez le caractère générique (*) pour la correspondance de motifs.

Exemple :

```
Include=/absolute/path/to/config/files/*.conf
```

ListenIP

Spécifiez l'adresse IP d'écoute de l'agent pour les connexions provenant du serveur/proxy. Il s'agit d'une liste séparée par des virgules.
 La première adresse IP est envoyée au serveur Zabbix, si la connexion s'y effectue, afin de récupérer la liste des vérifications actives.

Par défaut: 0.0.0.0

ListenPort

Spécifiez le port d'écoute de l'agent pour les connexions provenant du serveur/proxy.

Par défaut: 10050
 Plage: 1024-32767

LogFile

Spécifiez le fichier journal de l'agent.

Par défaut: /tmp/zabbix_agent2.log
 Obligatoire: Oui, si LogType est défini sur file; sinon non

LogFileSize

Définissez la taille maximale du fichier journal (en Mo).
0 - désactive la rotation automatique des journaux.
Remarque : si la limite de taille du fichier journal est atteinte et que la rotation du fichier échoue, quelle qu'en soit la raison, le fichier journal existant est tronqué et recréé.

Par défaut : 1
 Plage : 0-1024

LogType

Spécifiez le type de sortie du journal : - *file* - écrire le journal dans le fichier spécifié par le paramètre LogFile ; - *system* - écrire le journal dans syslog ; - *console* - écrire le journal vers la sortie standard.

Par défaut : file

PersistentBufferFile

Spécifiez le fichier utilisé pour conserver la base de données SQLite. Il doit s'agir d'un nom de fichier complet.
 Ce paramètre est utilisé uniquement si le tampon persistant est activé (EnablePersistentBuffer=1).

PersistentBufferPeriod

Définissez la période de conservation des données lorsqu'il n'y a pas de connexion au serveur Zabbix ou au proxy. Les données plus anciennes seront perdues.
 Les données de journal seront conservées. Ce paramètre est utilisé uniquement si le tampon persistant est activé (EnablePersistentBuffer=1).

Par défaut: 1h
 Plage: 1m-365d

PidFile

Spécifiez le fichier PID.

Par défaut : /tmp/zabbix_agent2.pid

Plugins.<PluginName>.System.Capacity

Définissez la limite de vérifications par plugin <PluginName> qui peuvent être exécutées en même temps.

Par défaut: 1000 Plage: 1-1000

Plugins.Log.MaxLinesPerSecond

Définit le nombre maximal de nouvelles lignes par seconde à envoyer au serveur/proxy Zabbix lors du traitement des contrôles actifs log ou logrt.
 La valeur fournie sera remplacée par le paramètre maxlines, indiqué dans la clé d'élément log et logrt.
Remarque : Zabbix traitera 10 fois plus de nouvelles lignes que la valeur définie dans MaxLinesPerSecond afin de rechercher la chaîne requise dans les éléments de journal.

Valeur par défaut : 20
 Plage : 1-1000

Plugins.SystemRun.LogRemoteCommands

Active l'enregistrement des commandes shell exécutées en tant qu'avertissements.
 Les commandes ne seront consignées que si elles sont exécutées à distance.
 Aucune entrée de journal ne sera créée si system.run[] est lancé localement par les paramètres HostMetadataItem, HostInterfaceItem ou HostnameItem.

Par défaut: 0
 Valeurs: 0 - désactivé, 1 - activé

PluginSocket

Spécifiez le chemin vers le socket UNIX pour les communications du plugin chargeable.

Par défaut : /tmp/agent.plugin.sock

PluginTimeout

Spécifie le délai d'attente pour les connexions aux plugins chargeables, en secondes.

Par défaut: Timeout
 Plage: 1-30

ProfilerDir

Le répertoire dans lequel les données de profilage seront écrites.

Par défaut : /tmp/zabbix_agent2_profiles

ProfilerMaxFilesPerProfile

Le nombre maximal de fichiers à conserver pour chaque type de profil. La limite s'applique aux fichiers terminés après chaque vidage de profil. Le profil CPU actif est collecté dans un fichier temporaire masqué et ne compte pas dans la limite. Les fichiers créés par periodic_prof_execute sont également soumis à la limite.

Valeur par défaut : 10
 Plage : 1-100

ProfilerInterval

L'intervalle d'écriture des données de profilage, en secondes.

Valeur par défaut : 3600
 Plage : 1-86400

RefreshActiveChecks

Spécifie la fréquence d'actualisation de la liste des vérifications actives, en secondes.
 Notez qu'après un échec d'actualisation des vérifications actives, la prochaine tentative d'actualisation sera effectuée dans 60 secondes.

Par défaut: 5
 Plage: 1-86400

Serveur

Spécifiez les adresses IP, éventuellement en notation CIDR, ou les noms DNS des serveurs/proxy Zabbix. Une liste séparée par des virgules.
 Les connexions entrantes ne seront acceptées que depuis les hôtes répertoriés ici. Si la prise en charge d'IPv6 est activée, alors 127.0.0.1, ::127.0.0.1, ::ffff:127.0.0.1 sont traités de la même manière et ::/0 autorisera toute

adresse IPv4 ou IPv6. `0.0.0.0/0` peut être utilisé pour autoriser toute adresse IPv4. Les espaces sont autorisés. Si ce paramètre n'est pas spécifié, les vérifications passives sont désactivées et l'agent n'écoutera sur aucun port TCP.

Exemple :

```
Server=127.0.0.1,192.168.1.0/24,::1,2001:db8::/32,zabbix.example.com
```

ServerActive

Spécifiez l'adresse du serveur/proxy Zabbix ou la configuration du cluster pour obtenir les vérifications actives.
 L'adresse du serveur/proxy est une adresse IP ou un nom DNS, avec un port facultatif séparé par deux-points.
La configuration du cluster est constituée d'une ou plusieurs adresses de membres du groupe de serveurs ou de proxy séparées par des points-virgules. Plusieurs serveurs/clusters Zabbix et proxies Zabbix peuvent être spécifiés, séparés par des virgules. Sauf en cas d'utilisation de groupes de proxy, il ne faut pas spécifier plus d'un proxy Zabbix pour chaque serveur/cluster Zabbix. Si un proxy Zabbix est spécifié, alors le serveur/cluster Zabbix correspondant à ce proxy ne doit pas être spécifié.
Plusieurs adresses séparées par des virgules peuvent être fournies pour utiliser plusieurs serveurs Zabbix indépendants en parallèle. Les espaces sont autorisés.
Si le port n'est pas spécifié, le port par défaut est utilisé.
Les adresses IPv6 doivent être placées entre crochets si le port de cet hôte est spécifié. Si le port n'est pas spécifié, les crochets pour les adresses IPv6 sont facultatifs.
Si ce paramètre n'est pas spécifié, les vérifications actives sont désactivées.

Exemple pour un proxy Zabbix :

```
ServerActive=127.0.0.1:10051
```

Exemple pour un groupe de proxy Zabbix :

```
ServerActive=proxy1.example.com;proxy2.example.com;proxy3.example.com;proxy4.example.com;proxy5.example.co
```

Exemple pour plusieurs serveurs :

```
ServerActive=127.0.0.1:20051,zabbix.domain,[::1]:30051,::1,[12fc::1]
```

Exemple pour la haute disponibilité :

```
ServerActive=zabbix.cluster.node1;zabbix.cluster.node2:20051;zabbix.cluster.node3
```

Exemple pour la haute disponibilité avec deux clusters et un serveur :

```
ServerActive=zabbix.cluster.node1;zabbix.cluster.node2:20051,zabbix.cluster2.node1;zabbix.cluster2.node2,z
```

SourceIP

Spécifiez l'adresse IP source pour :

- les connexions sortantes vers le serveur Zabbix ou le proxy Zabbix.
- l'établissement de connexions lors de l'exécution de certains éléments (`web.page.get`, `net.tcp.port`, etc.).

StatusPort

Spécifiez le port d'écoute pour les requêtes d'état HTTP (`http://localhost:<port>/status`).

Plage : 1024-32767

Délai d'attente

Spécifiez combien de temps attendre (en secondes) pour établir la connexion et échanger des données avec le proxy ou le serveur Zabbix.

Ce paramètre définit la durée de diverses opérations de communication, notamment :

- l'attente d'une réponse du serveur Zabbix ;
- l'envoi de requêtes au serveur Zabbix, y compris les requêtes de configuration d'élément et les données d'élément dans les vérifications actives ;
- la récupération des données de journal via logfile ;
- l'envoi de messages de heartbeat ;
- la durée maximale des vérifications `vfs.*` ;
- le rôle de solution de repli dans les scénarios où un serveur ou un proxy antérieur à la version 7.0 envoie des vérifications sans délais d'attente.

Ce délai d'attente **ne sera pas** utilisé pour les vérifications d'agent qui disposent de paramètres de délai d'attente configurables dans l'interface (au niveau global, du proxy ou de l'élément).

Par défaut : 3
 Plage : 1-30

TLSAccept

Spécifiez le niveau de chiffrement requis pour les connexions entrantes.
 Utilisé pour les vérifications passives.
 Plusieurs valeurs peuvent être spécifiées, séparées par des virgules : - *unencrypted* - accepter les connexions sans chiffrement (par défaut) ; - *psk* - accepter les connexions avec TLS et une clé prépartagée (PSK) ; - *cert* - accepter les connexions avec TLS et un certificat.

Obligatoire : oui, si les paramètres de certificat TLS ou de PSK sont définis (même pour une connexion *unencrypted*) ; sinon non

TLSCAFile

Spécifiez le chemin d'accès complet du fichier contenant les certificats de l'autorité de certification (CA) de niveau supérieur pour la vérification du certificat du pair, utilisé pour les communications chiffrées entre les composants Zabbix.

TLSCertFile

Spécifiez le chemin d'accès complet du fichier contenant le certificat de l'agent ou la chaîne de certificats, utilisé pour les communications chiffrées avec les composants Zabbix.

TLSCipherAll

Spécifiez la chaîne de chiffrement OpenSSL (TLS 1.2). Remplacez les critères de sélection de la suite de chiffrement par défaut pour le chiffrement basé sur les certificats et sur PSK.

Exemple :

```
EECDH+aRSA+AES128:RSA+aRSA+AES128:kECDHEPSK+AES128:kPSK+AES128
```

TLSCipherAll13

Spécifiez la chaîne de chiffrement OpenSSL (TLS 1.3) pour OpenSSL 1.1.1 ou une version plus récente. Remplacez les critères de sélection par défaut de la suite de chiffrement pour le chiffrement basé sur les certificats et sur PSK.

Exemple :

```
TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256
```

TLSCipherCert

Spécifiez la chaîne de chiffrement OpenSSL (TLS 1.2). Remplacez les critères de sélection de la suite de chiffrement par défaut pour le chiffrement basé sur les certificats.

Exemple :

```
EECDH+aRSA+AES128:RSA+aRSA+AES128
```

Notez que ce paramètre ne peut pas être utilisé avec `TLSAccept=cert,psk`; pour une connexion par certificat (`TLSConnect=cert`), utilisez plutôt `TLSCipherAll`.

TLSCipherCert13

Spécifiez la chaîne de chiffrement OpenSSL (TLS 1.3) pour OpenSSL 1.1.1 ou une version plus récente. Remplacez les critères de sélection par défaut de la suite de chiffrement pour le chiffrement basé sur des certificats.

Notez que ce paramètre ne peut pas être utilisé avec `TLSAccept=cert,psk`; pour une connexion par certificat (`TLSConnect=cert`), utilisez plutôt `TLSCipherAll13`.

TLSCipherPSK

Spécifiez la chaîne de chiffrement OpenSSL (TLS 1.2). Remplacez les critères de sélection de la suite de chiffrement par défaut pour le chiffrement basé sur PSK.

Exemple :

```
kECDHEPSK+AES128:kPSK+AES128
```

TLSCipherPSK13

Spécifiez la chaîne de chiffrement OpenSSL (TLS 1.3) pour OpenSSL 1.1.1 ou une version plus récente. Remplacez les critères de sélection par défaut de la suite de chiffrement pour le chiffrement basé sur PSK.

Exemple :

```
TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256
```

TLSConnect

Spécifiez le niveau de chiffrement requis pour les connexions de l'agent au serveur Zabbix ou au proxy.
 Utilisé pour les vérifications actives.
 Une seule valeur peut être spécifiée : - *unencrypted* - connexion sans chiffrement (par défaut) ; - *psk* - connexion à l'aide de TLS et d'une clé prépartagée (PSK) ; - *cert* - connexion à l'aide de TLS et d'un certificat.

Obligatoire : oui, si des paramètres de certificat TLS ou de PSK sont définis (même pour une connexion *unencrypted*) ; sinon non

TLSCRLFile

Spécifiez le chemin d'accès complet du fichier contenant les certificats révoqués. Ce paramètre est utilisé pour les communications chiffrées entre les composants Zabbix.

TLSKeyFile

Spécifiez le chemin d'accès complet du fichier contenant la clé privée de l'agent, utilisée pour les communications chiffrées entre les composants Zabbix.

TLSPSKFile

Spécifiez le chemin d'accès complet du fichier contenant la clé pré-partagée de l'agent, utilisée pour les communications chiffrées avec le serveur ou le proxy Zabbix.

TLSPSKIdentity

Spécifiez la chaîne d'identité de clé pré-partagée, utilisée pour les communications chiffrées avec le serveur Zabbix ou le proxy.

TLSServerCertIssuer

Spécifiez l'émetteur du certificat du serveur (proxy) autorisé.

TLSServerCertSubject

Spécifiez le sujet de certificat autorisé du serveur (proxy).

UnsafeUserParameters

Autoriser tous les caractères à être transmis dans les arguments des paramètres définis par l'utilisateur. Les caractères suivants ne sont pas autorisés : \ ' " ` * ? [ ] { } ~ $! & ; ( ) < > | # @ %
De plus, les caractères de nouvelle ligne ne sont pas autorisés.

Par défaut : 0
 Valeurs : 0 - ne pas autoriser, 1 - autoriser

UserParameter

Spécifiez un paramètre défini par l'utilisateur à surveiller. Il peut y avoir plusieurs paramètres définis par l'utilisateur.
Format : `UserParameter=<key>,<shell command>`
Notez que la commande shell ne doit pas renvoyer une chaîne vide ni uniquement un EOL. Les commandes shell peuvent utiliser des chemins relatifs si le paramètre `UserParameterDir` est spécifié.

Exemple :

```
UserParameter=system.test,who|wc -l
UserParameter=check_cpu,./custom_script.sh
```

UserParameterDir

Spécifie le chemin de recherche par défaut pour les commandes `UserParameter`. Si cette option est utilisée, l'agent change son répertoire de travail pour celui indiqué ici avant d'exécuter une commande. Ainsi, les commandes UserParameter peuvent utiliser un préfixe relatif `./` au lieu d'un chemin complet.
Une seule entrée est autorisée.

Exemple :

```
UserParameterDir=/opt/myscripts
```

Paramètres de configuration: Zabbix agent 2 (Windows)

Vue d'ensemble

Les paramètres de configuration permettent de personnaliser le comportement de Zabbix agent 2 sous Windows.

Indiquez les valeurs des paramètres dans le fichier de configuration de l'agent 2 de Zabbix (`zabbix_agent2.conf`). Tous les paramètres sont **facultatifs**, sauf indication explicite contraire.

Cette page est structurée comme suit :

- Vue d'ensemble des paramètres (pour afficher les détails complets d'un paramètre, cliquez sur son nom)
- Détails des paramètres

Paramètre	Description
Alias	Définir un alias pour une clé d'élément.
AllowKey	Autoriser l'exécution des clés d'élément correspondant à un modèle.
AllowKeyRegexp	Autoriser l'exécution des clés d'élément correspondant à un modèle basé sur une expression régulière.

Paramètre	Description
BufferSend	Définir la fréquence d'envoi des valeurs de la mémoire tampon au serveur Zabbix.
BufferSize	Définir le nombre maximal de valeurs dans la mémoire tampon.
ControlSocket	Indiquer le socket de contrôle utilisé pour envoyer des commandes d'exécution avec l'option `-R`.
DebugLevel	Définir le niveau de débogage.
DenyKey	Interdire l'exécution des clés d'élément correspondant à un modèle.
DenyKeyRegexp	Interdire l'exécution des clés d'élément correspondant à un modèle basé sur une expression régulière.
EnablePersistentBuffer	Activer l'utilisation d'un stockage local persistant pour les éléments actifs.
EnableProfiler	Activer le profilage périodique de Go au démarrage de l'agent.
ForceActiveChecksOnStart	Effectuer immédiatement les vérifications actives après le redémarrage pour la première configuration reçue.
HeartbeatFrequency	Définir la fréquence des messages de pulsation, en secondes.
HostInterface	Indiquer l'interface de l'hôte.
HostInterfaceItem	Indiquer un élément permettant d'obtenir l'interface de l'hôte.
HostMetadata	Indiquer les métadonnées de l'hôte.
HostMetadataItem	Indiquer un élément d'agent Zabbix utilisé pour obtenir les métadonnées de l'hôte.
Hostname	Indiquer le nom d'hôte.
HostnameItem	Indiquer un élément d'agent Zabbix permettant d'obtenir le nom d'hôte.
Include	Indiquer un répertoire ou des fichiers individuels à inclure dans le fichier de configuration.
ListenIP	Indiquer l'adresse IP d'écoute de l'agent pour les connexions provenant du serveur/proxy.
ListenPort	Indiquer le port d'écoute de l'agent.
LogFile	Indiquer le fichier journal de l'agent.
LogFileSize	Définir la taille maximale du fichier journal.
LogType	Indiquer le type de sortie du journal.
PerfCounter	Indiquer un nouveau paramètre <parameter_name> correspondant à la valeur moyenne du compteur de performances système <perf_counter_path> pour la période spécifiée <period> (en secondes).
PerfCounterEn	Indiquer un nouveau paramètre <parameter_name> correspondant à la valeur moyenne du compteur de performances système <perf_counter_path> pour la période spécifiée <period> (en secondes). Contrairement à PerfCounter, les chemins des compteurs de performances doivent être en anglais.
PersistentBufferFile	Indiquer le fichier dans lequel Zabbix agent 2 doit conserver la base de données SQLite.
PersistentBufferPeriod	Définir la durée de conservation des données lorsqu'il n'existe aucune connexion au serveur ou au proxy.
Plugins.<PluginName>.SystemCapacity	Définir la limite du nombre de vérifications par plugin pouvant être exécutées simultanément.
Plugins.Log.MaxLinesPerSecond	Définir le nombre maximal de nouvelles lignes que l'agent enverra par seconde au serveur Zabbix ou au proxy lors du traitement des vérifications actives `log`, `logrt` et `eventlog`.
Plugins.SystemRun.LogRemoteCommands	Activer la journalisation des commandes shell exécutées en tant qu'avertissements.
PluginSocket	Indiquer le chemin du canal nommé Windows utilisé pour les communications avec les plugins chargeables.
PluginTimeout	Définir le délai d'expiration des connexions avec les plugins chargeables, en secondes.
ProfilerDir	Répertoire dans lequel les données de profilage seront écrites.
ProfilerMaxFilesPerProfile	Nombre maximal de fichiers à conserver pour chaque type de profil.
ProfilerInterval	Intervalle d'écriture des données de profilage.
RefreshActiveChecks	Indiquer la fréquence d'actualisation de la liste des vérifications actives.
Server	Indiquer les adresses IP ou les noms DNS des serveurs/proxys Zabbix.
ServerActive	Indiquer l'adresse du serveur/proxy Zabbix ou la configuration du cluster depuis laquelle obtenir les vérifications actives.
SourceIP	Indiquer l'adresse IP source.
StatusPort	Définir le port sur lequel l'agent écoute les requêtes d'état HTTP (`http://localhost:<port>/status`).
Timeout	Indiquer le délai d'attente (en secondes) pour établir la connexion et échanger des données avec le serveur ou le proxy Zabbix.
TLSAccept	Indiquer le niveau de chiffrement requis pour les connexions entrantes.
TLSCAFile	Indiquer le chemin complet du fichier contenant les certificats de l'autorité de certification racine utilisés pour vérifier les certificats des pairs, dans le cadre des communications chiffrées entre les composants Zabbix.
TLSCertFile	Indiquer le chemin complet du fichier contenant le certificat de l'agent ou la chaîne de certificats, utilisé pour les communications chiffrées entre les composants Zabbix.
TLSCipherAll13	Indiquer la chaîne de chiffrement OpenSSL (TLS 1.3). Remplacer les critères de sélection de la suite de chiffrement par défaut pour le chiffrement basé sur des certificats et des clés prépartagées (PSK).

Paramètre	Description
TLSCipherCert13	Indiquer la chaîne de chiffrement OpenSSL (TLS 1.3). Remplacer les critères de sélection de la suite de chiffrement par défaut pour le chiffrement basé sur des certificats.
TLSCipherPSK13	Indiquer la chaîne de chiffrement OpenSSL (TLS 1.3). Remplacer les critères de sélection de la suite de chiffrement par défaut pour le chiffrement basé sur des clés prépartagées (PSK).
TLSConnect	Indiquer le niveau de chiffrement requis pour les connexions au serveur ou au proxy Zabbix.
TLSCRLFile	Indiquer le chemin complet du fichier contenant les certificats révoqués. Ce paramètre est utilisé pour les communications chiffrées entre les composants Zabbix.
TLSKeyFile	Indiquer le chemin complet du fichier contenant la clé privée de l'agent, utilisée pour les communications chiffrées entre les composants Zabbix.
TLSPSKFile	Indiquer le chemin complet du fichier contenant la clé prépartagée de l'agent, utilisée pour les communications chiffrées avec le serveur ou le proxy Zabbix.
TLSPSKIdentity	Indiquer la chaîne d'identité de la clé prépartagée, utilisée pour les communications chiffrées avec le serveur ou le proxy Zabbix.
TLSServerCertIssuer	Indiquer l'émetteur autorisé du certificat du serveur (proxy).
TLSServerCertSubject	Indiquer le sujet autorisé du certificat du serveur (proxy).
UnsafeUserParameters	Autoriser le passage de tous les caractères dans les arguments des paramètres définis par l'utilisateur.
UserParameter	Indiquer un paramètre défini par l'utilisateur à surveiller.
UserParameterDir	Indiquer le chemin de recherche par défaut des commandes `UserParameter`.

Notez que :

- Les valeurs par défaut reflètent les valeurs par défaut du processus, et non celles des fichiers de configuration fournis ;
- Les valeurs prennent en charge les variables d'environnement ;
- Zabbix prend uniquement en charge les fichiers de configuration encodés en UTF-8 sans BOM ;
- Les commentaires commençant par # sont pris en charge uniquement au début de la ligne.

Détails du paramètre

Alias

Définissez un alias pour une clé d'élément. Il peut être utilisé pour remplacer une clé d'élément longue et complexe par une clé plus courte et plus simple.
 Plusieurs paramètres `Alias` peuvent être présents. Plusieurs paramètres avec la même clé `Alias` ne sont pas autorisés.
 Différentes clés `Alias` peuvent faire référence à la même clé d'élément.
 Les alias peuvent être utilisés dans `HostMetadataItem`, mais pas dans le paramètre `HostnameItem`.

Exemple 1 : récupération de l'utilisation du fichier d'échange en pourcentage depuis le serveur.

```
Alias=pg_usage:perf_counter[\Paging File(_Total)\% Usage]
```

La clé abrégée `pg_usage` peut maintenant être utilisée pour récupérer les données.

Exemple 2 : obtention de la charge du processeur avec les paramètres par défaut et personnalisés.

```
Alias=cpu.load:system.cpu.load
Alias=cpu.load[*]:system.cpu.load[*]
```

Cela vous permet d'utiliser la clé `cpu.load` pour obtenir la charge du processeur avec les paramètres par défaut, ainsi que `cpu.load[percpu,avg15]` pour obtenir des données spécifiques sur la charge du processeur.

Exemple 3 : exécution de plusieurs règles de découverte de bas niveau traitant les mêmes éléments de découverte.

```
Alias=vfs.fs.discovery[*]:vfs.fs.discovery
```

Il est maintenant possible de configurer plusieurs règles de découverte à l'aide de `vfs.fs.discovery`, avec des paramètres différents pour chaque règle, par exemple `vfs.fs.discovery[foo]`, `vfs.fs.discovery[bar]`, etc.

AllowKey

Autoriser l'exécution des clés d'élément qui correspondent à un modèle. Le modèle de clé est une expression générique qui prend en charge le caractère * pour correspondre à n'importe quel nombre de caractères.
 Plusieurs règles de correspondance de clés peuvent être définies en combinaison avec `DenyKey`, `AllowKeyRegexp` et `DenyKeyRegexp`. Les paramètres sont traités un par un selon leur ordre d'apparition. L'ordre détermine quelle règle de correspondance aura la priorité. Voir aussi: Restriction des vérifications de l'agent.

AllowKeyRegexp

Autorise l'exécution des clés d'éléments correspondant à un motif. Le motif de clé est une expression régulière.
 Plusieurs règles de correspondance de clés peuvent être définies en combinaison avec `DenyKeyRegexp`, `AllowKey` et `DenyKey`. Les paramètres sont traités un par un, selon leur ordre d'apparition. L'ordre détermine quelle règle correspondante sera prioritaire. Voir également : Restriction des vérifications de l'agent.

Exemple :

```
AllowKeyRegexp=^system\.run\[(?:/usr)?/bin/hostname(?: -[fsd])?\]$
```

BufferSend

Définissez la fréquence d'envoi des valeurs du tampon vers le serveur/proxy Zabbix (en secondes).
Notez que si le tampon est plein, les données seront envoyées plus tôt.

Par défaut: 5
 Plage: 1-3600

BufferSize

Définit le nombre maximal de valeurs dans le tampon mémoire.
 L'agent enverra toutes les données collectées au serveur Zabbix ou au proxy si le tampon est plein.
Ce paramètre ne doit être utilisé que si le tampon persistant est désactivé (`EnablePersistentBuffer=0`).

Par défaut: 1000
 Plage: 2-65535

ControlSocket

Spécifiez le socket de contrôle, utilisé pour envoyer des commandes d'exécution avec l'option `-R`.

Par défaut: `\\.\pipe\agent.sock`

DebugLevel

Définissez le niveau de débogage : - *0* - informations de base sur le démarrage et l'arrêt des processus Zabbix ; - *1* - informations critiques ; - *2* - informations d'erreur ; - *3* - avertissements ; - *4* - pour le débogage (génère beaucoup d'informations) ; - *5* - débogage étendu (génère encore plus d'informations).

Par défaut : 3
 Plage : 0-5

DenyKey

Interdit l'exécution des clés d'élément correspondant à un modèle. Le modèle de clé est une expression générique qui prend en charge le caractère `*` pour correspondre à un nombre quelconque de caractères.
 Plusieurs règles de correspondance de clés peuvent être définies en combinaison avec `AllowKey`, `AllowKeyRegexp` et `DenyKeyRegexp`. Les paramètres sont traités un par un, selon leur ordre d'apparition. L'ordre détermine quelle règle correspondante est prioritaire.

Voir aussi : Restriction des vérifications de l'agent.

Par défaut : `DenyKey=system.run[*]`

DenyKeyRegexp

Refuser l'exécution des clés d'élément correspondant à un motif. Le motif de clé est une expression régulière.
 Plusieurs règles de correspondance de clés peuvent être définies en combinaison avec `AllowKeyRegexp`, `AllowKey` et `DenyKey`. Les paramètres sont traités un par un, selon leur ordre d'apparition. L'ordre détermine quelle règle correspondante sera prioritaire.
 Tous les éléments `system.run[]` sont refusés par défaut (`DenyKey=system.run[*]`), sauf si une autre règle `system.run[*]` est spécifiée.

Voir aussi : Restriction des vérifications de l'agent.

EnablePersistentBuffer

Activez l'utilisation du stockage persistant local pour les éléments actifs. Si le stockage persistant est désactivé, le tampon mémoire sera utilisé.

Par défaut: 0
 Valeurs: 0 - désactivé, 1 - activé

EnableProfiler

Active le profilage périodique de Go au démarrage de l'agent. Le profilage peut également être activé ou désactivé à l'exécution à l'aide de l'option `-R`. Pendant que `periodic_prof_execute` collecte les profils, toutes les commandes d'exécution du profileur sont rejetées car il est occupé.

Par défaut : 0
 Valeurs : 0 - désactivé, 1 - activé

ForceActiveChecksOnStart

Effectue les vérifications actives immédiatement après le redémarrage pour la première configuration reçue. Également disponible en tant que paramètre de configuration par plugin, par exemple : `Plugins.Uptime.System.ForceActiveChecksOnStart=1`

Par défaut : 0
 Valeurs : 0 - désactivé, 1 - activé

HeartbeatFrequency

Définissez la fréquence des messages de heartbeat en secondes. Utilisé pour surveiller la disponibilité des vérifications actives.
0 - messages de heartbeat désactivés.

Par défaut: 60
 Plage: 0-3600

HostInterface

Spécifiez l'interface de l'hôte (adresse IP ou nom DNS) utilisée lors de l'auto-enregistrement de l'hôte. Cette valeur sera utilisée pour renseigner l'interface dans le nouvel hôte créé, et permet de configurer explicitement une adresse IP ou DNS. Pour plus de détails, consultez Utiliser DNS comme interface par défaut.

Si elle n'est pas définie, la valeur sera obtenue à partir de HostInterfaceItem.

L'agent signalera une erreur et ne démarrera pas si la valeur dépasse la limite de 255 caractères.

Plage : 0-255 caractères

HostInterfaceItem

Spécifiez un élément permettant de déterminer l'interface du hôte (adresse IP ou nom DNS) lors de l'enregistrement automatique du hôte. Cette valeur est utilisée uniquement si `HostInterface` n'est pas défini. Pour plus de détails, consultez Utilisation du DNS comme interface par défaut.

Lors d'une demande d'enregistrement automatique, l'agent consignera un message d'avertissement si la valeur renvoyée par l'élément spécifié dépasse la limite de 255 caractères.

L'élément `system.run[]` est pris en charge quels que soient les paramètres `AllowKey`, `AllowKeyRegexp`, `DenyKey` ou `DenyKeyRegexp`.

HostMetadata

Spécifiez les métadonnées utilisées pour identifier ou distinguer le hôte lors de l'enregistrement automatique du hôte (agent actif). `HostMetadata` permet de distinguer les hôtes au-delà du nom d'hôte.

Si elle n'est pas définie, la valeur sera obtenue à partir de `HostMetadataItem`.

L'agent générera une erreur et ne démarrera pas si la valeur spécifiée dépasse la limite de 2034 octets ou s'il s'agit d'une chaîne qui n'est pas au format UTF-8. Lorsqu'un paramètre attend une adresse IP ou un nom DNS, les valeurs qui sont valides au format UTF-8, mais qui ne sont pas des adresses IP ou des noms DNS valides, seront également rejetées et signalées comme non valides.

Les métadonnées multilignes ne sont pas prises en charge : la sortie sera tronquée au premier saut de ligne.

Plage : 0-2034 octets

HostMetadataItem

Spécifiez un élément de l'agent Zabbix pour obtenir les métadonnées de l'hôte. Cette option est utilisée uniquement lorsque `HostMetadata` n'est pas défini.

La valeur de `HostMetadataItem` est récupérée à chaque tentative d'enregistrement automatique et utilisée uniquement lors du processus d'enregistrement automatique de l'hôte (agent actif). `HostMetadataItem` permet de distinguer les hôtes au-delà du nom d'hôte.

Les paramètres utilisateur et les alias sont pris en charge. L'élément `system.run[]` est pris en charge quels que soient les paramètres `AllowKey`, `AllowKeyRegexp`, `DenyKey` ou `DenyKeyRegexp`.

Lors d'une demande d'enregistrement automatique, l'agent consignera un avertissement si la valeur renvoyée par l'élément spécifié dépasse la limite de 65535 points de code UTF-8. La valeur renvoyée par l'élément doit être une chaîne UTF-8 ; sinon, elle sera ignorée. Lorsqu'un paramètre attend une adresse IP ou un nom DNS, les valeurs qui sont valides en UTF-8 mais qui ne sont pas des adresses IP ou des noms DNS valides seront également rejetées et signalées comme non valides.

Les métadonnées multilignes ne sont pas prises en charge : la sortie sera tronquée au premier saut de ligne.

Nom d'hôte

Spécifiez des noms d'hôte uniques, sensibles à la casse. Une liste délimitée par des virgules.
 Requis pour les vérifications actives et doit correspondre aux noms d'hôte tels qu'ils sont configurés sur le serveur. La valeur est obtenue à partir de HostnameItem si elle n'est pas définie.
 Caractères autorisés: alphanumériques, '.', ' ', '_' et '-'. Longueur maximale: 128 caractères par nom d'hôte, 2048 caractères pour l'ensemble de la ligne.

Par défaut: défini par HostnameItem

HostnameItem

Spécifiez un élément permettant d'obtenir le nom d'hôte. Cette option est utilisée uniquement lorsque `Hostname` n'est pas défini. Les paramètres utilisateur et les alias ne sont pas pris en charge, mais l'élément `system.run[]` est pris en charge quels que soient les valeurs de `AllowKey`, `AllowKeyRegexp`, `DenyKey` ou `DenyKeyRegexp`.

Valeur par défaut : `system.hostname`

Include

Indiquez un répertoire ou des fichiers individuels à inclure dans le fichier de configuration (situé par défaut dans `C:\Program Files\Zabbix Agent 2` si l'agent Zabbix est installé à l'aide des packages d'installation MSI pour Windows ; situé dans le dossier spécifié lors de l'installation si l'agent Zabbix est installé sous forme d'archive zip). Lors de l'installation de Zabbix, ce paramètre est défini sur `/usr/local/etc`, sauf si vous modifiez cette valeur lors de la compilation.

Règles générales pour le paramètre `Include` :

- Le chemin peut être relatif à l'emplacement du fichier `zabbix_agent2.conf` (par exemple, `Include=.\zabbix_agent2.d\plugins`
- Vous pouvez spécifier plusieurs paramètres `Include` dans un même fichier de configuration.
- Zabbix traite le paramètre `Include` dès qu'il le lit. Il est donc recommandé de le placer à la fin du fichier de configuration.

Règles applicables aux fichiers inclus :

- Chaque fichier doit être lisible.
- Chaque fichier doit avoir une syntaxe correcte. Sinon, le composant ne démarrera pas.

Règles applicables à l'inclusion de répertoires ou de modèles de fichiers :

- Ne présumez pas d'un ordre d'inclusion particulier (par exemple, les fichiers ne sont pas inclus dans l'ordre alphabétique).
- Ne définissez pas le même paramètre dans plusieurs fichiers inclus (par exemple, pour remplacer un paramètre général par un paramètre spécifique), car vous ne pouvez pas prévoir quelle valeur sera utilisée.
- Faites attention aux copies de sauvegarde que certains éditeurs de texte créent automatiquement (par exemple, la modification de `include/my_specific.conf` peut créer `include/my_specific.conf.BAK`), car les deux fichiers seront inclus.

Pour inclure uniquement les fichiers pertinents d'un répertoire, utilisez le caractère générique (`*`) pour la correspondance de modèles.

Exemple :

```
Include=C:\Program Files\Zabbix Agent2\zabbix_agent2.d\*.conf
```

ListenIP

Spécifiez l'adresse IP d'écoute de l'agent pour les connexions provenant du serveur/proxy. Une liste séparée par des virgules.
 La première adresse IP est envoyée au serveur Zabbix, si la connexion s'effectue vers celui-ci, afin de récupérer la liste des vérifications actives.

Par défaut: `0.0.0.0`

ListenPort

Spécifiez le port d'écoute de l'agent pour les connexions provenant du serveur/proxy.

Par défaut: 10050
 Plage: 1024-32767

LogFile

Spécifiez le fichier journal de l'agent.

Par défaut: `c:\zabbix_agent2.log`
 Obligatoire: Oui, si `LogType` est défini sur `file`; sinon non

LogFileSize

Définissez la taille maximale du fichier journal (en Mo).
0 - désactive la rotation automatique des journaux.
Remarque : si la limite de taille du fichier journal est atteinte et que la rotation du fichier échoue, quelle qu'en soit la raison, le fichier journal existant est tronqué et recréé.

Par défaut : 1
 Plage : 0-1024

LogType

Spécifiez le type de sortie du journal : - *file* - écrire le journal dans le fichier spécifié par le paramètre `LogFile` ; - *console* - écrire le journal sur la sortie standard.

Par défaut : `file`

PerfCounter

Définissez un nouveau paramètre <parameter_name> qui correspond à la valeur moyenne du compteur de performances système <perf_counter_path> pour la période spécifiée <period> (en secondes).
Syntaxe: <parameter_name>,"<perf_counter_path>",<period>

Par exemple, si vous souhaitez obtenir le nombre moyen d'interruptions du processeur par seconde pour la dernière minute, vous pouvez définir un nouveau paramètre "interrupts" comme suit:

```
PerfCounter = interrupts,"\Processor(0)\Interrupts/sec",60
```

Veuillez noter les guillemets doubles autour du chemin du compteur de performances. Le nom du paramètre (interrupts) doit être utilisé comme clé d'élément lors de la création d'un élément. Les échantillons utilisés pour calculer la valeur moyenne seront prélevés chaque seconde.
Vous pouvez exécuter `typeperf -qx` pour obtenir la liste de tous les compteurs de performances disponibles dans Windows.

PerfCounterEn

Définissez un nouveau paramètre <parameter_name> qui correspond à la valeur moyenne du compteur de performances système <perf_counter_path> pour la période spécifiée <period> (en secondes). Par rapport à PerfCounter, les chemins de perfcounter doivent être en anglais. Pris en charge uniquement sur **Windows Server 2008/Vista** et versions ultérieures.
Syntaxe: <parameter_name>,"<perf_counter_path>",<period>

Par exemple, si vous souhaitez recevoir le nombre moyen d'interruptions du processeur par seconde pour la dernière minute, vous pouvez définir un nouveau paramètre "interrupts" comme suit :

```
PerfCounterEn = interrupts,"\Processor(0)\Interrupts/sec",60
```

Veuillez noter les guillemets doubles autour du chemin du compteur de performances. Le nom du paramètre (interrupts) doit être utilisé comme clé d'élément lors de la création d'un élément. Les échantillons utilisés pour calculer la valeur moyenne seront prélevés chaque seconde.
Vous pouvez trouver la liste des chaînes en anglais en consultant la clé de registre suivante : `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Perflib\009`.

PersistentBufferFile

Spécifiez le fichier dans lequel l'agent Zabbix 2 doit conserver la base de données SQLite. Doit être un nom de fichier complet.
 Ce paramètre n'est utilisé que si le tampon persistant est activé (`EnablePersistentBuffer=1`).

PersistentBufferPeriod

Définit la durée de conservation des données lorsqu'il n'y a pas de connexion au serveur ou au proxy. Les données plus anciennes seront perdues. Les données de journal seront conservées.
 Ce paramètre est utilisé uniquement si le tampon persistant est activé (`EnablePersistentBuffer=1`).

Par défaut: 1h
 Plage: 1m-365d

Plugins.<PluginName>.System.Capacity

Définissez la limite du nombre de vérifications par plugin `<PluginName>` qui peuvent être exécutées en même temps.

Par défaut: 1000 Plage: 1-1000

Plugins.Log.MaxLinesPerSecond

Définit le nombre maximal de nouvelles lignes que l'agent enverra par seconde au serveur Zabbix ou au proxy lors du traitement des vérifications actives `log`, `logrt` et `eventlog`. La valeur fournie sera remplacée par le paramètre `maxlines`, indiqué dans la clé d'élément `log`, `logrt` ou `eventlog`.
Remarque : Zabbix traitera 10 fois plus de nouvelles lignes que la valeur définie dans `MaxLinesPerSecond` afin de rechercher la chaîne requise dans les éléments de journal.

Par défaut : 20
 Plage : 1-1000

Plugins.SystemRun.LogRemoteCommands

Active l'enregistrement des commandes shell exécutées en tant qu'avertissements. Les commandes seront consignées uniquement si elles sont exécutées à distance. Aucune entrée de journal ne sera créée si `system.run[]` est lancé localement par les paramètres `HostMetadataItem`, `HostInterfaceItem` ou `HostnameItem`.

Par défaut: 0
 Valeurs: 0 - désactivé, 1 - activé

PluginSocket

Spécifiez le chemin du pipe nommé Windows pour les communications des plugins chargeables.

Par défaut : `\\.\pipe\agent.plugin.sock`

PluginTimeout

Définit le délai d'attente pour les connexions avec des plugins chargeables, en secondes.

Par défaut: `Timeout<br>` Plage: 1-30

ProfilerDir

Le répertoire dans lequel les données de profilage seront écrites.

Par défaut : `c:\zabbix_pprof<br>`

ProfilerMaxFilesPerProfile

Le nombre maximal de fichiers à conserver pour chaque type de profil. La limite s'applique aux fichiers terminés après chaque vidage de profil. Le profil CPU actif est collecté dans un fichier temporaire masqué et ne compte pas dans la limite. Les fichiers créés par `periodic_prof_execute` sont également soumis à la limite.

Valeur par défaut : 10
 Plage : 1-100

ProfilerInterval

L'intervalle d'écriture des données de profilage, en secondes.

Valeur par défaut : 3600
 Plage : 1-86400

RefreshActiveChecks

Spécifiez la fréquence d'actualisation de la liste des vérifications actives, en secondes.
 Notez qu'après un échec d'actualisation des vérifications actives, la prochaine actualisation sera tentée dans 60 secondes.

Par défaut: 5
 Plage: 1-86400

Serveur

Spécifiez les adresses IP, éventuellement en notation CIDR, ou les noms DNS des serveurs/proxy Zabbix. Une liste séparée par des virgules.
 Si le paramètre n'est pas spécifié, les vérifications passives sont désactivées et l'agent n'écoutera sur aucun port TCP.
 Les connexions entrantes ne seront acceptées que depuis les hôtes répertoriés ici. Si la prise en charge d'IPv6 est activée, alors `127.0.0.1`, `::127.0.0.1`, `::ffff:127.0.0.1` sont traités de la même manière et `::/0` autorisera n'importe quelle adresse IPv4 ou IPv6. `0.0.0.0/0` peut être utilisé pour autoriser n'importe quelle adresse IPv4. Les espaces sont autorisés.

Exemple :

```
Server=127.0.0.1,192.168.1.0/24,::1,2001:db8::/32,zabbix.example.com
```

ServerActive

Spécifiez l'adresse du serveur/proxy Zabbix ou la configuration du cluster à partir duquel obtenir les vérifications actives.
 Si le paramètre n'est pas spécifié, les vérifications actives sont désactivées.

L'adresse du serveur/proxy est une adresse IP ou un nom DNS, avec éventuellement un port séparé par deux-points. Si le port n'est pas spécifié, le port par défaut est utilisé.
 Les adresses IPv6 doivent être placées entre crochets si un port est spécifié pour cet hôte. Si le port n'est pas spécifié, les crochets pour les adresses IPv6 sont facultatifs.

La configuration du cluster est constituée d'une ou plusieurs adresses de membres du groupe de serveur ou de proxy, séparées par des points-virgules.

Plusieurs serveurs/clusters Zabbix et proxies Zabbix peuvent être spécifiés, séparés par des virgules.

Sauf en cas d'utilisation de groupes de proxy, il ne faut pas spécifier plus d'un proxy Zabbix pour chaque serveur/cluster Zabbix. Si un proxy Zabbix est spécifié, alors le serveur/cluster Zabbix correspondant à ce proxy ne doit pas être spécifié.

Plusieurs adresses séparées par des virgules peuvent être fournies pour utiliser plusieurs serveurs Zabbix indépendants en parallèle. Les espaces sont autorisés.

Exemple pour un proxy Zabbix :

```
ServerActive=127.0.0.1:10051
```

Exemple pour un groupe de proxy Zabbix :

```
ServerActive=proxy1.example.com;proxy2.example.com;proxy3.example.com;proxy4.example.com;proxy5.example.co
```

Exemple pour plusieurs serveurs :

```
ServerActive=127.0.0.1:20051,zabbix.domain,\[::1\]:30051,::1,\[12fc::1\]
```

Exemple pour la haute disponibilité :

```
ServerActive=zabbix.cluster.node1;zabbix.cluster.node2:20051;zabbix.cluster.node3
```

Exemple pour la haute disponibilité avec deux clusters et un serveur :

```
ServerActive=zabbix.cluster.node1;zabbix.cluster.node2:20051,zabbix.cluster2.node1;zabbix.cluster2.node2,z
```

SourceIP

Spécifiez l'adresse IP source pour :

- les connexions sortantes vers le serveur Zabbix ou le proxy Zabbix ;
- l'établissement de connexions lors de l'exécution de certains éléments (`web.page.get`, `net.tcp.port`, etc.).

StatusPort

Définissez le port sur lequel l'agent écoute les requêtes d'état HTTP (`http://localhost:<port>/status`).

Plage : 1024-32767

Délai d'attente

Spécifiez la durée d'attente (en secondes) pour établir la connexion et échanger des données avec le proxy ou le serveur Zabbix.

Ce paramètre définit la durée de diverses opérations de communication, notamment :

- l'attente d'une réponse du serveur Zabbix ;
- l'envoi de requêtes au serveur Zabbix, y compris les requêtes de configuration des éléments et les données des éléments dans les vérifications actives ;
- la récupération des données de journal via la surveillance des fichiers journaux ou du journal des événements Windows ;
- l'envoi de messages de battement de coeur ;
- la durée maximale des vérifications `vfs.*` ;
- le rôle de solution de repli dans les scénarios où un serveur ou un proxy antérieur à la version 7.0 envoie des vérifications sans délais d'attente.

Ce délai d'attente ne sera **pas** utilisé pour les vérifications d'agent qui disposent de paramètres de délai d'attente configurables dans l'interface (au niveau global, du proxy ou de chaque élément).

Par défaut: 3
 Plage: 1-30

TLSAccept

Spécifiez le niveau de chiffrement requis pour les connexions entrantes.
 Utilisé pour les vérifications passives.

Plusieurs valeurs peuvent être spécifiées, séparées par des virgules :

- *unencrypted* - accepter les connexions sans chiffrement (par défaut) ;
- *psk* - accepter les connexions avec TLS et une clé prépartagée (PSK) ;
- *cert* - accepter les connexions avec TLS et un certificat.

Obligatoire : oui, si des paramètres de certificat TLS ou de PSK sont définis (même pour une connexion *unencrypted*) ; sinon non

TLSCAFile

Spécifiez le chemin complet du fichier contenant les certificats de l'autorité de certification racine pour la vérification du certificat du pair, utilisé pour les communications chiffrées entre les composants Zabbix.

TLSCertFile

Spécifiez le chemin d'accès complet du fichier contenant le certificat de l'agent ou la chaîne de certificats, utilisé pour les communications chiffrées avec les composants Zabbix.

TLSCipherAll13

Spécifiez la chaîne de chiffrement OpenSSL (TLS 1.3). Remplacez les critères de sélection de la suite de chiffrement par défaut pour le chiffrement basé sur les certificats et sur PSK.

Exemple :

```
TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256
```

TLSCipherCert13

Spécifiez la chaîne de chiffrement OpenSSL (TLS 1.3). Remplacez les critères de sélection par défaut de la suite de chiffrement pour le chiffrement basé sur les certificats.

Notez que ce paramètre ne peut pas être utilisé avec `TLSAccept=cert,psk` ; pour une connexion par certificat (`TLSConnect=cert`), utilisez plutôt `TLSCipherAll13`.

TLSCipherPSK13

Spécifiez la chaîne de chiffrement OpenSSL (TLS 1.3). Remplacez les critères de sélection par défaut de la suite de chiffrement pour le chiffrement basé sur PSK.

Exemple :

```
TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256
```

TLSConnect

Spécifiez le niveau de chiffrement requis pour les connexions de l'agent au serveur Zabbix ou au proxy.
 Utilisé pour les vérifications actives.
 Une seule valeur peut être spécifiée : - *unencrypted* - se connecter sans chiffrement (par défaut) ; - *psk* - se connecter en utilisant TLS et une clé prépartagée (PSK) ; - *cert* - se connecter en utilisant TLS et un certificat.

Obligatoire : oui, si des paramètres de certificat TLS ou de PSK sont définis (même pour une connexion *unencrypted*) ; sinon non

TLSCRLFile

Spécifiez le chemin d'accès complet du fichier contenant les certificats révoqués. Ce paramètre est utilisé pour les communications chiffrées entre les composants Zabbix.

TLSKeyFile

Spécifiez le chemin d'accès complet du fichier contenant la clé privée de l'agent, utilisée pour les communications chiffrées entre les composants Zabbix.

TLSPSKFile

Spécifiez le chemin d'accès complet du fichier contenant la clé pré-partagée de l'agent, utilisée pour les communications chiffrées avec le serveur Zabbix ou le proxy.

TLSPSKIdentity

Spécifiez la chaîne d'identité de la clé pré-partagée, utilisée pour les communications chiffrées avec le serveur ou le proxy Zabbix.

TLSServerCertIssuer

Spécifiez l'émetteur du certificat autorisé du serveur (proxy).

TLSServerCertSubject

Spécifiez le sujet de certificat du serveur (proxy) autorisé.

UnsafeUserParameters

Autoriser tous les caractères à être transmis dans les arguments des paramètres définis par l'utilisateur. Les caractères suivants ne sont pas autorisés : \ ' " ` * ? [ ] { } ~ $! & ; ( ) < > | # @ %
De plus, les caractères de nouvelle ligne ne sont pas autorisés.

Par défaut : 0
 Valeurs : 0 - ne pas autoriser, 1 - autoriser

UserParameter

Spécifiez un paramètre défini par l'utilisateur à surveiller. Il peut y avoir plusieurs paramètres définis par l'utilisateur.
Format: `UserParameter=<key>,<shell command>`
Notez que la commande shell ne doit pas renvoyer une chaîne vide ni uniquement un retour de ligne. Les commandes shell peuvent utiliser des chemins relatifs si le paramètre `UserParameterDir` est spécifié.

Exemple:

```
UserParameter=system.test,dir /b | find /c /v ""
UserParameter=check_cpu,.\custom\_script.bat
```

UserParameterDir

Spécifie le chemin de recherche par défaut pour les commandes `UserParameter`. Si cette option est utilisée, l'agent changera son répertoire de travail vers celui indiqué ici avant d'exécuter une commande. Ainsi, les commandes UserParameter peuvent utiliser un préfixe relatif `.\` au lieu d'un chemin complet.
Une seule entrée est autorisée.

Exemple :

```
UserParameterDir=C:\opt\myscripts
```

Paramètres de configuration: Plugins

Voir les paramètres de configuration des plugins de Zabbix agent 2 :

- Plugin Ceph
- Plugin Docker
- Plugin Ember+
- Plugin Memcached
- Plugin Modbus
- Plugin MongoDB
- Plugin MQTT
- Plugin MSSQL
- Plugin MySQL
- Plugin NVIDIA GPU
- Plugin Oracle
- Plugin PostgreSQL
- Plugin Redis
- Plugin S.M.A.R.T.

Principes de configuration des plugins

Cette page décrit les principes de configuration des plugins et les bonnes pratiques.

Tous les plugins sont configurés à l'aide du paramètre `Plugins.*`, qui peut faire partie du fichier de configuration de l'agent 2 ou du fichier de configuration d'un plugin. Si un plugin utilise un fichier de configuration distinct, le chemin vers ce fichier doit être spécifié dans le paramètre `Include` du fichier de configuration de l'agent 2 de Zabbix.

Un paramètre de plugin typique possède la structure suivante :

`Plugins.<PluginName>.<Parameter>=<Value>`

De plus, il existe deux groupes spécifiques de paramètres :

- `Plugins.<PluginName>.Default.<Parameter>=<Value>` est utilisé pour définir les valeurs de paramètres par défaut.
- `Plugins.<PluginName>.<SessionName>.<Parameter>=<Value>` est utilisé pour définir des ensembles distincts de paramètres pour différentes cibles de supervision via des sessions nommées.

Tous les noms de paramètres doivent respecter les exigences suivantes :

- Il est recommandé de mettre une majuscule au début du nom de vos plugins.
- Le paramètre doit commencer par une majuscule.
- Les caractères spéciaux ne sont pas autorisés.
- L'imbrication n'est pas limitée par un niveau maximal.
- Le nombre de paramètres n'est pas limité.

Par exemple, pour effectuer des vérifications actives qui ne possèdent pas d'intervalle de mise à jour *Scheduling* immédiatement après le redémarrage de l'agent, uniquement pour le plugin Uptime, définissez `Plugins.Uptime.System.ForceActiveChecksOnStart=1` dans le fichier de configuration. De même, pour définir une limite personnalisée pour les vérifications simultanées du plugin CPU, définissez `Plugins.CPU.System.Capacity=N` dans le fichier de configuration.

Valeurs par défaut

Vous pouvez définir les valeurs par défaut des paramètres liés à la connexion (URI, nom d'utilisateur, mot de passe, etc.) dans le fichier de configuration au format suivant :

```
Plugins.<PluginName>.Default.<Parameter>=<Value>
```

Par exemple, `Plugins.Mysql.Default.Username=zabbix`, `Plugins.MongoDB.Default.Uri=tcp://127.0.0.1:27017`, etc.

Si aucune valeur pour un tel paramètre n'est fournie dans une clé d'élément ou dans les paramètres de la session nommée, le plugin utilise la valeur par défaut. Si un paramètre par défaut n'est pas défini non plus, les valeurs par défaut codées en dur sont utilisées.

> **Note:**
> Si une clé d'élément ne comporte aucun paramètre, Zabbix agent 2 tente de collecter la métrique à l'aide des valeurs définies dans la section des paramètres par défaut.

Sessions nommées

Les sessions nommées représentent un niveau supplémentaire de paramètres de plugin et peuvent être utilisées pour spécifier des ensembles distincts de paramètres d'authentification pour chacune des instances surveillées. Chaque paramètre de session nommée doit avoir la structure suivante :

```
Plugins.<PluginName>.Sessions.<SessionName>.<Parameter>=<Value>
```

Un nom de session peut être utilisé comme paramètre de clé d'élément connString à la place de la spécification séparée d'une URI, d'un nom d'utilisateur et/ou d'un mot de passe.

Dans les clés d'élément, le premier paramètre peut être soit un connString, soit une URI. Si le premier paramètre de clé ne correspond à aucun nom de session, il sera traité comme une URI. Notez que la transmission d'identifiants URI intégrés dans la clé d'élément n'est pas prise en charge ; utilisez plutôt des paramètres de session nommée.

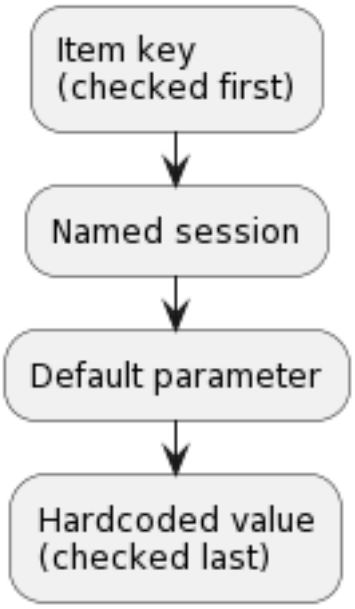
La liste des paramètres de session nommée disponibles dépend du plugin.

Il est possible de remplacer les paramètres de session en spécifiant de nouvelles valeurs dans les paramètres de clé d'élément (voir exemple).

Si un paramètre n'est pas défini pour la session nommée, Zabbix agent 2 utilisera la valeur définie dans le paramètre de plugin par défaut.

Priorité des paramètres

Les plugins de l'agent Zabbix 2 recherchent les valeurs des paramètres liés à la connexion dans l'ordre suivant :



1. Le premier paramètre de clé d'élément est comparé aux noms de session. Si aucune correspondance n'est trouvée, il est traité comme une valeur réelle ; dans ce cas, l'étape 3 sera ignorée. Si une correspondance est trouvée, la valeur du paramètre (généralement une URI) doit être définie dans la session nommée.
2. Les autres paramètres seront pris à partir de la clé d'élément s'ils sont définis.
3. Si un paramètre de clé d'élément (par exemple, password) est vide, le plugin recherchera le paramètre correspondant de la session nommée.
4. Si le paramètre de session n'est pas non plus spécifié, la valeur définie dans le paramètre par défaut correspondant sera utilisée.
5. Si tout le reste échoue, le plugin utilisera la valeur par défaut codée en dur.

Exemple 1

Supervision de deux instances "MySQL1" et "MySQL2".

Paramètres de configuration :

```
Plugins.Mysql.Sessions.MySQL1.Uri=tcp://127.0.0.1:3306
Plugins.Mysql.Sessions.MySQL1.User=mysql1_user
Plugins.Mysql.Sessions.MySQL1.Password=unique_password
Plugins.Mysql.Sessions.MySQL2.Uri=tcp://192.0.2.1:3306
Plugins.Mysql.Sessions.MySQL2.User=mysql2_user
Plugins.Mysql.Sessions.MySQL2.Password=different_password
```

À la suite de cette configuration, chaque nom de session peut être utilisé comme connString dans une clé d'élément, par exemple `mysql.ping[MySQL1]` ou `mysql.ping[MySQL2]`.

Exemple 2

Fourniture de certains paramètres dans la clé d'élément.

Paramètres de configuration :

```
Plugins.PostgreSQL.Sessions.Session1.Uri=tcp://192.0.2.234:5432
Plugins.PostgreSQL.Sessions.Session1.User=old_username
Plugins.PostgreSQL.Sessions.Session1.Password=session_password
```

Clé d'élément : `pgsql.ping[session1,new_username,,postgres]`

À la suite de cette configuration, l'agent se connectera à PostgreSQL en utilisant les paramètres suivants :

- URI à partir du paramètre de session : `192.0.2.234:5432`
- Nom d'utilisateur à partir de la clé d'élément : `new_username`
- Mot de passe à partir du paramètre de session (puisqu'il est omis dans la clé d'élément) : `session_password`
- Nom de la base de données à partir de la clé d'élément : `postgres`

Exemple 3

Collecte d'un élément à l'aide des paramètres de configuration par défaut.

Paramètres de configuration :

```
Plugins.PostgreSQL.Default.Uri=tcp://192.0.2.234:5432
Plugins.PostgreSQL.Default.User=zabbix
Plugins.PostgreSQL.Default.Password=password
```

Clé d'élément: `pgsql.ping[,,,postgres]`

À la suite de cette configuration, l'agent se connectera à PostgreSQL en utilisant les paramètres suivants :

- URI par défaut : `192.0.2.234:5432`
- Nom d'utilisateur par défaut : `zabbix`
- Mot de passe par défaut : `password`
- Nom de la base de données issu de la clé d'élément : `postgres`

Connexions

Certains plugins prennent en charge la collecte de métriques à partir de plusieurs instances simultanément. Les instances locales et distantes peuvent être surveillées. Les connexions TCP et Unix-socket sont prises en charge.

Il est recommandé de configurer les plugins afin de conserver les connexions aux instances dans un état ouvert. Les avantages sont une réduction de la congestion réseau, de la latence, ainsi que de l'utilisation du CPU et de la mémoire grâce au nombre plus faible de connexions. La bibliothèque cliente se charge de cela.

> **Note:**
> La période pendant laquelle les connexions inutilisées doivent rester ouvertes peut être définie par le paramètre `Plugins.<PluginName>.KeepAlive`. Exemple: `Plugins.Memcached.KeepAlive`

Plugin Ceph

Vue d'ensemble

Les paramètres de configuration permettent de personnaliser le comportement du plugin Ceph.

Il est recommandé de les spécifier dans leur propre fichier de configuration (par exemple, `ceph.conf`), plutôt que dans le fichier de configuration de l'agent 2. Utilisez ensuite la directive Include pour inclure ce fichier dans la configuration de Zabbix agent 2.

Tous les paramètres sont **facultatifs**, sauf indication explicite précisant que le paramètre est obligatoire.

Cette page est structurée comme suit :

- Vue d'ensemble des paramètres (pour voir tous les détails d'un paramètre, cliquez sur son nom)
- Détails des paramètres

Parameter	Description
Plugins.Ceph.Default.ApiKey	Spécifie la clé API par défaut pour la connexion à Ceph.
Plugins.Ceph.Default.Mode	Spécifie le mode par défaut pour la connexion à Ceph.
Plugins.Ceph.Default.URI	Spécifie l'URI par défaut pour la connexion à Ceph.
Plugins.Ceph.Default.User	Spécifie le nom d'utilisateur par défaut pour la connexion à Ceph.
Plugins.Ceph.InsecureSkipVerify	Indique si un client HTTP doit vérifier la chaîne de certificats du serveur et le nom d'hôte.
Plugins.Ceph.KeepAlive	Définit le temps d'attente maximal avant la fermeture des connexions de plugin inutilisées.
Plugins.Ceph.Sessions.<SessionName>.ApiKey	Spécifie la clé API de la session nommée pour la connexion à Ceph.
Plugins.Ceph.Sessions.<SessionName>.Mode	Spécifie le mode de la session nommée pour la connexion à Ceph.
Plugins.Ceph.Sessions.<SessionName>.URI	Spécifie l'URI de la session nommée pour la connexion à Ceph.
Plugins.Ceph.Sessions.<SessionName>.User	Spécifie le nom d'utilisateur de la session nommée pour la connexion à Ceph.
Plugins.Ceph.Timeout	Définit le délai d'exécution de la requête.

Notez que :

- Les valeurs par défaut reflètent les valeurs par défaut du processus, et non celles des fichiers de configuration fournis ;
- Les valeurs prennent en charge les variables d'environnement ;
- Zabbix prend en charge les fichiers de configuration uniquement en encodage UTF-8 sans BOM ;
- Les commentaires commençant par "#" ne sont pris en charge qu'au début de la ligne.

Détails du paramètre

Plugins.Ceph.Default.ApiKey

Spécifiez la clé API par défaut pour la connexion à Ceph.
 Utilisée si aucune valeur n'est spécifiée dans une clé d'élément ou une session nommée.

Plugins.Ceph.Default.Mode

Spécifiez le mode par défaut pour la connexion à Ceph.
 Le mode natif n'est pris en charge que sous Linux et à partir de Ceph 16. Le mode restful ne fonctionnera pas avec la version 20 de Ceph ou une version plus récente.
 Notez que l'ensemble des identifiants utilisateur diffère selon le mode et qu'ils ne sont pas compatibles entre eux.

Par défaut: `restful` (obsolète)
 Plage: native/restful

Plugins.Ceph.Default.URI

Spécifie l'URI par défaut pour la connexion à Ceph.
 Utilisé si aucune valeur n'est spécifiée dans une clé d'élément ou une session nommée.

Ne doit pas inclure d'identifiants intégrés (ils seront ignorés).
Doit correspondre au format URI.
Seul le schéma `https` est pris en charge ; le schéma peut être omis.
Le port peut être omis (par défaut=8003).

Par défaut: `https://localhost:8003`

Exemples:

```
https://127.0.0.1:8003
localhost
```

Plugins.Ceph.Default.User

Spécifiez le nom d'utilisateur par défaut pour la connexion à Ceph.
 Utilisé si aucune valeur n'est spécifiée dans une clé d'élément ou une session nommée.

Plugins.Ceph.InsecureSkipVerify

Indique si un client HTTP doit vérifier la chaîne de certificats et le nom d'hôte du serveur.
 Si *true*, TLS accepte n'importe quel certificat présenté par le serveur ainsi que n'importe quel nom d'hôte figurant dans ce certificat. Dans ce mode, TLS est vulnérable aux attaques de type man-in-the-middle (à utiliser uniquement pour les tests).

Valeur par défaut: `false`
 Plage: false/true

Plugins.Ceph.KeepAlive

Définissez le temps d'attente maximal (en secondes) avant la fermeture des connexions de plugin inutilisées.

Par défaut: 300
 Plage: 60-900

Plugins.Ceph.Sessions.<SessionName>.ApiKey

Spécifiez la clé API de la session nommée pour la connexion à Ceph.
 <SessionName> - définissez le nom d'une session à utiliser dans les clés d'élément.

Plugins.Ceph.Sessions.<SessionName>.Mode

Spécifiez le mode de session nommé pour la connexion à Ceph.
 <SessionName> - définissez le nom d'une session à utiliser dans les clés d'élément.
 Le mode natif n'est pris en charge que sous Linux et à partir de Ceph 16. Le mode restful ne fonctionnera pas avec Ceph version 20 ou ultérieure.
 Notez que l'ensemble des identifiants utilisateur diffère selon le mode et qu'ils ne sont pas compatibles entre eux.

Par défaut: `restful` (obsolète)
 Plage: native/restful

Plugins.Ceph.Sessions.<SessionName>.URI

Spécifiez l'URI de session nommée pour la connexion à Ceph.
 <SessionName> - définissez le nom d'une session à utiliser dans les clés d'élément.
Ne doit pas inclure d'identifiants intégrés (ils seront ignorés).
Doit correspondre au format URI.
 Seul le schéma `https` est pris en charge ; le schéma peut être omis.
Un port peut être omis (par défaut=8003).

Par défaut: `https://localhost:8003`

Exemples:

```
https://127.0.0.1:8003
localhost
```

Plugins.Ceph.Sessions.<SessionName>.User

Spécifiez le nom d'utilisateur de la session nommée pour la connexion à Ceph.
 <SessionName> - définissez le nom d'une session à utiliser dans les clés d'élément.

Plugins.Ceph.Timeout

Définit le délai d'exécution de la requête (durée d'attente avant d'interrompre une requête si elle ne se termine pas) en secondes.

Par défaut: `global timeout`
 Plage: 1-30

Voir aussi

- Instructions pour configurer les plugins
- Description des paramètres de configuration généraux de Zabbix agent 2 : Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)

Plugin Docker

Vue d'ensemble

Les paramètres de configuration permettent de personnaliser le comportement du plugin Docker.

Il est recommandé de les spécifier dans leur propre fichier de configuration (par exemple, `docker.conf`), plutôt que dans le fichier de configuration de l'agent 2. Utilisez ensuite la directive Include pour inclure ce fichier dans la configuration de Zabbix agent 2.

Tous les paramètres sont **facultatifs**, sauf indication explicite précisant que le paramètre est obligatoire.

Cette page est structurée comme suit :

- Vue d'ensemble des paramètres (pour voir tous les détails d'un paramètre, cliquez sur son nom)
- Détails des paramètres

Parameter	Description
Plugins.Docker.Endpoint	Spécifiez l'emplacement du socket UNIX du démon Docker..
Plugins.Docker.Timeout	Spécifiez le délai d'exécution de la requête.

Notez que :

- Les valeurs par défaut reflètent les valeurs par défaut du processus, et non celles des fichiers de configuration fournis ;
- Les valeurs prennent en charge les variables d'environnement ;
- Zabbix prend en charge les fichiers de configuration uniquement en encodage UTF-8, sans BOM ;
- Les commentaires commençant par "#" ne sont pris en charge qu'au début de la ligne.

Détails du paramètre

Plugins.Docker.Endpoint

Spécifiez l'emplacement du socket UNIX du démon Docker.
 Doit contenir un schéma (seul `unix://` est pris en charge).

Par défaut: `unix:///var/run/docker.sock`

Plugins.Docker.Timeout

Spécifie le délai d'exécution de la requête (la durée, en secondes, à attendre qu'une requête se termine avant de l'interrompre).

Par défaut: `global timeout<br> Plage: 1-30`

Voir aussi

- Description des paramètres de configuration généraux de Zabbix agent 2 : Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- Instructions pour configurer les plugins

Plugin Ember+

Vue d'ensemble

Les paramètres de configuration permettent de personnaliser le comportement du plugin Ember+.

Il est recommandé de les spécifier dans leur propre fichier de configuration (par exemple, `emberplus.conf`), plutôt que dans le fichier de configuration de l'agent 2. Utilisez ensuite la directive Include pour inclure ce fichier dans la configuration de Zabbix agent 2.

Le plugin Ember+ est un plugin chargeable et est disponible, avec une description complète, dans le dépôt du plugin Ember+.

Ce plugin n'est actuellement disponible qu'en compilation à partir des sources (pour Unix et Windows).

Tous les paramètres sont **facultatifs**, sauf indication explicite précisant que le paramètre est obligatoire.

Cette page est structurée comme suit :

- Vue d'ensemble des paramètres (pour afficher les détails complets d'un paramètre, cliquez sur son nom)
- Détails des paramètres

Parameter	Description
Plugins.EmberPlus.Default.Uri	Spécifie l'URI par défaut pour la connexion à Ember+.
Plugins.EmberPlus.KeepAlive	Définit le temps d'attente maximal avant la fermeture des connexions de plugin inutilisées.
Plugins.EmberPlus.Sessions.<SessionName>.Uri	Spécifie l'URI de la session nommée pour la connexion à Ember+.
Plugins.EmberPlus.System.Path	Spécifie le chemin vers l'exécutable du plugin Ember+.
Plugins.EmberPlus.Timeout	Définit le délai d'expiration d'exécution de la requête.

Notez que :

- Les valeurs par défaut reflètent les valeurs par défaut du processus, et non celles des fichiers de configuration fournis.
- Les valeurs prennent en charge les variables d'environnement ;
- Zabbix prend en charge les fichiers de configuration uniquement en encodage UTF-8 sans BOM.
- Les commentaires commençant par "#" ne sont pris en charge qu'au début de la ligne.

Détails du paramètre

Plugins.EmberPlus.Default.Uri

Spécifiez l'URI par défaut pour la connexion à Ember+.
 Le seul schéma pris en charge est `tcp://`. Un schéma peut être omis. Les identifiants intégrés seront ignorés.

Par défaut: `tcp://localhost:9998`

Plugins.EmberPlus.KeepAlive

Définissez le temps d'attente maximal (en secondes) avant la fermeture des connexions de plugin inutilisées.

Par défaut: 300
 Plage: 60-900

Plugins.EmberPlus.Sessions.<SessionName>.Uri

Spécifiez l'URI de la session nommée pour la connexion à Ember+.
 Le seul schéma pris en charge est `tcp://`. Un schéma peut être omis. Les identifiants intégrés seront ignorés.
 <SessionName> - définit le nom d'une session à utiliser dans les clés d'élément.

Par défaut: `tcp://localhost:9998`

Plugins.EmberPlus.System.Path

Spécifiez le chemin vers l'exécutable du plugin Ember+.

Exemple : `Plugins.EmberPlus.System.Path=/usr/libexec/zabbix/zabbix-agent2-plugin-ember-plus`

Plugins.EmberPlus.Timeout

Définissez la durée, en secondes, d'attente d'une réponse du serveur lors de la première connexion et des opérations suivantes de la session.

Par défaut: `global timeout<br>` Plage: 1–30

Options

Parameter	Description
-V --version	Afficher la version du plugin et les informations de licence.
-h --help	Afficher les informations d'aide (raccourci).
-t, --test <item key>	Lancer le plugin pour les tests (la configuration du plugin est ignorée).

Voir aussi :

- Description des paramètres de configuration généraux de Zabbix agent 2 : Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- Instructions pour configurer les plugins

Plugin GPU NVIDIA

Vue d'ensemble

Les paramètres de configuration permettent de personnaliser le comportement du plugin NVIDIA GPU.

Il est recommandé de les spécifier dans leur propre fichier de configuration (par exemple, `nvidiagpu.conf`), plutôt que dans le fichier de configuration de l'agent 2. Utilisez ensuite la directive Include pour inclure ce fichier dans la configuration de Zabbix agent 2.

Le plugin NVIDIA GPU est un plugin chargeable et est disponible, avec une description complète, dans le dépôt du plugin NVIDIA GPU.

Tous les paramètres sont **facultatifs**, sauf indication explicite précisant qu'un paramètre est obligatoire.

Cette page est structurée comme suit :

- Vue d'ensemble des paramètres (pour voir les détails complets d'un paramètre, cliquez sur son nom)
- Détails des paramètres

Parameter	Description
Plugins.NVIDIA.System.Path	Spécifie le chemin vers l'exécutable du plugin NVIDIA GPU.
Plugins.NVIDIA.Timeout	Spécifie le délai maximal (en secondes) à attendre une réponse du serveur lors des tentatives de connexion et des opérations ultérieures dans la session.

Notez que :

- Les valeurs par défaut reflètent les valeurs par défaut du processus, et non celles des fichiers de configuration fournis ;
- Les valeurs prennent en charge les variables d'environnement ;
- Zabbix prend en charge les fichiers de configuration uniquement en encodage UTF-8 sans BOM ;
- Les commentaires commençant par "#" ne sont pris en charge qu'au début de la ligne.

Détails du paramètre

Plugins.NVIDIA.System.Path

Spécifiez le chemin vers l'exécutable du plugin GPU NVIDIA.

Exemple :

```
Plugins.NVIDIA.System.Path=/path/to/executable/nvidia
```

Plugins.NVIDIA.Timeout

Spécifie le délai maximal (en secondes) d'attente d'une réponse du serveur lors des tentatives de connexion et des opérations ultérieures dans la session.
 Le délai d'attente global du type d'élément ou le délai d'attente individuel de l'élément remplacera cette valeur s'il est supérieur.

Par défaut: `global timeout<br>` Plage: 1–30

Voir aussi

- Description des paramètres de configuration généraux de Zabbix agent 2 : Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- Instructions pour configurer les plugins

Plugin Memcached

Vue d'ensemble

Les paramètres de configuration permettent de personnaliser le comportement du plugin Memcached.

Il est recommandé de les spécifier dans leur propre fichier de configuration (par exemple, `memcached.conf`), plutôt que dans le fichier de configuration de l'agent 2. Utilisez ensuite la directive Include pour inclure ce fichier dans la configuration de Zabbix agent 2.

Tous les paramètres sont **facultatifs**, sauf indication explicite précisant que le paramètre est obligatoire.

Cette page est structurée comme suit :

- Vue d'ensemble des paramètres (pour voir tous les détails d'un paramètre, cliquez sur son nom)
- Détails des paramètres

Parameter	Description
Plugins.Memcached.Default.Password	Spécifiez le mot de passe par défaut pour la connexion à Memcached.
Plugins.Memcached.Default.Uri	Spécifiez l'URI par défaut pour la connexion à Memcached.
Plugins.Memcached.Default.User	Spécifiez le nom d'utilisateur par défaut pour la connexion à Memcached.
Plugins.Memcached.KeepAlive	Définissez le temps d'attente maximal avant la fermeture des connexions de plugin inutilisées.
Plugins.Memcached.Sessions.*.Password	Spécifiez le mot de passe de la session nommée pour la connexion à Memcached.
Plugins.Memcached.Sessions.*.Uri	Spécifiez l'URI de la session nommée pour la connexion à Memcached.
Plugins.Memcached.Sessions.*.User	Spécifiez le nom d'utilisateur de la session nommée pour la connexion à Memcached.
Plugins.Memcached.Timeout	Définissez le délai d'expiration d'exécution de la requête.

Notez que :

- Les valeurs par défaut reflètent les valeurs par défaut du processus, et non celles des fichiers de configuration fournis ;
- Les valeurs prennent en charge les variables d'environnement ;
- Zabbix prend en charge les fichiers de configuration uniquement au format UTF-8 sans BOM ;
- Les commentaires commençant par "#" ne sont pris en charge qu'au début de la ligne.

Détails du paramètre

Plugins.Memcached.Default.Password

Spécifiez le mot de passe par défaut pour la connexion à Memcached.
 Utilisé si aucune valeur n'est spécifiée dans une clé d'élément ou une session nommée.

Plugins.Memcached.Default.Uri

Spécifie l'URI par défaut pour la connexion à Memcached.
 Utilisé si aucune valeur n'est spécifiée dans une clé d'élément ou une session nommée.
 Les identifiants intégrés seront ignorés.
 Doit correspondre au format URI. Schémas pris en charge: `tcp`, `unix`.
 Un schéma peut être omis.
 Un port peut être omis (par défaut=11211).

Par défaut: `tcp://localhost:11211`

Exemples:

```
tcp://localhost:11211
localhost
unix:/var/run/memcached.sock
```

Plugins.Memcached.Default.User

Spécifiez le nom d'utilisateur par défaut pour la connexion à Memcached.
 Utilisé si aucune valeur n'est spécifiée dans une clé d'élément ou une session nommée.

Plugins.Memcached.KeepAlive

Définit le temps d'attente maximal (en secondes) avant la fermeture des connexions de plugin inutilisées.

Par défaut: 300
 Plage: 60–900

Plugins.Memcached.Sessions.<SessionName>.Password

Spécifiez le mot de passe de la session nommée pour la connexion à Memcached.
 <SessionName> - définissez le nom d'une session à utiliser dans les clés d'éléments.

Plugins.Memcached.Sessions.<SessionName>.Uri

Spécifiez l'URI de session nommé pour la connexion à Memcached.
 <SessionName> - définissez le nom d'une session à utiliser dans les clés d'élément.
 Les identifiants intégrés seront ignorés.
 Doit correspondre au format URI. Schémas pris en charge: `tcp`, `unix`.
 Un schéma peut être omis.
 Un port peut être omis (par défaut=11211).

Exemples:

```
tcp://localhost:11211
localhost
unix:/var/run/memcached.sock
```

Plugins.Memcached.Sessions.<SessionName>.User

Spécifiez le nom d'utilisateur de la session nommée pour la connexion à Memcached.
 <SessionName> - définit le nom d'une session à utiliser dans les clés d'élément.

Plugins.Memcached.Timeout

Définissez le délai d'exécution de la requête (la durée, en secondes, d'attente de l'achèvement d'une requête avant de l'interrompre).

Par défaut: `global timeout`
 Plage: 1-30

Voir aussi :

- Description des paramètres de configuration généraux de Zabbix agent 2 : Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- Instructions pour configurer les plugins

Plugin Modbus

Vue d'ensemble

Les paramètres de configuration permettent de personnaliser le comportement du plugin Modbus.

Il est recommandé de les spécifier dans leur propre fichier de configuration (par exemple, `modbus.conf`), plutôt que dans le fichier de configuration de l'agent 2. Utilisez ensuite la directive Include pour inclure ce fichier dans la configuration de Zabbix agent 2.

Tous les paramètres sont **facultatifs**, sauf indication explicite précisant que le paramètre est obligatoire.

Cette page est structurée comme suit :

- Vue d'ensemble des paramètres (pour voir les détails complets d'un paramètre, cliquez sur son nom)
- Détails des paramètres

Parameter	Description
Plugins.Modbus.Sessions.<SessionName>.Endpoint	Spécifie le point de terminaison de session nommé pour la connexion à Modbus.
Plugins.Modbus.Sessions.<SessionName>.SlaveID	Spécifie l'ID esclave de la session nommée pour la connexion à Modbus.
Plugins.Modbus.Sessions.<SessionName>.Timeout	Définit le délai d'attente de la session nommée en secondes.

Notez que :

- Les valeurs par défaut reflètent les valeurs par défaut du processus, et non celles des fichiers de configuration fournis ;
- Les valeurs prennent en charge les variables d'environnement ;
- Zabbix prend en charge les fichiers de configuration uniquement en encodage UTF-8, sans BOM ;
- Les commentaires commençant par "#" ne sont pris en charge qu'au début de la ligne.

Détails du paramètre

Plugins.Modbus.Sessions.<SessionName>.Endpoint

Spécifiez le point de terminaison de session nommé pour la connexion à Modbus.
 La chaîne de connexion se compose d'un schéma de protocole, d'une adresse d'hôte et d'un port ou du nom d'un port série ainsi que de ses attributs.
 <SessionName> - définit le nom d'une session à utiliser dans les clés d'élément.

Plugins.Modbus.Sessions.<SessionName>.SlaveID

Spécifiez l'ID esclave de la session nommée pour la connexion à Modbus.
 <SessionName> - définit le nom d'une session à utiliser dans les clés d'élément.

Exemple : `Plugins.Modbus.Sessions.MB1.SlaveID=20`

Note que ce paramètre de session nommée n'est vérifié que si la valeur fournie dans le paramètre d'ID esclave de la clé d'élément est vide.

Plugins.Modbus.Sessions.<SessionName>.Timeout

Définissez le délai d'expiration de la session nommée en secondes.
 <SessionName> - définit le nom d'une session à utiliser dans les clés d'élément.

Exemple : `Plugins.Modbus.Sessions.MB1.Timeout=2`

Si vous devez définir le délai d'expiration d'exécution de la requête (durée d'attente avant d'interrompre une requête pour qu'elle se termine), utilisez le formulaire de configuration de l'élément.

Voir aussi

- Description des paramètres de configuration généraux de Zabbix agent 2 : Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- Instructions pour configurer les plugins

Plugin MongoDB

Vue d'ensemble

Les paramètres de configuration permettent de personnaliser le comportement du plugin MongoDB.

Il est recommandé de les spécifier dans leur propre fichier de configuration (par exemple, `mongodb.conf`), plutôt que dans le fichier de configuration de l'agent 2. Utilisez ensuite la directive Include pour inclure ce fichier dans la configuration de Zabbix agent 2.

Le plugin MongoDB est un plugin chargeable et est disponible, avec une description complète, dans le dépôt du plugin MongoDB.

Tous les paramètres sont **facultatifs**, sauf indication explicite précisant que le paramètre est obligatoire.

Lors de la connexion à MongoDB, le plugin vérifie sa configuration dans un ordre spécifique, qui détermine quelle valeur utiliser. L'ordre, de la priorité la plus élevée à la plus faible, est le suivant :

1. Paramètres spécifiés dans l'URI de connexion (par exemple, les paramètres de requête définis après ? dans `Plugins.MongoDB.Sessio` ont la priorité la plus élevée).
2. Paramètres dans la clé d'élément de Zabbix agent 2 (par exemple, mongodb.collection.stats).
3. Paramètres définis dans une session nommée (par exemple, `Plugins.MongoDB.Sessions.<SessionName>.User`, `Plugins.MongoDB.Sessions.<SessionName>.Password`).
4. Valeurs par défaut (par exemple, `Plugins.MongoDB.Default.User`, `Plugins.MongoDB.Default.Password`).

Par exemple :

- Si vous spécifiez la base de données pour l'authentification d'une connexion au serveur MongoDB dans le paramètre `*.<SessionName>.Uri` (par exemple, `Plugins.MongoDB.Sessions.<session_name>.Uri=mongodb://user:password@127` le plugin considérera que le paramètre `*.<SessionName>.AuthSource` est défini sur `admin`, même si le paramètre `*.<SessionName>.AuthSource` est défini sur une autre valeur.
- Si vous spécifiez l'URI dans le paramètre `*.<SessionName>.Uri` (par exemple, `Plugins.MongoDB.Sessions.<session_name>.U` mais fournissez le nom d'utilisateur et le mot de passe dans la clé d'élément de Zabbix agent 2, le plugin utilisera le nom d'utilisateur et le mot de passe de la clé d'élément, même si `Plugins.MongoDB.Sessions.<SessionName>.User` et `Plugins.MongoDB.Sessions.<SessionName>.Password` sont configurés.

Attention:
Si le schéma `mongodb+srv://` est utilisé, l'URI peut récupérer des paramètres à partir d'un enregistrement DNS TXT. Les paramètres obtenus à partir de cet enregistrement TXT remplacent toutes les autres configurations, y compris les paramètres de l'URI, les paramètres de la clé d'élément de Zabbix Agent 2 et les paramètres de session.
 Un seul enregistrement TXT peut exister pour le nom d'hôte utilisé dans la chaîne de connexion.

Cette page est structurée comme suit :

- Vue d'ensemble des paramètres (pour voir tous les détails d'un paramètre, cliquez sur son nom)
- Détails des paramètres

Parameter	Description
Plugins.MongoDB.Default.AuthMechanism	Spécifiez le mécanisme d'authentification par défaut pour la connexion au serveur MongoDB.
Plugins.MongoDB.Default.AuthSource	Spécifiez la base de données par défaut pour l'authentification de la connexion au serveur MongoDB.
Plugins.MongoDB.Default.Dialect	Désactivez la découverte automatique des nœuds de l'ensemble de réplicas MongoDB.
Plugins.MongoDB.Default.Password	Spécifiez le mot de passe par défaut.
Plugins.MongoDB.Default.ReplicaSet	Spécifiez le nom par défaut de l'ensemble de réplicas, utilisé pour la découverte automatique des nœuds de l'ensemble de réplicas MongoDB.
Plugins.MongoDB.Default.ReadPreference	Spécifiez l'ordre par défaut pour la connexion aux nœuds de l'ensemble de réplicas MongoDB ou le repli.
Plugins.MongoDB.Default.TLSConnect	Spécifiez le niveau de chiffrement requis pour les communications entre Zabbix agent 2 et les bases de données surveillées.
Plugins.MongoDB.Default.TLSCAFile	Spécifiez le chemin complet du fichier contenant les certificats des AC racines pour la vérification du certificat du pair.
Plugins.MongoDB.Default.TLSCertFile	Spécifiez le chemin complet du fichier contenant le certificat de l'agent ou la chaîne de certificats.
Plugins.MongoDB.Default.TLSKeyFile	Spécifiez le chemin complet du fichier contenant la clé privée de la base de données.
Plugins.MongoDB.Default.Uri	Spécifiez l'URI par défaut.
Plugins.MongoDB.Default.User	Spécifiez le nom d'utilisateur par défaut.
Plugins.MongoDB.KeepAlive	Définissez le temps d'attente maximal avant la fermeture des connexions inutilisées du plugin.
Plugins.MongoDB.Sessions.<SessionName>.AuthMechanism	Spécifiez le mécanisme d'authentification de la session nommée pour la connexion au serveur MongoDB.
Plugins.MongoDB.Sessions.<SessionName>.AuthSource	Spécifiez la base de données de la session nommée pour l'authentification de la connexion au serveur MongoDB.
Plugins.MongoDB.Sessions.<SessionName>.Dialect	Désactivez la découverte automatique des nœuds de l'ensemble de réplicas MongoDB pour la session nommée.
Plugins.MongoDB.Sessions.<SessionName>.Password	Spécifiez le mot de passe de la session nommée.
Plugins.MongoDB.Sessions.<SessionName>.ReplicaSet	Spécifiez le nom de l'ensemble de réplicas de la session nommée, utilisé pour la découverte automatique des nœuds de l'ensemble de réplicas MongoDB.
Plugins.MongoDB.Sessions.<SessionName>.ReadPreference	Spécifiez l'ordre de la session nommée pour la connexion aux nœuds de l'ensemble de réplicas MongoDB ou le repli.
Plugins.MongoDB.Sessions.<SessionName>.TLSConnect	Spécifiez le niveau de chiffrement requis pour les communications entre Zabbix agent 2 et les bases de données surveillées pour la session nommée.
Plugins.MongoDB.Sessions.<SessionName>.TLSCAFile	Spécifiez le chemin complet du fichier contenant les certificats des AC racines pour la vérification du certificat du pair pour la session nommée.
Plugins.MongoDB.Sessions.<SessionName>.TLSCertFile	Spécifiez le chemin complet du fichier contenant le certificat de l'agent ou la chaîne de certificats pour la session nommée.
Plugins.MongoDB.Sessions.<SessionName>.TLSKeyFile	Spécifiez le chemin complet du fichier contenant la clé privée de la base de données pour la session nommée.
Plugins.MongoDB.Sessions.<SessionName>.Uri	Spécifiez l'URI de la session nommée.
Plugins.MongoDB.Sessions.<SessionName>.User	Spécifiez le nom d'utilisateur de la session nommée.
Plugins.MongoDB.System.Path	Spécifiez le chemin vers l'exécutable du plugin.
Plugins.MongoDB.Timeout	Définissez le délai d'attente d'exécution de la requête.

Notez que :

- Les valeurs par défaut reflètent les valeurs par défaut du processus, et non celles des fichiers de configuration fournis.
- Les valeurs prennent en charge les variables d'environnement.
- Zabbix prend en charge les fichiers de configuration uniquement en encodage UTF-8 sans BOM.
- Les commentaires commençant par "#" ne sont pris en charge qu'au début de la ligne.

Détails du paramètre

Plugins.MongoDB.Default.AuthMechanism

Spécifiez le mécanisme d'authentification par défaut pour la connexion au serveur MongoDB.
 <SessionName> - définit le nom d'une session à utiliser dans les clés d'élément.

Plage: `SCRAM-SHA-1`, `SCRAM-SHA-256` ou `MONGODB-X509`

Plugins.MongoDB.Default.AuthSource

Spécifiez la base de données par défaut pour l'authentification de la connexion au serveur MongoDB.
 <SessionName> - définit le nom d'une session à utiliser dans les clés d'élément.

Par défaut: `admin`

Si le mécanisme d'authentification MONGODB-X509 est utilisé, ce paramètre est automatiquement défini sur $external; le modifier à toute autre valeur provoquera une erreur au lancement.

Plugins.MongoDB.Default.Direct

Désactiver la découverte automatique des nœuds de l'ensemble de réplicas MongoDB.
 <SessionName> - définit le nom d'une session à utiliser dans les clés d'élément.

Valeur par défaut: false
 Plage: true - se connecter uniquement au serveur spécifié, en ignorant les autres membres de l'ensemble de réplicas, false - découvrir automatiquement tous les nœuds de l'ensemble de réplicas.

Si le schéma tcp:// (obsolète) est utilisé, le paramètre est défini sur true par défaut.

Plugins.MongoDB.Default.Password

Spécifiez le mot de passe par défaut.
 <SessionName> - définissez le nom d'une session à utiliser dans les clés d'élément.

Plugins.MongoDB.Default.ReplicaSet

Spécifiez le nom par défaut du jeu de réplicas, utilisé pour la découverte automatique des nœuds du jeu de réplicas MongoDB.
 <SessionName> - définir le nom d'une session à utiliser dans les clés d'élément.
 Si ce paramètre est utilisé avec le schéma mongodb+srv://, il se comportera comme si plusieurs URL avaient été fournies ; un serveur DNS peut avoir un enregistrement TXT qui peut remplacer ce paramètre (ou tout autre paramètre).
 Indisponible si le paramètre Plugins.MongoDB.Sessions.<SessionName>.Direct est défini sur true ; l'agent ne parviendra pas à démarrer.

Plugins.MongoDB.Default.ReadPreference

Spécifiez l'ordre par défaut pour la connexion aux nœuds de l'ensemble de réplication MongoDB ou le repli.
 <SessionName> - définissez le nom d'une session à utiliser dans les clés des éléments.
 Indisponible si le paramètre Plugins.MongoDB.Sessions.<SessionName>.Direct est défini sur true ; l'agent ne démarrera pas.

Plage : primary, primaryPreferred, secondary, secondaryPreferred, nearest, PrimaryMode, PrimaryPreferredMode, SecondaryMode, SecondaryPreferredMode ou NearestMode

Les valeurs primary, primaryPreferred, secondary, secondaryPreferred et nearest sont recommandées. Les valeurs se terminant par Mode sont prises en charge comme alias de compatibilité hérités.

Plugins.MongoDB.Default.TLSConnect

Spécifiez le niveau de chiffrement requis pour les communications entre Zabbix agent 2 et les bases de données surveillées.
 <SessionName> - définit le nom d'une session à utiliser dans les clés d'élément.
 Pris en charge depuis la version 1.2.1 du plugin.

Plage : required - exige une connexion TLS, verify_ca - vérifie les certificats, verify_full - vérifie les certificats et l'adresse IP

Plugins.MongoDB.Default.TLSCAFile

Spécifiez le chemin complet du fichier contenant le ou les certificats CA de niveau supérieur pour la vérification du certificat du pair, utilisés pour les communications chiffrées entre l'agent 2 Zabbix et les bases de données surveillées.
 <SessionName> - définissez le nom d'une session à utiliser dans les clés d'élément.

Obligatoire: yes, si Plugins.MongoDB.Sessions.<SessionName>.TLSConnect est défini sur verify_ca ou verify_full

Plugins.MongoDB.Default.TLSCertFile

Spécifiez le chemin complet du fichier contenant le certificat de l'agent ou la chaîne de certificats, utilisé pour les communications chiffrées entre Zabbix agent 2 et les bases de données surveillées.
 <SessionName> - définit le nom d'une session à utiliser dans les clés d'élément.

Obligatoire: yes, si Plugins.MongoDB.Sessions.<SessionName>.TLSKeyFile est spécifié

Plugins.MongoDB.Default.TLSKeyFile

Spécifiez le chemin d'accès complet du fichier contenant la clé privée de la base de données, utilisée pour les communications chiffrées entre l'agent 2 Zabbix et les bases de données surveillées.
 <SessionName> - définissez le nom d'une session à utiliser dans les clés d'élément.

Obligatoire: yes, si Plugins.MongoDB.Sessions.<SessionName>.TLSCertFile est spécifié

Plugins.MongoDB.Default.Uri

Spécifiez l'URI par défaut.
 <SessionName> - définit le nom d'une session à utiliser dans les clés d'élément.
 Doit correspondre au format URI.
 Schémas pris en charge: mongodb:// et mongodb+srv://; le schéma tcp:// est

obsolète et conservé pour la compatibilité ascendante avec les configurations existantes.
 Un port peut être omis (par défaut=27017).
Exemples: “

Par défaut: `tcp://127.0.0.1:27017`

Exemples:

```
mongodb://127.0.0.1:27017
mongodb+srv://example.com
localhost
```

Plugins.MongoDB.Default.User

Spécifiez le nom d’utilisateur par défaut.
 <SessionName> - définit le nom d’une session à utiliser dans les clés d’élément.

Plugins.MongoDB.KeepAlive

Définit le temps d’attente maximal (en secondes) avant la fermeture des connexions de plugin inutilisées.

Par défaut: 300
 Plage: 60–900

Plugins.MongoDB.Sessions.<SessionName>.AuthMechanism

Spécifiez le mécanisme d’authentification de la session nommée pour la connexion au serveur MongoDB.
 <SessionName> - définit le nom d’une session à utiliser dans les clés d’élément.

Plage: `SCRAM-SHA-1`, `SCRAM-SHA-256` ou `MONGODB-X509`

Plugins.MongoDB.Sessions.<SessionName>.AuthSource

Spécifiez la base de données de session nommée pour l’authentification de la connexion au serveur MongoDB.
 <SessionName> - définit le nom d’une session à utiliser dans les clés d’élément.

Par défaut : `admin`

Si le mécanisme d’authentification `MONGODB-X509` est utilisé, ce paramètre est automatiquement défini sur `$external` ; le modifier à toute autre valeur provoquera une erreur au lancement.

Plugins.MongoDB.Sessions.<SessionName>.Direct

Désactiver la découverte automatique des nœuds du jeu de réplicas MongoDB pour la session nommée.
 <SessionName> - définir le nom d’une session à utiliser dans les clés d’élément.

Par défaut: `false`
 Plage: `true` - se connecter uniquement au serveur spécifié, en ignorant les autres membres du jeu de réplicas, `false` - découvrir automatiquement tous les nœuds du jeu de réplicas.

Si le schéma `tcp://` (obsolète) est utilisé, le paramètre est défini sur `true` par défaut.

Plugins.MongoDB.Sessions.<SessionName>.Password

Spécifie le mot de passe de la session nommée.
 <SessionName> - définit le nom d’une session à utiliser dans les clés d’élément.

Plugins.MongoDB.Sessions.<SessionName>.ReplicaSet

Spécifie le nom de l’ensemble de réplicas de la session nommée, utilisé pour la découverte automatique des nœuds de l’ensemble de réplicas MongoDB.
 <SessionName> - définit le nom d’une session à utiliser dans les clés d’élément.
 Si ce paramètre est utilisé avec le schéma `mongodb+srv://`, il se comportera comme si plusieurs URL avaient été fournies ; un serveur DNS peut avoir un enregistrement TXT qui peut remplacer ce paramètre (ou tout autre paramètre).
 Indisponible si le paramètre `Plugins.MongoDB.Sessions.<SessionName>.Direct` est défini sur `true` ; l’agent ne démarrera pas.

Plugins.MongoDB.Sessions.<SessionName>.ReadPreference

Spécifiez l’ordre de session nommé pour la connexion aux nœuds de l’ensemble de réplication MongoDB ou le basculement.
 <SessionName> - définissez le nom d’une session à utiliser dans les clés d’éléments.
 Indisponible si le paramètre `Plugins.MongoDB.Sessions.<SessionName>.Direct` est défini sur `true` ; l’agent ne démarrera pas.

Plage : `primary`, `primaryPreferred`, `secondary`, `secondaryPreferred`, `nearest`, `PrimaryMode`, `PrimaryPreferredMode`, `SecondaryMode`, `SecondaryPreferredMode` ou `NearestMode`

Les valeurs `primary`, `primaryPreferred`, `secondary`, `secondaryPreferred` et `nearest` sont recommandées. Les valeurs se terminant par `Mode` sont prises en charge comme alias de compatibilité hérités.

Plugins.MongoDB.Sessions.<SessionName>.TLSConnect

Spécifie le niveau de chiffrement requis pour les communications entre Zabbix agent 2 et les bases de données surveillées pour la session nommée.
 <SessionName> - définit le nom d'une session à utiliser dans les clés d'élément.
 Pris en charge depuis la version 1.2.1 du plugin.

Plage: `required` - exige une connexion TLS, `verify_ca` - vérifie les certificats, `verify_full` - vérifie les certificats et l'adresse IP

Plugins.MongoDB.Sessions.<SessionName>.TLSCAFile

Spécifiez le chemin complet du fichier contenant le ou les certificats CA de niveau supérieur pour la vérification du certificat du pair, utilisés pour les communications chiffrées entre l'agent 2 Zabbix et les bases de données surveillées pour la session nommée.
 <SessionName> - définissez le nom d'une session à utiliser dans les clés d'élément.

Obligatoire: `yes`, si `Plugins.MongoDB.Sessions.<SessionName>.TLSConnect` est défini sur `verify_ca` ou `verify_full`

Plugins.MongoDB.Sessions.<SessionName>.TLSCertFile

Spécifiez le chemin complet du fichier contenant le certificat de l'agent ou la chaîne de certificats, utilisé pour les communications chiffrées entre Zabbix agent 2 et les bases de données surveillées pour la session nommée.
 <SessionName> - définit le nom d'une session à utiliser dans les clés d'élément.

Obligatoire: `yes`, si `Plugins.MongoDB.Sessions.<SessionName>.TLSKeyFile` est spécifié

Plugins.MongoDB.Sessions.<SessionName>.TLSKeyFile

Spécifiez le chemin complet du fichier contenant la clé privée de la base de données, utilisée pour les communications chiffrées entre l'agent 2 Zabbix et les bases de données surveillées pour la session nommée.
 <SessionName> - définit le nom d'une session à utiliser dans les clés d'élément.

Obligatoire: `yes`, si `Plugins.MongoDB.Sessions.<SessionName>.TLSCertFile` est spécifié

Plugins.MongoDB.Sessions.<SessionName>.Uri

Spécifie l'URI de session nommée.
 <SessionName> - définit le nom d'une session à utiliser dans les clés d'élément.
 Doit correspondre au format URI.
 Schémas pris en charge : `mongodb://` et `mongodb+srv://` ; le schéma `tcp://` est obsolète et conservé pour la compatibilité ascendante avec les configurations existantes.
 Un port peut être omis (par défaut=27017).
Exemples : "

Par défaut : `tcp://127.0.0.1:27017`

Exemples :

```
mongodb://127.0.0.1:27017
mongodb+srv://example.com
localhost
```

Plugins.MongoDB.Sessions.<SessionName>.User

Spécifie le nom d'utilisateur de la session nommée.
 <SessionName> - définit le nom d'une session à utiliser dans les clés d'élément.

Plugins.MongoDB.System.Path

Spécifiez le chemin vers l'exécutable du plugin MongoDB.

Exemple d'utilisation : `Plugins.MongoDB.System.Path=/usr/libexec/zabbix/zabbix-agent2-plugin-mongodb`

Plugins.MongoDB.Timeout

Définit le délai d'exécution de la requête (la durée, en secondes, d'attente de l'exécution d'une requête avant de l'interrompre).

Par défaut: `global timeout`
 Plage: 1-30

Options

Parameter	Description
-V --version	Afficher la version du plugin et les informations de licence.
-h --help	Afficher les informations d'aide (raccourci).
-t, --test <item key>	Lancer le plugin pour les tests (la configuration du plugin est ignorée).

Voir aussi

- Description des paramètres de configuration généraux de Zabbix agent 2 : Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)

- Instructions pour configurer les plugins

Plugin MQTT

Vue d'ensemble

Les paramètres de configuration permettent de personnaliser le comportement du plugin MQTT.

Il est recommandé de les spécifier dans leur propre fichier de configuration (par exemple, `mqtt.conf`), plutôt que dans le fichier de configuration de l'agent 2. Utilisez ensuite la directive Include pour inclure ce fichier dans la configuration de Zabbix agent 2.

Tous les paramètres sont **facultatifs**, sauf indication explicite précisant que le paramètre est obligatoire.

Cette page est structurée comme suit :

- Vue d'ensemble des paramètres (pour voir les détails complets d'un paramètre, cliquez sur son nom)
- Détails des paramètres

Parameter	Description
Plugins.MQTT.Default.Password	Spécifie le mot de passe par défaut pour la connexion à MQTT.
Plugins.MQTT.Default.TLSCAFile	Spécifie le chemin complet du fichier contenant le ou les certificats CA de niveau supérieur pour la vérification du certificat du pair.
Plugins.MQTT.Default.TLSCertFile	Spécifie le chemin complet du fichier contenant le certificat de l'agent ou la chaîne de certificats.
Plugins.MQTT.Default.TLSKeyFile	Spécifie le chemin complet du fichier contenant la clé privée MQTT.
Plugins.MQTT.Default.Topic	Spécifie le topic par défaut pour l'abonnement MQTT.
Plugins.MQTT.Default.Url	Spécifie la chaîne de connexion par défaut au broker MQTT.
Plugins.MQTT.Default.User	Spécifie le nom d'utilisateur par défaut pour la connexion à MQTT.
Plugins.MQTT.Sessions.<Session name>.Password	Spécifie le mot de passe de la session nommée pour la connexion à MQTT.
Plugins.MQTT.Sessions.<Session name>.TLSCAFile	Spécifie le chemin complet du fichier contenant le ou les certificats CA de niveau supérieur pour la vérification du certificat du pair pour la session nommée.
Plugins.MQTT.Sessions.<Session name>.TLSCertFile	Spécifie le chemin complet du fichier contenant le certificat de l'agent ou la chaîne de certificats pour la session nommée.
Plugins.MQTT.Sessions.<Session name>.TLSKeyFile	Spécifie le chemin complet du fichier contenant la clé privée MQTT pour la session nommée.
Plugins.MQTT.Sessions.<Session name>.Topic	Spécifie le topic de la session nommée pour l'abonnement MQTT.
Plugins.MQTT.Sessions.<Session name>.Url	Spécifie la chaîne de connexion au broker MQTT pour la session nommée.
Plugins.MQTT.Sessions.<Session name>.User	Spécifie le nom d'utilisateur de la session nommée pour la connexion à MQTT.

Notez que :

- Les valeurs par défaut reflètent les valeurs par défaut du processus, et non celles des fichiers de configuration fournis ;
- Les valeurs prennent en charge les variables d'environnement ;
- Zabbix prend en charge les fichiers de configuration uniquement en encodage UTF-8 sans BOM ;
- Les commentaires commençant par "#" ne sont pris en charge qu'au début de la ligne.

Détails du paramètre

Plugins.MQTT.Default.Password

Spécifiez le mot de passe par défaut pour la connexion à MQTT.
 Utilisé si aucune valeur n'est spécifiée dans une clé d'élément ou une session nommée.

Plugins.MQTT.Default.TLSCAFile

Spécifiez le chemin complet du fichier contenant le ou les certificats d'autorité de certification racine pour la vérification du certificat du pair lors des communications chiffrées entre l'agent 2 Zabbix et le broker MQTT.
 Utilisé si aucune valeur n'est spécifiée dans une session nommée.

Plugins.MQTT.Default.TLSCertFile

Spécifiez le chemin complet du fichier contenant le certificat de l'agent ou la chaîne de certificats pour les communications chiffrées entre l'agent 2 Zabbix et le broker MQTT.
 Utilisé si aucune valeur n'est spécifiée dans une session nommée.

Plugins.MQTT.Default.TLSKeyFile

Spécifiez le chemin d'accès complet du fichier contenant la clé privée MQTT pour les communications chiffrées entre l'agent 2 Zabbix et le broker MQTT.
 Utilisé si aucune valeur n'est spécifiée dans une session nommée.

Plugins.MQTT.Default.Topic

Spécifie le topic par défaut pour l'abonnement MQTT.
 Utilisé si aucune valeur n'est spécifiée dans une clé d'élément ou une session nommée.
 Le topic peut contenir des caractères génériques ("+","#").

Exemples :

```
path/to/file
path/to/#
path/+/topic
```

Plugins.MQTT.Default.Url

Spécifie la chaîne de connexion par défaut du courtier MQTT.
 Utilisé si aucune valeur n'est spécifiée dans une clé d'élément ou une session nommée.
 Ne doit pas inclure de paramètres de requête.
 Doit correspondre au format d'URL.
 Schémas pris en charge: `tcp` (par défaut), `ws`, `tls`
 Un schéma peut être omis.
 Un port peut être omis (par défaut=1883)

Par défaut: `tcp://localhost:1883`

Exemples:

```
tcp://host:1883
localhost
ws://host:8080
```

Plugins.MQTT.Default.User

Spécifiez le nom d'utilisateur par défaut pour la connexion à MQTT.
 Utilisé si aucune valeur n'est spécifiée dans une clé d'élément ou une session nommée.

Plugins.MQTT.Sessions.<SessionName>.Password

Spécifiez le mot de passe de la session nommée pour la connexion à MQTT.
 <SessionName> - définissez le nom d'une session à utiliser dans les clés d'élément.

Plugins.MQTT.Sessions.<SessionName>.TLSCAFile

Spécifiez le chemin d'accès complet du fichier contenant le ou les certificats d'AC racine pour la vérification du certificat du pair pour la session nommée, utilisé pour les communications chiffrées entre l'agent 2 Zabbix et le broker MQTT.
 <SessionName> - définit le nom d'une session à utiliser dans les clés d'élément.

Plugins.MQTT.Sessions.<SessionName>.TLSCertFile

Spécifiez le chemin complet du fichier contenant le certificat de l'agent ou la chaîne de certificats pour la session nommée, utilisé pour les communications chiffrées entre Zabbix agent 2 et le broker MQTT.
 <SessionName> - définit le nom d'une session à utiliser dans les clés d'élément.

Plugins.MQTT.Sessions.<SessionName>.TLSKeyFile

Spécifiez le chemin d'accès complet du fichier contenant la clé privée MQTT pour la session nommée, utilisé pour les communications chiffrées entre l'agent Zabbix 2 et le broker MQTT.
 <SessionName> - définissez le nom d'une session à utiliser dans les clés d'élément.

Plugins.MQTT.Sessions.<SessionName>.Topic

Spécifiez le topic de session nommé pour l'abonnement MQTT.
 <SessionName> - définissez le nom d'une session à utiliser dans les clés d'élément.
 Le topic peut contenir des caractères génériques ("+","#").

Exemples :

```
path/to/file
path/to/#
path/+/topic
```

Plugins.MQTT.Sessions.<SessionName>.Url

Spécifie la chaîne de connexion du courtier MQTT de la session nommée.
 <SessionName> - définit le nom d'une session à utiliser dans les clés d'élément.
 Ne doit pas inclure de paramètres de requête.
 Doit correspondre au format d'URL.
 Schémas pris en charge: `tcp` (par défaut), `ws`, `tls`
 Un schéma peut être omis.
 Un port peut être omis (par défaut=1883)

Par défaut: `tcp://localhost:1883`

Exemples:

```
tcp://host:1883
localhost
ws://host:8080
```

Plugins.MQTT.Sessions.<SessionName>.User

Spécifiez le nom d'utilisateur de la session nommée pour la connexion à MQTT.
 <SessionName> - définissez le nom d'une session à utiliser dans les clés d'élément.

Si vous devez définir le délai d'attente d'exécution de la requête (durée d'attente avant d'interrompre une requête pour qu'elle se termine), utilisez le formulaire de configuration de l'élément.

Voir aussi

- Description des paramètres de configuration généraux de Zabbix agent 2 : Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- Instructions de configuration des plugins

Plugin MSSQL

Vue d'ensemble

Les paramètres de configuration permettent de personnaliser le comportement du plugin MSSQL.

Il est recommandé de les spécifier dans leur propre fichier de configuration (par exemple, `mssql.conf`), plutôt que dans le fichier de configuration de l'agent 2. Utilisez ensuite la directive Include pour inclure ce fichier dans la configuration de Zabbix agent 2.

Le plugin MSSQL est un plugin chargeable et est disponible, avec une description complète, dans le dépôt du plugin MSSQL.

Tous les paramètres sont **facultatifs**, sauf indication explicite précisant que le paramètre est obligatoire.

Cette page est structurée comme suit :

- Vue d'ensemble des paramètres (pour voir les détails complets d'un paramètre, cliquez sur son nom)
- Détails des paramètres

Parameter	Description
Plugins.MSSQL.CustomQueriesPath	Spécifiez le chemin du répertoire contenant les fichiers `.sql` définis par l'utilisateur avec des requêtes personnalisées que le plugin peut exécuter.
Plugins.MSSQL.CustomQueriesEnabled	Activez l'exécution de la clé d'élément `mssql.custom.query`.
Plugins.MSSQL.Default.CAExtraFile	Spécifiez le chemin de fichier par défaut vers le certificat de clé publique de l'autorité de certification (CA) qui a émis le certificat du serveur MSSQL.
Plugins.MSSQL.Default.Database	Spécifiez le nom de base de données par défaut.
Plugins.MSSQL.Default.Encrypt	Spécifiez le niveau de chiffrement par défaut.
Plugins.MSSQL.Default.HostnameInCertificate	Spécifiez par défaut le nom commun (CN) du certificat du serveur MSSQL.
Plugins.MSSQL.Default.Password	Spécifiez le mot de passe par défaut.
Plugins.MSSQL.Default.TLSMinVersion	Spécifiez par défaut la version minimale de TLS.
Plugins.MSSQL.Default.TrustServerCertificate	Faites confiance au certificat du serveur sans le valider par défaut.
Plugins.MSSQL.Default.Uri	Spécifiez l'URI par défaut.
Plugins.MSSQL.Default.User	Spécifiez le nom d'utilisateur par défaut.
Plugins.MSSQL.KeepAlive	Spécifiez le temps d'attente maximal (en secondes) avant la fermeture des connexions inutilisées du plugin.
Plugins.MSSQL.Sessions.<SessionName>.CAExtraFile	Spécifiez le chemin du fichier de session nommée vers le certificat de clé publique de l'autorité de certification (CA) qui a émis le certificat du serveur MSSQL.
Plugins.MSSQL.Sessions.<SessionName>.Database	Spécifiez le nom de base de données de la session nommée.
Plugins.MSSQL.Sessions.<SessionName>.Encrypt	Spécifiez le niveau de chiffrement de la session nommée.
Plugins.MSSQL.Sessions.<SessionName>.HostnameInCertificate	Spécifiez le nom commun (CN) du certificat du serveur MSSQL pour la session nommée.
Plugins.MSSQL.Sessions.<SessionName>.Password	Spécifiez le mot de passe de la session nommée.
Plugins.MSSQL.Sessions.<SessionName>.TLSMinVersion	Spécifiez la version minimale de TLS de la session nommée.
Plugins.MSSQL.Sessions.<SessionName>.TrustServerCertificate	Faites confiance au certificat du serveur de la session nommée sans le valider.
Plugins.MSSQL.Sessions.<SessionName>.Uri	Spécifiez l'URI de la session nommée.
Plugins.MSSQL.Sessions.<SessionName>.User	Spécifiez le nom d'utilisateur de la session nommée.
Plugins.MSSQL.System.Path	Spécifiez le chemin de l'exécutable du plugin MSSQL.
Plugins.MSSQL.Timeout	Définissez la durée, en secondes, d'attente d'une réponse du serveur lors de la première connexion et lors des opérations suivantes dans la session.

Notez que :

- Les valeurs par défaut reflètent les valeurs par défaut du processus, et non celles des fichiers de configuration fournis.
- Les valeurs prennent en charge les variables d'environnement ;
- Zabbix prend en charge les fichiers de configuration uniquement en encodage UTF-8 sans BOM.

- Les commentaires commençant par "#" ne sont pris en charge qu'au début de la ligne.

Détails du paramètre

Plugins.MSSQL.CustomQueriesDir

Spécifie le chemin du répertoire contenant les fichiers `.sql` définis par l'utilisateur avec des requêtes personnalisées que le plugin peut exécuter.
 Le plugin charge tous les fichiers .sql disponibles dans le répertoire configuré au démarrage. Cela signifie que toute modification apportée aux fichiers de requêtes personnalisées ne sera prise en compte qu'après le redémarrage du plugin. Le plugin est démarré et arrêté en même temps que Zabbix agent 2.

Valeurs par défaut :
`/usr/local/share/zabbix/custom-queries/mssql` - pour les systèmes Unix,
`*:\Program Files\Zabbix Agent 2\Custom Queries\MSSQL` - pour les systèmes Windows (où `*` est le nom du lecteur provenant de la variable d'environnement `ProgramFiles`)

Plugins.MSSQL.CustomQueriesEnabled

Active l'exécution de la clé d'élément `mssql.custom.query`.
 Si cette option est désactivée, aucune requête ne sera chargée depuis le chemin du répertoire des requêtes personnalisées.

Par défaut: `false`
 Plage: `true/false`

Plugins.MSSQL.Default.CACertPath

Spécifie le chemin de fichier par défaut vers le certificat de clé publique de l'autorité de certification (CA) qui a émis le certificat du serveur MSSQL.
 Le certificat doit être au format PEM.

Plugins.MSSQL.Default.Database

Spécifiez le nom de la base de données par défaut.

Plugins.MSSQL.Default.Encrypt

Spécifie le niveau de chiffrement par défaut.

Plage :
true - l'envoi des données entre le plugin et le serveur est chiffré ;
false - l'envoi des données entre le plugin et le serveur n'est pas chiffré au-delà du paquet de connexion ;
strict - l'envoi des données entre le plugin et le serveur est chiffré de bout en bout à l'aide de TDS8 ;
disable - l'envoi des données entre le plugin et le serveur n'est pas chiffré.

Plugins.MSSQL.Default.HostNameInCertificate

Spécifiez par défaut le nom commun (CN) du certificat du serveur MSSQL.

Plugins.MSSQL.Default.Password

Spécifiez le mot de passe par défaut pour un serveur MSSQL protégé.

Plugins.MSSQL.Default.TLSMinVersion

Spécifiez par défaut la version minimale de TLS.

Plage : `1.0`, `1.1`, `1.2`, `1.3`

Plugins.MSSQL.Default.TrustServerCertificate

Faire confiance au certificat du serveur sans le valider par défaut.

Plage : `true`, `false`

Plugins.MSSQL.Default.Uri

Spécifiez l'URI par défaut pour la connexion au serveur MSSQL.
 Les identifiants intégrés seront ignorés.
 Le seul schéma pris en charge est `sqlserver://`.
 Le schéma peut être omis.

Par défaut: `sqlserver://localhost:1433`

Plugins.MSSQL.Default.User

Spécifiez le nom d'utilisateur par défaut pour un serveur MSSQL protégé.

Plugins.MSSQL.KeepAlive

Spécifie le temps d'attente maximal (en secondes) avant que les connexions de plugin inutilisées ne soient fermées.

Par défaut: 300
 Plage: 60-900

Plugins.MSSQL.Sessions.<SessionName>.CACertPath

Spécifiez le chemin du fichier de session nommé vers le certificat de clé publique de l'autorité de certification (CA) qui a émis le certificat du serveur MSSQL.
 <SessionName> - définit le nom d'une session à utiliser dans les clés d'élément.
 Le certificat doit être au format PEM.

Plugins.MSSQL.Sessions.<SessionName>.Database

Spécifie le nom de la base de données de la session nommée.
 <SessionName> - définit le nom d'une session à utiliser dans les clés d'élément.

Plugins.MSSQL.Sessions.<SessionName>.Encrypt

Spécifie le niveau de chiffrement de la session nommée.
 <SessionName> - définit le nom d'une session à utiliser dans les clés d'élément.

Plage :
true - les données envoyées entre le plugin et le serveur sont chiffrées ;
false - les données envoyées entre le plugin et le serveur ne sont pas chiffrées au-delà du paquet de connexion ;
strict - les données envoyées entre le plugin et le serveur sont chiffrées de bout en bout à l'aide de TDS8 ;
disable - les données envoyées entre le plugin et le serveur ne sont pas chiffrées.

Plugins.MSSQL.Sessions.<SessionName>.HostNameInCertificate

Spécifiez le nom commun (CN) du certificat du serveur MSSQL pour la session nommée.
 <SessionName> - définissez le nom d'une session à utiliser dans les clés d'élément.

Plugins.MSSQL.Sessions.<SessionName>.Password

Spécifiez le mot de passe de la session nommée pour un serveur MSSQL protégé.
 <SessionName> - définit le nom d'une session à utiliser dans les clés d'élément.

Plugins.MSSQL.Sessions.<SessionName>.TLSMinVersion

Spécifie la version TLS minimale de la session nommée.
 <SessionName> - définit le nom d'une session à utiliser dans les clés d'élément.

Plage : `1.0`, `1.1`, `1.2`, `1.3`

Plugins.MSSQL.Sessions.<SessionName>.TrustServerCertificate

Faire confiance au certificat du serveur de la session nommée sans le valider.
 <SessionName> - définit le nom d'une session à utiliser dans les clés d'élément.

Plage : `true`, `false`

Plugins.MSSQL.Sessions.<SessionName>.Uri

Spécifiez l'URI de session nommée pour la connexion au serveur MSSQL.
 <SessionName> - définit le nom d'une session à utiliser dans les clés d'élément.
 Les identifiants intégrés seront ignorés.
 Le seul schéma pris en charge est `sqlserver://`.
 Le schéma peut être omis.

Par défaut: `sqlserver://localhost:1433`

Plugins.MSSQL.Sessions.<SessionName>.User

Spécifiez le nom d'utilisateur de la session nommée pour un serveur MSSQL protégé.
 <SessionName> - définissez le nom d'une session à utiliser dans les clés d'élément.

Plugins.MSSQL.System.Path

Spécifiez le chemin vers l'exécutable du plugin MSSQL.
 Paramètre global pour le plugin MSSQL. Appliqué à toutes les connexions.

Exemple: `Plugins.MSSQL.System.Path=/usr/libexec/zabbix/zabbix-agent2-plugin-mssql`

Plugins.MSSQL.Timeout

Définissez la durée, en secondes, d'attente d'une réponse du serveur lors de la première connexion et des opérations suivantes dans la session.

Par défaut: `global timeout`
 Plage: 1-30

Options

Parameter	Description
-V --version	Afficher la version du plugin et les informations de licence.
-h --help	Afficher les informations d'aide (raccourci).

Parameter	Description
-t, --test <item key>	Lancer le plugin pour les tests (la configuration du plugin est ignorée).

Voir aussi

- Description des paramètres de configuration généraux de Zabbix agent 2 : Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- Instructions pour la configuration des plugins

Plugin MySQL

Vue d'ensemble

Les paramètres de configuration permettent de personnaliser le comportement du plugin MySQL.

Il est recommandé de les spécifier dans leur propre fichier de configuration (par exemple, `mysql.conf`), plutôt que dans le fichier de configuration de l'agent 2. Utilisez ensuite la directive Include pour inclure ce fichier dans la configuration de Zabbix agent 2.

Tous les paramètres sont **facultatifs**, sauf indication explicite que le paramètre est obligatoire.

Cette page est structurée comme suit :

- Vue d'ensemble des paramètres (pour voir tous les détails d'un paramètre, cliquez sur son nom)
- Détails des paramètres

Parameter	Description
Plugins.Mysql.CallTimeout	Spécifie le délai maximal, en secondes, à attendre qu'une requête soit exécutée.
Plugins.Mysql.CustomQueriesPath	Spécifie le chemin complet du répertoire contenant les fichiers `.sql` avec des requêtes personnalisées.
Plugins.Mysql.CustomQueriesEnabled	Active l'exécution de la clé d'élément `mysql.custom.query`.
Plugins.Mysql.Default.Password	Spécifie le mot de passe par défaut.
Plugins.Mysql.Default.TLSConnect	Spécifie le niveau de chiffrement par défaut pour les communications entre Zabbix agent 2 et les bases de données surveillées.
Plugins.Mysql.Default.TLSCAFile	Spécifie le chemin complet du fichier contenant les certificats des autorités de certification de niveau supérieur pour la vérification du certificat du pair.
Plugins.Mysql.Default.TLSCertFile	Spécifie le chemin complet du fichier contenant le certificat de l'agent ou la chaîne de certificats.
Plugins.Mysql.Default.TLSKeyFile	Spécifie le chemin complet du fichier contenant la clé privée de la base de données.
Plugins.Mysql.Default.Uri	Spécifie l'URI par défaut.
Plugins.Mysql.Default.User	Spécifie le nom d'utilisateur par défaut.
Plugins.Mysql.KeepAlive	Spécifie le temps d'attente maximal, en secondes, avant la fermeture des connexions de plugin inutilisées.
Plugins.Mysql.Sessions.<SessionName>.Password	Spécifie le mot de passe de la session nommée.
Plugins.Mysql.Sessions.<SessionName>.TLSConnect	Spécifie le niveau de chiffrement de la session nommée pour les communications entre Zabbix agent 2 et les bases de données surveillées.
Plugins.Mysql.Sessions.<SessionName>.TLSCAFile	Spécifie le chemin complet du fichier contenant les certificats des autorités de certification de niveau supérieur pour la vérification du certificat du pair de la session nommée.
Plugins.Mysql.Sessions.<SessionName>.TLSCertFile	Spécifie le chemin complet du fichier contenant le certificat de l'agent ou la chaîne de certificats de la session nommée.
Plugins.Mysql.Sessions.<SessionName>.TLSKeyFile	Spécifie le chemin complet du fichier contenant la clé privée de la base de données de la session nommée.
Plugins.Mysql.Sessions.<SessionName>.Uri	Spécifie l'URI de la session nommée.
Plugins.Mysql.Sessions.<SessionName>.User	Spécifie le nom d'utilisateur de la session nommée.
Plugins.Mysql.Timeout	Définit le délai maximal, en secondes, d'attente lors de l'établissement d'une connexion.

Notez que :

- Les valeurs par défaut reflètent les valeurs par défaut du processus, et non celles des fichiers de configuration fournis ;
- Les valeurs prennent en charge les variables d'environnement ;
- Zabbix prend en charge les fichiers de configuration uniquement en encodage UTF-8 sans BOM ;
- Les commentaires commençant par "#" ne sont pris en charge qu'au début de la ligne.

Détails du paramètre

Plugins.Mysql.CallTimeout

Spécifie la durée maximale, en secondes, d'attente pour qu'une requête soit exécutée.

Par défaut: `global timeout`
 Plage: 1-30

Plugins.Mysql.CustomQueriesPath

Spécifiez le chemin complet du répertoire contenant les fichiers `.sql` avec des requêtes personnalisées.

Valeurs par défaut :
`/usr/local/share/zabbix/custom-queries/mysql` - pour les systèmes Unix,
`*:\Program Files\Zabbix Agent 2\Custom Queries\Mysql` - pour les systèmes Windows (où `*` est le nom du lecteur provenant de la variable d'environnement `ProgramFiles`)

Plugins.Mysql.CustomQueriesEnabled

Activer l'exécution de la clé d'élément `mysql.custom.query`.
 Si cette option est désactivée, aucune requête ne sera chargée depuis le chemin du répertoire des requêtes personnalisées.

Par défaut: `false`
 Plage: `true/false`

Plugins.Mysql.Default.Password

Spécifiez le mot de passe par défaut pour un serveur MySQL protégé.

Plugins.Mysql.Default.TLSConnect

Spécifiez le niveau de chiffrement par défaut pour les communications entre Zabbix agent 2 et les bases de données surveillées.

Plage :
required - exiger une connexion TLS;
verify_ca - vérifier les certificats;
verify_full - vérifier les certificats et l'adresse IP

Plugins.Mysql.Default.TLSCAFile

Spécifiez le chemin complet du fichier contenant le ou les certificats d'autorité de certification racine pour la vérification du certificat du pair, utilisé pour les communications chiffrées entre l'agent 2 Zabbix et les bases de données surveillées.

Obligatoire: `yes`, si `Plugins.Mysql.Default.TLSConnect` est défini sur `verify_ca` ou `verify_full`

Plugins.Mysql.Default.TLSCertFile

Spécifiez le chemin complet du fichier contenant le certificat de l'agent ou la chaîne de certificats, utilisé pour les communications chiffrées entre l'agent Zabbix 2 et les bases de données surveillées.

Obligatoire: `yes`, si `Plugins.Mysql.Default.TLSKeyFile` est spécifié

Plugins.Mysql.Default.TLSKeyFile

Indiquez le chemin d'accès complet du fichier contenant la clé privée de la base de données, utilisée pour les communications chiffrées entre l'agent 2 Zabbix et les bases de données surveillées.

Obligatoire: `yes`, si `Plugins.Mysql.Default.TLSCertFile` est spécifié

Plugins.Mysql.Default.Uri

Spécifiez l'URI par défaut pour la connexion au serveur MySQL.
 Les identifiants intégrés seront ignorés.
 Schémas pris en charge : `tcp`, `unix`
 Le schéma peut être omis.
 Le port peut être omis (par défaut=3306).

Exemples :

```
tcp://localhost:3306
localhost
unix:/var/run/mysql.sock
```

Plugins.Mysql.Default.User

Spécifiez le nom d'utilisateur par défaut pour un serveur MySQL protégé.

Plugins.Mysql.KeepAlive

Spécifiez le temps d'attente maximal (en secondes) avant la fermeture des connexions de plugin inutilisées.

Par défaut: 300
 Plage: 60-900

Plugins.Mysql.Sessions.<SessionName>.Password

Spécifiez le mot de passe de la session nommée pour un serveur MySQL protégé.
 <SessionName> - définissez le nom d'une session à utiliser dans les clés d'élément.

Plugins.Mysql.Sessions.<SessionName>.TLSConnect

Spécifiez le niveau de chiffrement de la session nommée pour les communications entre l'agent 2 Zabbix et les bases de données surveillées.
 <SessionName> - définit le nom d'une session à utiliser dans les clés d'élément.

Plage :
required - exiger une connexion TLS ;
verify_ca - vérifier les certificats ;
verify_full - vérifier les certificats et l'adresse IP

Plugins.Mysql.Sessions.<SessionName>.TLSCAFile

Spécifiez le chemin complet du fichier contenant les certificats des autorités de certification racines pour la vérification du certificat du pair dans une session nommée, utilisé pour les communications chiffrées entre Zabbix agent 2 et les bases de données surveillées.
 <SessionName> - définissez le nom d'une session à utiliser dans les clés d'élément.

Obligatoire: `yes`, si `Plugins.Mysql.Sessions.<SessionName>.TLSConnect` est défini sur `verify_ca` ou `verify_full`

Plugins.Mysql.Sessions.<SessionName>.TLSCertFile

Spécifiez le chemin d'accès complet du fichier contenant le certificat de l'agent ou la chaîne de certificats de la session nommée, utilisé pour les communications chiffrées entre Zabbix agent 2 et les bases de données surveillées.
 <SessionName> - définit le nom d'une session à utiliser dans les clés d'élément.

Obligatoire: `yes`, si `Plugins.Mysql.Sessions.<SessionName>.TLSKeyFile` est spécifié

Plugins.Mysql.Sessions.<SessionName>.TLSKeyFile

Spécifiez le chemin d'accès complet du fichier contenant la clé privée de la base de données de la session nommée, utilisée pour les communications chiffrées entre l'agent 2 Zabbix et les bases de données surveillées.
 <SessionName> - définissez le nom d'une session à utiliser dans les clés d'éléments.

Obligatoire: `yes`, si `Plugins.Mysql.Sessions.<SessionName>.TLSCertFile` est spécifié

Plugins.Mysql.Sessions.<SessionName>.Uri

Spécifiez l'URI de session nommée pour la connexion au serveur MySQL.
 <SessionName> - définit le nom d'une session à utiliser dans les clés d'élément.
 Les identifiants intégrés seront ignorés.
 Schémas pris en charge: `tcp`, `unix`
 Le schéma peut être omis.
 Le port peut être omis (par défaut=3306).

Exemples:

```
tcp://localhost:3306
localhost
unix:/var/run/mysql.sock
```

Plugins.Mysql.Sessions.<SessionName>.User

Spécifiez le nom d'utilisateur de la session nommée pour un serveur MySQL protégé.
 <SessionName> - définissez le nom d'une session à utiliser dans les clés d'élément.

Plugins.Mysql.Timeout

Définit le temps maximal, en secondes, d'attente lors de l'établissement d'une connexion.

Par défaut: `global timeout`
 Plage: 1–30

Voir aussi

- Description des paramètres de configuration généraux de Zabbix agent 2 : Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- Instructions pour configurer les plugins

Plugin Oracle

Vue d'ensemble

Les paramètres de configuration permettent de personnaliser le comportement du plugin Oracle.

Il est recommandé de les spécifier dans leur propre fichier de configuration (par exemple, `oracle.conf`), plutôt que dans le fichier de configuration de l'agent 2. Utilisez ensuite la directive Include pour inclure ce fichier dans la configuration de Zabbix agent 2.

Tous les paramètres sont **facultatifs**, sauf indication explicite précisant que le paramètre est obligatoire.

Cette page est structurée comme suit :

- Vue d'ensemble des paramètres (pour voir les détails complets d'un paramètre, cliquez sur son nom)
- Détails des paramètres

Parameter	Description
Plugins.Oracle.CallTimeout	Le temps d'attente maximal, en secondes, pour qu'une requête soit exécutée. Ce paramètre n'a actuellement aucun effet ; le réglage item timeout est utilisé à la place.
Plugins.Oracle.ConnectTimeout	Spécifiez le temps d'attente maximal, en secondes, pour l'établissement d'une connexion.
Plugins.Oracle.CustomQueriesPath	Spécifiez le chemin complet du répertoire contenant les fichiers `.sql` avec des requêtes personnalisées.
Plugins.Oracle.CustomQueriesEnabled	Active l'exécution de la clé d'élément `oracle.custom.query`.
Plugins.Oracle.Default.Password	Spécifiez le mot de passe par défaut.
Plugins.Oracle.Default.Service	Spécifiez le nom de service par défaut.
Plugins.Oracle.Default.URI	Spécifiez l'URI par défaut.
Plugins.Oracle.Default.User	Spécifiez le nom d'utilisateur par défaut.
Plugins.Oracle.KeepAlive	Définissez le temps d'attente maximal, en secondes, avant la fermeture des connexions de plugin inutilisées.
Plugins.Oracle.ResolveTNS	Spécifiez comment interpréter la chaîne de connexion (ConnString) pour la connexion au serveur Oracle.
Plugins.Oracle.Sessions.<SessionName>.Password	Spécifiez le mot de passe de la session nommée.
Plugins.Oracle.Sessions.<SessionName>.Service	Spécifiez le nom de service de la session nommée.
Plugins.Oracle.Sessions.<SessionName>.URI	Spécifiez l'URI de la session nommée.
Plugins.Oracle.Sessions.<SessionName>.User	Spécifiez le nom d'utilisateur de la session nommée.

Notez que :

- Les valeurs par défaut reflètent les valeurs par défaut du processus, et non celles présentes dans les fichiers de configuration fournis.
- Les valeurs prennent en charge les variables d'environnement.
- Zabbix prend en charge les fichiers de configuration uniquement en encodage UTF-8 sans BOM.
- Les commentaires commençant par "#" ne sont pris en charge qu'au début de la ligne.

Détails du paramètre

Plugins.Oracle.CallTimeout

Le temps d'attente maximal, en secondes, pour qu'une requête soit exécutée.
 Ce paramètre n'a actuellement aucun effet. Le temps d'attente maximal est déterminé par le délai d'attente de l'élément à la place.

Par défaut: `Timeout`[1]
 Plage: 1-30

Plugins.Oracle.ConnectTimeout

Spécifie le temps d'attente maximal, en secondes, pour l'établissement d'une connexion.

Par défaut: `Timeout`[1]
 Plage: 1-30

Plugins.Oracle.CustomQueriesPath

Spécifiez le chemin complet du répertoire contenant les fichiers `.sql` avec des requêtes personnalisées.

Valeurs par défaut :
`/usr/local/share/zabbix/custom-queries/oracle` - pour les systèmes Unix,
`*:\Program Files\Zabbix Agent 2\Custom Queries\Oracle` - pour les systèmes Windows (où `*` est le nom du lecteur provenant de la variable d'environnement `ProgramFiles`)

Exemple :

```
/etc/zabbix/oracle/sql
```

Plugins.Oracle.CustomQueriesEnabled

Active l'exécution de la clé d'élément `oracle.custom.query`.
 Si cette option est désactivée, aucune requête ne sera chargée depuis le chemin du répertoire des requêtes personnalisées.

Par défaut: `false`
 Plage: `true/false`

Plugins.Oracle.Default.Password

Spécifiez le mot de passe par défaut pour la connexion à Oracle.
 Utilisé si aucune valeur n'est spécifiée dans une clé d'élément ou une session nommée.

Plugins.Oracle.Default.Service

Spécifiez le nom de service par défaut pour la connexion à Oracle (SID n'est pas pris en charge).
 Utilisé si aucune valeur n'est spécifiée dans une clé d'élément ou une session nommée.

Plugins.Oracle.Default.Uri

Spécifiez l'URI par défaut pour la connexion à Oracle.
 Utilisé si aucune valeur n'est spécifiée dans une clé d'élément ou une session nommée.
 Les identifiants intégrés seront ignorés.
 Doit correspondre au format URI.
 Schémas pris en charge: `tcp`, `tcps`
 Le schéma peut être omis.
 Le port peut être omis (default=1521).
 Il est également possible de spécifier la clé TNS ou la valeur TNS comme chaîne de connexion. La valeur TNS doit être composée sans espaces.

Exemples:

```
tcp://127.0.0.1:1521
localhost
zbx_tns_example (TNS key)
(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=localhost)(PORT=1521))(CONNECT_DATA=(SERVICE_NAME=xe))) (TNS val
```

Plugins.Oracle.Default.User

Spécifiez le nom d'utilisateur par défaut pour la connexion à Oracle.
 Utilisé si aucune valeur n'est spécifiée dans une clé d'élément ou une session nommée.

Plugins.Oracle.KeepAlive

Définissez le temps d'attente maximal (en secondes) avant la fermeture des connexions de plugin inutilisées.

Par défaut: 300
 Plage: 60-900

Plugins.Oracle.ResolveTNS

Spécifiez comment interpréter la chaîne de connexion (ConnString) pour se connecter au serveur Oracle.
 Si elle est définie sur `false`, le schéma et le port par défaut seront ajoutés à ConnString (s'ils sont absents).
 Si l'option est définie sur `true`, le schéma et le port par défaut seront omis (sauf s'ils sont explicitement spécifiés dans ConnString), et ConnString sera transmis au client Oracle tel quel. Si le client Oracle trouve ConnString dans le fichier `tnsnames.ora`, la description de connexion trouvée sera utilisée pour se connecter au serveur Oracle.

Par défaut: `true`
 Plage: `true/false`

Plugins.Oracle.Sessions.<SessionName>.Password

Spécifiez le mot de passe de la session nommée pour la connexion à Oracle.
 <SessionName> - définissez le nom d'une session à utiliser dans les clés d'élément.

Plugins.Oracle.Sessions.<SessionName>.Service

Spécifiez le nom du service de session nommé pour la connexion à Oracle (SID n'est pas pris en charge).
 <SessionName> - définissez le nom d'une session à utiliser dans les clés d'élément.

Plugins.Oracle.Sessions.<SessionName>.Uri

Spécifiez l'URI de session nommée pour la connexion à Oracle.
 <SessionName> - définit le nom d'une session à utiliser dans les clés d'élément.
 Les identifiants intégrés seront ignorés.
 Doit correspondre au format URI.
 Schémas pris en charge: `tcp`, `tcps`
 Le schéma peut être omis.
 Le port peut être omis (default=1521).
 Il est également possible de spécifier la clé TNS ou la valeur TNS comme chaîne de connexion. La valeur TNS doit être composée sans espaces.

Exemples:

```
tcp://127.0.0.1:1521
localhost
zbx_tns_example (TNS key)
(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=localhost)(PORT=1521))(CONNECT_DATA=(SERVICE_NAME=xe))) (TNS val
```

Plugins.Oracle.Sessions.<SessionName>.User

Spécifiez le nom d'utilisateur de la session nommée pour la connexion à Oracle.
 <SessionName> - définissez le nom d'une session à utiliser dans les clés d'élément.

Notes de bas de page

[1] Le délai d'expiration est défini par l'option `Timeout` dans le fichier de configuration (`zabbix_agent2.conf`) de l'agent 2 (UNIX) ou de l'agent 2 (Windows) Zabbix.

Voir aussi

- Description des paramètres de configuration généraux de Zabbix agent 2 : Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- Instructions pour configurer les plugins

Plugin PostgreSQL

Vue d'ensemble

Les paramètres de configuration permettent de personnaliser le comportement du plugin PostgreSQL.

Il est recommandé de les spécifier dans leur propre fichier de configuration (par exemple, `postgresql.conf`), plutôt que dans le fichier de configuration de l'agent 2. Utilisez ensuite la directive Include pour inclure ce fichier dans la configuration de Zabbix agent 2.

Le plugin PostgreSQL est un plugin chargeable et est disponible, avec une description complète, dans le dépôt du plugin PostgreSQL.

Tous les paramètres sont **facultatifs**, sauf indication explicite précisant que le paramètre est obligatoire.

Cette page est structurée comme suit :

- Vue d'ensemble des paramètres (pour voir les détails complets d'un paramètre, cliquez sur son nom)
- Détails des paramètres

Parameter	Description
Plugins.PostgreSQL.CallTimeout	Définit le temps d'attente maximal, en secondes, pour qu'une requête soit exécutée.
Plugins.PostgreSQL.CustomQueriesPath	Spécifie le chemin complet du répertoire contenant les fichiers `.sql` avec des requêtes personnalisées.
Plugins.PostgreSQL.CustomQueriesEnabled	Active l'exécution de la clé d'élément `postgresql.custom.query`.
Plugins.PostgreSQL.Default.CacheMode	Spécifie le mode de cache par défaut pour la connexion PostgreSQL.
Plugins.PostgreSQL.Default.Database	Spécifie le nom de base de données par défaut.
Plugins.PostgreSQL.Default.Password	Spécifie le mot de passe par défaut.
Plugins.PostgreSQL.Default.TLSCAFile	Spécifie le chemin complet du fichier contenant le ou les certificats CA de niveau supérieur pour la vérification du certificat du pair.
Plugins.PostgreSQL.Default.TLSCertFile	Spécifie le chemin complet du fichier contenant le certificat PostgreSQL ou la chaîne de certificats.
Plugins.PostgreSQL.Default.TLSConnect	Spécifie le niveau de chiffrement par défaut pour les communications entre Zabbix agent 2 et les bases de données surveillées.
Plugins.PostgreSQL.Default.TLSKeyFile	Spécifie le chemin complet du fichier contenant la clé privée PostgreSQL.
Plugins.PostgreSQL.Default.Uri	Spécifie l'URI par défaut.
Plugins.PostgreSQL.Default.User	Spécifie le nom d'utilisateur par défaut.
Plugins.PostgreSQL.KeepAlive	Définit le temps d'attente maximal, en secondes, avant la fermeture des connexions du plugin inutilisées.
Plugins.PostgreSQL.Sessions.<SessionName>.CacheMode	Spécifie le mode de cache de la session nommée pour la connexion PostgreSQL.
Plugins.PostgreSQL.Sessions.<SessionName>.Database	Spécifie le nom de base de données de la session nommée.
Plugins.PostgreSQL.Sessions.<SessionName>.Password	Spécifie le mot de passe de la session nommée.
Plugins.PostgreSQL.Sessions.<SessionName>.TLSCAFile	Spécifie le chemin complet du fichier contenant le ou les certificats CA de niveau supérieur pour la vérification du certificat du pair.
Plugins.PostgreSQL.Sessions.<SessionName>.TLSCertFile	Spécifie le chemin complet du fichier contenant le certificat PostgreSQL ou la chaîne de certificats.
Plugins.PostgreSQL.Sessions.<SessionName>.TLSConnect	Spécifie le niveau de chiffrement de la session nommée pour les communications entre Zabbix agent 2 et les bases de données surveillées.
Plugins.PostgreSQL.Sessions.<SessionName>.TLSKeyFile	Spécifie le chemin complet du fichier contenant la clé privée PostgreSQL.
Plugins.PostgreSQL.Sessions.<SessionName>.Uri	Spécifie l'URI de la session nommée.
Plugins.PostgreSQL.Sessions.<SessionName>.User	Spécifie le nom d'utilisateur de la session nommée.

|Plugins.PostgreSQL.System.Path|Spécifie le chemin vers l'exécutable du plugin PostgreSQL.| |Plugins.PostgreSQL.Timeout|Définit le délai d'exécution de la requête.|

Notez que :

- Les valeurs par défaut reflètent les valeurs par défaut du processus, et non celles des fichiers de configuration fournis.
- Les valeurs prennent en charge les variables d'environnement ;
- Zabbix prend en charge les fichiers de configuration uniquement en encodage UTF-8 sans BOM.
- Les commentaires commençant par "#" ne sont pris en charge qu'au début de la ligne.

Détails du paramètre

Plugins.PostgreSQL.CallTimeout

Définit le temps d'attente maximal, en secondes, pour qu'une requête soit exécutée.

Par défaut: `global timeout`
 Plage: 1-30

Plugins.PostgreSQL.CustomQueriesPath

Spécifiez le chemin complet du répertoire contenant les fichiers `.sql` avec des requêtes personnalisées.

Valeurs par défaut :
`/usr/local/share/zabbix/custom-queries/postgreSQL` - pour les systèmes Unix,
`*:\Program Files\Zabbix Agent 2\Custom Queries\PostgreSQL` - pour les systèmes Windows (où `*` est le nom du lecteur provenant de la variable d'environnement `ProgramFiles`)

Exemple :

```
/etc/zabbix/oracle/sql
```

Plugins.PostgreSQL.CustomQueriesEnabled

Active l'exécution de la clé d'élément `postgresql.custom.query`.
 Si cette option est désactivée, aucune requête ne sera chargée depuis le chemin du répertoire des requêtes personnalisées.

Par défaut: `false`
 Plage: `true/false`

Plugins.PostgreSQL.Default.CacheMode

Spécifie le mode de cache par défaut pour la connexion PostgreSQL.

Par défaut: `prepare`
 Plage:
`prepare` - crée des instructions préparées sur le serveur PostgreSQL;
`describe` - utilise l'instruction préparée anonyme pour décrire une instruction sans en créer une sur le serveur.

Notez que `describe` est principalement utile lorsque l'environnement n'autorise pas les instructions préparées, par exemple lors de l'utilisation d'un pooler de connexions comme PgBouncer.

Plugins.PostgreSQL.Default.Database

Spécifiez la base de données par défaut pour la connexion à PostgreSQL.
 Utilisée si aucune valeur n'est spécifiée dans une clé d'élément ou une session nommée.

Plugins.PostgreSQL.Default.Password

Spécifiez le mot de passe par défaut pour la connexion à PostgreSQL.
 Utilisé si aucune valeur n'est spécifiée dans une clé d'élément ou une session nommée.

Plugins.PostgreSQL.Default.TLSCAFile

Spécifiez le chemin complet du fichier contenant le certificat de l'autorité de certification (CA) de niveau supérieur pour la vérification du certificat du pair lors des communications chiffrées entre Zabbix agent 2 et les bases de données surveillées.
 Utilisé si aucune valeur n'est spécifiée dans une session nommée.

Obligatoire: `yes`, si `Plugins.PostgreSQL.Default.TLSConnect` est défini sur `verify_ca` ou `verify_full`

Plugins.PostgreSQL.Default.TLSCertFile

Spécifiez le chemin complet du fichier contenant le certificat PostgreSQL ou la chaîne de certificats pour les communications chiffrées entre l'agent Zabbix 2 et les bases de données surveillées.
 Utilisé si aucune valeur n'est spécifiée dans une session nommée.

Obligatoire: `yes`, si `Plugins.PostgreSQL.Default.TLSKeyFile` est spécifié

Plugins.PostgreSQL.Default.TLSConnect

Spécifie le niveau de chiffrement par défaut pour les communications entre Zabbix agent 2 et les bases de données surveillées.
 Utilisé si aucune valeur n'est spécifiée dans une session nommée.

Plage :
`required` - se connecter en utilisant TLS comme mode de transport sans vérification d'identité ;
`verify_ca` - se connecter en utilisant TLS et vérifier le certificat ;
`verify_full` - se connecter en utilisant TLS, vérifier le certificat et vérifier que l'identité de la base de données (CN) spécifiée par DBHost correspond à son certificat.

Un type de chiffrement non défini signifie une connexion non chiffrée.

Plugins.PostgreSQL.Default.TLSKeyFile

Spécifiez le chemin d'accès complet du fichier contenant la clé privée PostgreSQL pour les communications chiffrées entre l'agent 2 Zabbix et les bases de données surveillées.
 Utilisé si aucune valeur n'est spécifiée dans une session nommée.

Obligatoire: `yes`, si `Plugins.PostgreSQL.Default.TLSCertFile` est spécifié

Plugins.PostgreSQL.Default.Uri

Spécifiez l'URI par défaut pour la connexion à PostgreSQL.
 Utilisé si aucune valeur n'est spécifiée dans une clé d'élément ou une session nommée.
 Les identifiants intégrés seront ignorés.
 Doit correspondre au format URI.
 Schémas pris en charge: `tcp`, `unix`

Exemples:

```
tcp://127.0.0.1:5432
tcp://localhost
unix:/var/run/postgresql/.s.PGSQL.5432
```

Plugins.PostgreSQL.Default.User

Spécifiez le nom d'utilisateur par défaut pour la connexion à PostgreSQL.
 Utilisé si aucune valeur n'est spécifiée dans une clé d'élément ou une session nommée.

Plugins.PostgreSQL.KeepAlive

Définissez le temps d'attente maximal (en secondes) avant que les connexions de plugin inutilisées ne soient fermées.

Plugins.PostgreSQL.Sessions.<SessionName>.CacheMode

Spécifiez le mode de cache de session nommé pour la connexion PostgreSQL.
 <SessionName> - définit le nom d'une session à utiliser dans les clés d'élément.

Par défaut: `prepare`
 Plage:
 `prepare` - crée des instructions préparées sur le serveur PostgreSQL;
 `describe` - utilise l'instruction préparée anonyme pour décrire une instruction sans en créer une sur le serveur.

Notez que `describe` est principalement utile lorsque l'environnement n'autorise pas les instructions préparées, par exemple lors de l'utilisation d'un pooler de connexions comme PgBouncer.

Plugins.PostgreSQL.Sessions.<SessionName>.Database

Spécifiez la base de données de session nommée pour la connexion à PostgreSQL.
 <SessionName> - définissez le nom d'une session à utiliser dans les clés d'élément.

Plugins.PostgreSQL.Sessions.<SessionName>.Password

Spécifiez le mot de passe de la session nommée pour la connexion à PostgreSQL.
 <SessionName> - définit le nom d'une session à utiliser dans les clés d'élément.
 Doit correspondre au format du mot de passe.

Plugins.PostgreSQL.Sessions.<SessionName>.TLSCAFile

Spécifiez le chemin d'accès complet du fichier contenant le ou les certificats CA racine pour la vérification du certificat du pair lors des communications chiffrées entre l'agent Zabbix 2 et les bases de données surveillées.
 <SessionName> - définit le nom d'une session à utiliser dans les clés d'élément.

Obligatoire: `yes`, si `Plugins.PostgreSQL.Sessions.<SessionName>.TLSConnect` est défini sur `verify_ca` ou `verify_full`

Plugins.PostgreSQL.Sessions.<SessionName>.TLSCertFile

Spécifiez le chemin complet du fichier contenant le certificat PostgreSQL ou la chaîne de certificats pour les communications chiffrées entre l'agent Zabbix 2 et les bases de données surveillées.
 <SessionName> - définit le nom d'une session à utiliser dans les clés d'élément.

Obligatoire : `yes`, si `Plugins.PostgreSQL.Sessions.<SessionName>.TLSKeyFile` est spécifié

Plugins.PostgreSQL.Sessions.<SessionName>.TLSConnect

Spécifie le niveau de chiffrement nommé de la session pour les communications entre l'agent Zabbix 2 et les bases de données surveillées.
 <SessionName> - définit le nom d'une session à utiliser dans les clés d'élément.

Plage :
`required` - se connecter en utilisant TLS comme mode de transport sans vérification d'identité ;
`verify_ca` - se connecter en utilisant TLS et vérifier le certificat ;
`verify_full` - se connecter en utilisant TLS, vérifier le certificat et vérifier que l'identité de la base de données (CN) spécifiée par DBHost correspond à son certificat.

Un type de chiffrement non défini signifie une connexion non chiffrée.

Plugins.PostgreSQL.Sessions.<SessionName>.TLSKeyFile

Spécifiez le chemin d'accès complet du fichier contenant la clé privée PostgreSQL pour les communications chiffrées entre l'agent Zabbix 2 et les bases de données surveillées.
 <SessionName> - définit le nom d'une session à utiliser dans les clés d'élément.

Obligatoire: `yes`, si `Plugins.PostgreSQL.Sessions.<SessionName>.TLSCertFile` est spécifié.

Plugins.PostgreSQL.Sessions.<SessionName>.Uri

Spécifiez l'URI de session nommée pour la connexion à PostgreSQL.
 <SessionName> - définissez le nom d'une session à utiliser dans les clés d'élément.
 Les identifiants intégrés seront ignorés.
 Doit correspondre au format URI.
 Schémas pris en charge: `tcp`, `unix`

Exemples:

```
tcp://127.0.0.1:5432
tcp://localhost
unix:/var/run/postgresql/.s.PGSQL.5432
```

Plugins.PostgreSQL.Sessions.<SessionName>.User

Spécifiez le nom d'utilisateur de la session nommée pour la connexion à PostgreSQL.
 <SessionName> - définissez le nom d'une session à utiliser dans les clés d'élément.

Plugins.PostgreSQL.System.Path

Spécifiez le chemin vers l'exécutable du plugin PostgreSQL.

Obligatoire: `yes`

Exemple:

```
Plugins.PostgreSQL.System.Path=/usr/libexec/zabbix/zabbix-agent2-plugin-postgresql
```

Plugins.PostgreSQL.Timeout

Définissez le délai d'exécution de la requête (la durée, en secondes, d'attente avant d'interrompre une requête si elle ne se termine pas).

Par défaut: `global timeout`
 Plage: 1-30

Options

Parameter	Description
-V --version	Afficher la version du plugin et les informations de licence.
-h --help	Afficher les informations d'aide (raccourci).
-t, --test <item key>	Lancer le plugin pour les tests (la configuration du plugin est ignorée).

Voir aussi

- Description des paramètres de configuration généraux de Zabbix agent 2 : Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- Instructions pour configurer les plugins

Plugin Redis

Vue d'ensemble

Les paramètres de configuration permettent de personnaliser le comportement du plugin Redis.

Il est recommandé de les spécifier dans leur propre fichier de configuration (par exemple, `redis.conf`), plutôt que dans le fichier de configuration de l'agent 2. Utilisez ensuite la directive Include pour inclure ce fichier dans la configuration de Zabbix agent 2.

Tous les paramètres sont **facultatifs**, sauf indication explicite précisant que le paramètre est obligatoire.

Cette page est structurée comme suit :

- Vue d'ensemble des paramètres (pour voir tous les détails d'un paramètre, cliquez sur son nom)
- Détails des paramètres

Parameter	Description
Plugins.Redis.Default.Password	Spécifiez le mot de passe par défaut pour la connexion à Redis.
Plugins.Redis.Default.TLSConnect	Spécifiez le niveau de chiffrement requis pour les communications de Zabbix agent 2.
Plugins.Redis.Default.TLSCAFile	Spécifiez le chemin complet du fichier contenant les certificats des autorités de certification (CA) de niveau supérieur.
Plugins.Redis.Default.TLSCertFile	Spécifiez le chemin complet du fichier contenant le certificat de l'agent.
Plugins.Redis.Default.TLSKeyFile	Spécifiez le chemin complet du fichier contenant la clé privée de l'agent.
Plugins.Redis.Default.Uri	Spécifiez l'URI par défaut pour la connexion à Redis.

Parameter	Description
Plugins.Redis.Default.User	Spécifiez le nom d'utilisateur par défaut pour la connexion à Redis.
Plugins.Redis.KeepAlive	Définissez le temps d'attente maximal (en secondes) avant la fermeture des connexions de plugin inutilisées.
Plugins.Redis.Sessions.<SessionName>.Password	Spécifiez le mot de passe de la session nommée pour la connexion à Redis.
Plugins.Redis.Sessions.<SessionName>.TLSConnect	Spécifiez le niveau de chiffrement requis pour les communications de Zabbix agent 2 pour la session nommée.
Plugins.Redis.Sessions.<SessionName>.TLSCAFile	Spécifiez le chemin complet du fichier contenant les certificats des autorités de certification (CA) de niveau supérieur pour la session nommée.
Plugins.Redis.Sessions.<SessionName>.TLSCertFile	Spécifiez le chemin complet du fichier contenant le certificat de l'agent pour la session nommée.
Plugins.Redis.Sessions.<SessionName>.TLSKeyFile	Spécifiez le chemin complet du fichier contenant la clé privée de l'agent pour la session nommée.
Plugins.Redis.Sessions.<SessionName>.Uri	Spécifiez l'URI de la session nommée pour la connexion à Redis.
Plugins.Redis.Sessions.<SessionName>.User	Spécifiez le nom d'utilisateur de la session nommée pour la connexion à Redis.
Plugins.Redis.Timeout	Définissez le délai d'exécution de la requête.

Notez que :

- Les valeurs par défaut reflètent les valeurs par défaut du processus, et non celles des fichiers de configuration fournis.
- Les valeurs prennent en charge les variables d'environnement.
- Zabbix prend en charge les fichiers de configuration uniquement en encodage UTF-8 sans BOM.
- Les commentaires commençant par "#" ne sont pris en charge qu'au début de la ligne.

Détails du paramètre

Plugins.Redis.Default.Password

Spécifiez le mot de passe par défaut pour la connexion à Redis.
 Utilisé si aucune valeur n'est spécifiée dans une clé d'élément ou une session nommée.

Plugins.Redis.Default.TLSConnect

Spécifiez le niveau de chiffrement requis pour les communications de Zabbix agent 2.
 Une valeur vide ou non définie est considérée comme sans TLS.

Plage : `required` - chiffré, non vérifié (uniquement pour les tests), `verify_ca` - chiffré, certificat du serveur, `verify_full` - chiffré, certificat du serveur et nom du serveur vérifié avec SAN

Plugins.Redis.Default.TLSCAFile

Spécifiez le chemin d'accès complet du fichier contenant le ou les certificats d'autorité de certification (CA) de niveau supérieur.

Plugins.Redis.Default.TLSCertFile

Spécifiez le chemin d'accès complet du fichier contenant le certificat de l'agent.

Plugins.Redis.Default.TLSKeyFile

Spécifiez le chemin d'accès complet du fichier contenant la clé privée de l'agent.

Plugins.Redis.Default.Uri

Spécifiez l'URI par défaut pour la connexion à Redis.
 Utilisé si aucune valeur n'est spécifiée dans une clé d'élément ou une session nommée.
 Les identifiants intégrés seront ignorés. Doit correspondre au format URI.
 Schémas pris en charge: `tcp`, `unix`
 Un schéma peut être omis.
 Un port peut être omis (par défaut=6379).

Par défaut: `tcp://localhost:6379`

Exemples:

```
tcp://localhost:6379
localhost
unix:/var/run/redis.sock
```

Plugins.Redis.Default.User

Spécifiez le nom d'utilisateur par défaut pour la connexion à Redis.
 Utilisé si aucune valeur n'est spécifiée dans une clé d'élément ou une session nommée.

Par défaut: `default`

Plugins.Redis.KeepAlive

Définit le temps d'attente maximal (en secondes) avant la fermeture des connexions de plugin inutilisées.

Par défaut: 300
 Plage: 60-900

Plugins.Redis.Sessions.<SessionName>.Password

Spécifiez le mot de passe de la session nommée pour la connexion à Redis.
 <SessionName> - définissez le nom de la session à utiliser dans les clés d'élément.

Plugins.Redis.Sessions.<SessionName>.TLSConnect

Spécifie le niveau de chiffrement requis pour les communications de Zabbix agent 2 pour la session nommée.
 <SessionName> - définit le nom de la session à utiliser dans les clés d'élément.
 Une valeur vide ou non définie est considérée comme l'absence de TLS.

Plage: `required` - chiffré, non vérifié (uniquement pour les tests), `verify_ca` - chiffré, certificat du serveur, `verify_full` - chiffré, certificat du serveur et nom du serveur vérifié avec SAN

Plugins.Redis.Sessions.<SessionName>.TLSCAFile

Spécifiez le chemin complet du fichier contenant le ou les certificats CA de niveau supérieur pour la session nommée.
 <SessionName> - définit le nom de la session à utiliser dans les clés d'élément.

Obligatoire: `yes`, si `Plugins.Redis.Sessions.<SessionName>.TLSConnect` est défini sur `verify_ca` ou `verify_full`

Plugins.Redis.Sessions.<SessionName>.TLSCertFile

Spécifiez le chemin d'accès complet du fichier contenant le certificat de l'agent pour la session nommée.
 <SessionName> - définissez le nom de la session à utiliser dans les clés d'élément.

Plugins.Redis.Sessions.<SessionName>.TLSKeyFile

Spécifiez le chemin d'accès complet du fichier contenant la clé privée de l'agent pour la session nommée.
 <SessionName> - définissez le nom de session à utiliser dans les clés d'élément.

Plugins.Redis.Sessions.<SessionName>.Uri

Spécifiez l'URI de session nommée pour la connexion à Redis.
 <SessionName> - définissez le nom de session à utiliser dans les clés d'élément.
 Les identifiants intégrés seront ignorés. Doit correspondre au format URI.
 Schémas pris en charge: `tcp`, `unix`
 Un schéma peut être omis.
 Un port peut être omis (par défaut=6379).

Par défaut: `tcp://localhost:6379`

Exemples:

```
tcp://localhost:6379
localhost
unix:/var/run/redis.sock
```

Plugins.Redis.Sessions.<SessionName>.User

Spécifiez le nom d'utilisateur de la session nommée pour la connexion à Redis.
 <SessionName> - définissez le nom de la session à utiliser dans les clés d'élément.

Plugins.Redis.Timeout

Définit le délai d'exécution de la requête (la durée, en secondes, d'attente avant de mettre fin à une requête si elle ne se termine pas).

Par défaut: `global timeout`
 Plage: 1-30

Voir aussi

- Description des paramètres de configuration généraux de Zabbix agent 2 : Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- Instructions pour configurer les plugins

Plugin SMART

Vue d'ensemble

Les paramètres de configuration permettent de personnaliser le comportement du plugin SMART.

Il est recommandé de les spécifier dans leur propre fichier de configuration (par exemple, `smart.conf`), plutôt que dans le fichier de configuration de l'agent 2. Utilisez ensuite la directive Include pour inclure ce fichier dans la configuration de Zabbix agent 2.

Tous les paramètres sont **facultatifs**, sauf indication explicite précisant que le paramètre est obligatoire.

Cette page est structurée comme suit :

- Vue d'ensemble des paramètres (pour voir les détails complets d'un paramètre, cliquez sur son nom)
- Détails des paramètres

Parameter	Description
Plugins.Smart.Path	Spécifiez le chemin vers l'exécutable smartctl.
Plugins.Smart.Timeout	Définissez le délai d'attente d'exécution de la requête (la durée, en secondes, pendant laquelle attendre qu'une requête se termine avant de l'interrompre).

Notez que :

- Les valeurs par défaut reflètent les valeurs par défaut du processus, et non celles des fichiers de configuration fournis ;
- Les valeurs prennent en charge les variables d'environnement ;
- Le chemin vers l'exécutable `smartctl` doit être défini soit en ajoutant son répertoire à la variable d'environnement `PATH` du système, soit en configurant `Plugins.Smart.Path` ; cela s'applique à Linux comme à Windows ;
- Zabbix prend en charge uniquement les fichiers de configuration encodés en UTF-8 sans BOM ;
- Les commentaires commençant par "#" ne sont pris en charge qu'au début de la ligne.

Détails du paramètre

Plugins.Smart.Path

Spécifiez le chemin vers l'exécutable smartctl.

Par défaut: `smartctl`

Plugins.Smart.Timeout

Définissez le délai d'exécution de la requête (la durée, en secondes, d'attente de l'exécution d'une requête avant de l'interrompre).

Par défaut: `global timeout<br>` Plage: `1-30`

Voir aussi

- Description des paramètres de configuration généraux de Zabbix agent 2 : Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- Instructions pour configurer les plugins

Protocole du plugin Zabbix agent 2

Le protocole de Zabbix agent 2 est basé sur le code, la taille et le modèle de données.

Code

Type	Size	Comments
Byte	4	Type de charge utile, actuellement seul JSON est pris en charge.

Taille

Type	Taille	Commentaires
Byte	4	Taille de la charge utile actuelle en octets.

Données de charge utile

Type	Taille	Commentaires
Byte	Defini par le champ *Size*	Données au format JSON.

Définition des données de charge utile

Données communes

Ces paramètres sont présents dans toutes les requêtes/réponses :

Name	Type	Comments
id	uint32	Pour les requêtes - l'identifiant incrémental utilisé pour associer les requêtes aux réponses. Unique dans une direction de requête (c'est-à-dire de l'agent vers le plugin ou du plugin vers l'agent). Pour les réponses - l'ID de la requête correspondante.
type	uint32	Le type de requête.

Requête de journalisation

Requête envoyée par un plugin pour écrire un message de journal dans le fichier journal de l'agent.

direction	plugin → agent
response	no

Paramètres spécifiques aux requêtes de journalisation :

Name	Type	Comments
severity	uint32	Le niveau de gravité du message (niveau de journalisation).
message	string	Le message à journaliser.

Exemple :

```
{"id":0,"type":1,"severity":3,"message":"message"}
```

Requête d'enregistrement

Requête envoyée par l'agent pendant la phase de démarrage de l'agent afin d'obtenir les métriques fournies pour enregistrer un plugin.

direction	agent → plugin
response	yes

Paramètres spécifiques aux requêtes d'enregistrement :

Name	Type	Comments
version	string	La version du protocole <major>.<minor>

Exemple :

```
{"id":1,"type":2,"version":"1.0"}
```

Réponse d'enregistrement

Réponse du plugin à la requête d'enregistrement.

direction	plugin → agent
response	n/a

Paramètres spécifiques aux réponses d'enregistrement :

Name	Type	Comments
name	string	Le nom du plugin.

Name	Type	Comments
metrics	array of strings (optional)	Les métriques avec leurs descriptions telles qu'utilisées dans le plugin. Retourne RegisterMetrics(). Absent si une erreur est renvoyée.
interfaces	uint32 (optional)	Le masque de bits des interfaces prises en charge par le plugin. Absent si une erreur est renvoyée.
error	string (optional)	Un message d'erreur renvoyé si un plugin ne peut pas être démarré. Absent si des métriques sont renvoyées.

Exemples :

```
{"id":2,"type":3,"metrics":["external.test", "External exporter Test."], "interfaces": 4}
```

ou

```
{"id":2,"type":3,"error":"error message"}
```

Demande de démarrage

Une demande d'exécution de la fonction Start de l'interface Runner.

direction	agent → plugin
response	no

La demande ne comporte pas de paramètres spécifiques, elle contient uniquement les paramètres de données communes.

Exemple :

```
{"id":3,"type":4}
```

Terminer la requête

Une requête envoyée par l'agent pour arrêter un plugin.

direction	agent → plugin
response	no

La requête ne possède pas de paramètres spécifiques, elle contient uniquement les paramètres de données communes.

Exemple :

```
{"id":3,"type":5}
```

Demande d'export

Une demande d'exécution de la fonction Export de l'interface Exporter.

direction	agent → plugin
response	no

Paramètres spécifiques aux demandes d'export :

Name	Type	Comments
key	string	La clé du plugin.
parameters	array of strings (optional)	Les paramètres de la fonction Export.

Exemple :

```
{"id":4,"type":6,"key":"test.key","parameters":["foo","bar"]}
```

Réponse d'export

Réponse de la fonction Export de l'interface Exporter.

direction	plugin → agent
response	n/a

Paramètres spécifiques aux réponses d'export :

Name	Type	Comments
value	string (optional)	Valeur de réponse de la fonction Export. Absente si une erreur est renvoyée.
error	string (optional)	Message d'erreur si la fonction Export ne s'est pas exécutée avec succès. Absent si une valeur est renvoyée.

Exemples :

```
{"id":5,"type":7,"value":"response"}
```

ou

```
{"id":5,"type":7,"error":"error message"}
```

Configurer la requête

Une requête pour exécuter la fonction *Configure* de l'interface *Configurator*.

direction	agent → plugin
response	n/a

Paramètres spécifiques aux requêtes *Configure* :

Name	Type	Comments
global_options	JSON object	Objet JSON contenant les options de configuration globales de l'agent.
private_options	JSON object (optional)	Objet JSON contenant les options de configuration privées du plugin, si fournies.

Exemple :

```
{"id":6,"type":8,"global_options":{...},"private_options":{...}}
```

Valider la requête

Une requête pour exécuter la fonction *Validate* de l'interface *Configurator*.

direction	agent → plugin
response	yes

Paramètres spécifiques aux requêtes *Validate* :

Name	Type	Comments
private_options	objet JSON (facultatif)	Objet JSON contenant les options de configuration privées du plugin, si elles sont fournies.

Exemple :

```
{"id":7,"type":9,"private_options":{...}}
```

Valider la réponse

Réponse de la fonction *Validate* de l'interface *Configurator*.

direction	plugin → agent
response	n/a

Paramètres spécifiques aux réponses *Validate* :

Name	Type	Comments
error	string (optional)	Un message d'erreur renvoyé si la fonction Validate n'est pas exécutée avec succès. Absent si elle est exécutée avec succès.

Exemple :

```
{"id":8,"type":10}
```

ou

```
{"id":8,"type":10,"error":"error message"}
```

Java gateway

Présentation générale Zabbix Java gateway peut être installé à partir du code source ou des paquets.

La prise en charge native de la supervision des applications JMX existe sous la forme d'un daemon Zabbix appelé « Zabbix Java gateway ». Zabbix Java gateway est un daemon écrit en Java. Pour connaître la valeur d'un compteur JMX particulier sur un hôte, le serveur Zabbix interroge Zabbix Java gateway, qui utilise l'API de gestion JMX pour interroger à distance l'application concernée. L'application ne nécessite l'installation d'aucun logiciel supplémentaire ; elle doit simplement être démarrée avec l'option `-Dcom.sun.management.jmxremote` sur la ligne de commande.

Java gateway accepte les connexions entrantes du serveur Zabbix ou du proxy et peut uniquement être utilisé comme « proxy passif ». Contrairement au proxy Zabbix, il peut également être utilisé depuis un proxy Zabbix (les proxies Zabbix ne peuvent pas être chaînés). L'accès à chaque Java gateway est configuré directement dans le fichier de configuration du serveur Zabbix ou du proxy Zabbix ; par conséquent, un seul Java gateway peut être configuré par serveur Zabbix ou proxy Zabbix. Si un hôte possède des éléments de type *JMX agent* et des éléments d'autres types, seuls les éléments *JMX agent* seront transmis à Java gateway pour être récupérés.

Lorsqu'un élément doit être mis à jour via Java gateway, le serveur ou le proxy Zabbix se connecte à Java gateway et demande la valeur, que Java gateway récupère à son tour et renvoie au serveur ou au proxy. Ainsi, Java gateway ne met aucune valeur en cache.

Le serveur ou le proxy Zabbix dispose d'un type spécifique de processus qui se connectent à Java gateway, contrôlé par l'option `StartJavaPollers`. En interne, Java gateway démarre plusieurs threads, contrôlés par l'option `START_POLLERS`. Du côté du serveur, si une connexion prend plus de « Timeout » secondes, elle est interrompue, mais Java gateway peut toujours être occupé à récupérer la valeur du compteur JMX. Pour résoudre ce problème, Java gateway fournit l'option `TIMEOUT`, qui permet de définir un délai d'expiration pour les opérations réseau JMX.

Voir également :

- Paramètres de configuration : Zabbix Java gateway
- Configuration à partir des paquets Debian/Ubuntu
- Configuration à partir des paquets RHEL
- Configuration à partir des sources

Le serveur ou le proxy Zabbix essaiera de regrouper autant que possible les requêtes vers une même cible JMX (selon les intervalles des éléments) et de les envoyer au Java gateway dans une seule connexion pour de meilleures performances.

Il est recommandé que `StartJavaPollers` soit inférieur ou égal à `START_POLLERS`; sinon, il peut arriver qu'aucun thread ne soit disponible dans le Java gateway pour traiter les requêtes entrantes. Dans ce cas, le Java gateway utilise `ThreadPoolExecutor.CallerRunsPolicy`, ce qui signifie que le thread principal traitera la requête entrante et n'acceptera temporairement aucune nouvelle requête.

Si vous essayez de surveiller des applications Java basées sur Wildfly avec le Java gateway Zabbix, veuillez installer la dernière version de `jboss-client.jar` disponible sur la page de téléchargement de Wildfly.

Paramètres de configuration: Zabbix Java gateway

Si vous utilisez les scripts `startup.sh` et `shutdown.sh` pour démarrer Zabbix Java gateway, vous pouvez spécifier les paramètres de configuration nécessaires dans le fichier `settings.sh`. Les scripts de démarrage et d'arrêt chargent le fichier de paramètres et se chargent de convertir les variables shell (listées dans la première colonne) en propriétés Java (listées dans la deuxième colonne).

Si vous démarrez Zabbix Java gateway manuellement en exécutant `java` directement, vous spécifiez les propriétés Java correspondantes sur la ligne de commande.

Variable	Property	Mandatory	Range	Default	Description
LISTEN_IP	zabbix.listenIP	no		0.0.0.0	Adresse IP sur laquelle écouter.
LISTEN_PORT	zabbix.listenPort	no	1024-32767	10052	Port sur lequel écouter.
PID_FILE	zabbix.pidFile	no		/tmp/zabbix_java.pid	Nom du fichier PID. S'il est omis, Zabbix Java Gateway est démarré comme application console.
PROPERTIES_FILE	zabbix.propertiesFile	no			Nom du fichier de propriétés. Peut être utilisé pour définir des propriétés supplémentaires au format clé-valeur, de manière à ce qu'elles ne soient pas visibles sur la ligne de commande, ou pour écraser celles qui existent déjà. Par exemple: "javax.net.ssl.trustStorePassword=<p
START_POLLERS	zabbix.startPollers	no	1-1000	5	Nombre de threads de travail à démarrer.
TIMEOUT	zabbix.timeout	no	1-30	3	Durée d'attente pour les opérations réseau (en secondes).

Warning:
Le port 10052 n'est pas enregistré auprès de l'IANA.

Configuration à partir des paquets Debian/Ubuntu

Aperçu

Si installé à partir de paquets, les informations suivantes vous aideront à configurer le Java gateway.

Configuration et démarrage de Java gateway

La configuration de Java gateway peut être ajustée dans le fichier :

`/etc/zabbix/zabbix_java_gateway.conf`

Pour plus de détails, consultez les paramètres de configuration de Zabbix Java gateway.

Pour démarrer Zabbix Java gateway :

```
systemctl restart zabbix-java-gateway
```

Pour démarrer automatiquement Zabbix Java gateway au démarrage :

```
systemctl enable zabbix-java-gateway
```

Configuration du serveur pour une utilisation avec Java gateway

Une fois Java gateway démarrée et opérationnelle, vous devez indiquer au serveur Zabbix où trouver Zabbix Java gateway. Cela se fait en spécifiant les paramètres `JavaGateway` et `JavaGatewayPort` dans le fichier de configuration du serveur. Si l'hôte sur lequel l'application JMX s'exécute est surveillé par un proxy Zabbix, vous devez alors spécifier les paramètres de connexion dans le fichier de configuration du proxy à la place.

```
JavaGateway=192.168.3.14
JavaGatewayPort=10052
```

Par défaut, le serveur ne démarre aucun processus lié à la surveillance JMX. Si vous souhaitez toutefois l'utiliser, vous devez spécifier le nombre d'instances pré-forkées de collecteurs Java. Vous procédez de la même manière que pour spécifier les collecteurs et les trappeurs classiques.

```
StartJavaPollers=5
```

N'oubliez pas de redémarrer le serveur ou le proxy une fois la configuration terminée.

Débogage de Java gateway

Le fichier journal de Java gateway de Zabbix est :

```
/var/log/zabbix/zabbix_java_gateway.log
```

Si vous souhaitez augmenter le niveau de journalisation, modifiez le fichier :

```
/etc/zabbix/zabbix_java_gateway_logback.xml
```

et remplacez `info` dans `level="info"` par debug ou même `trace` (pour un dépannage approfondi) :

```
<configuration scan="true" scanPeriod="15 seconds">
[...]
      <root level="info">
              <appender-ref ref="FILE" />
      </root>

</configuration>
```

Surveillance JMX

Consultez la page Surveillance JMX pour plus de détails.

Configuration à partir des paquets RHEL

Vue d'ensemble

Si installé à partir de paquets, les informations suivantes vous aideront à configurer le Java gateway.

Configuration et démarrage de Java gateway

Les paramètres de configuration de Zabbix Java gateway peuvent être ajustés dans le fichier :

```
/etc/zabbix/zabbix_java_gateway.conf
```

Pour plus de détails, consultez les paramètres de configuration de Zabbix Java gateway.

Pour démarrer Zabbix Java gateway :

```
systemctl restart zabbix-java-gateway
```

Pour démarrer automatiquement Zabbix Java gateway au démarrage :

RHEL 7 et versions ultérieures :

```
systemctl enable zabbix-java-gateway
```

RHEL antérieur à la version 7 :

```
chkconfig --level 12345 zabbix-java-gateway on
```

Configuration du serveur pour une utilisation avec Java gateway

Une fois Java gateway démarrée et opérationnelle, vous devez indiquer au serveur Zabbix où trouver Zabbix Java gateway. Cela se fait en spécifiant les paramètres `JavaGateway` et `JavaGatewayPort` dans le fichier de configuration du serveur. Si l'hôte sur

lequel l'application JMX s'exécute est surveillé par un proxy Zabbix, vous devez alors spécifier les paramètres de connexion dans le fichier de configuration du proxy à la place.

```
JavaGateway=192.168.3.14
JavaGatewayPort=10052
```

Par défaut, le serveur ne démarre aucun processus lié à la surveillance JMX. Si vous souhaitez toutefois l'utiliser, vous devez spécifier le nombre d'instances pré-forkées de collecteurs Java. Vous procédez de la même manière que pour spécifier les collecteurs et les trappeurs classiques.

```
StartJavaPollers=5
```

N'oubliez pas de redémarrer le serveur ou le proxy une fois la configuration terminée.

Débogage de Java gateway

Le fichier journal de Java gateway de Zabbix est :

```
/var/log/zabbix/zabbix_java_gateway.log
```

Si vous souhaitez augmenter le niveau de journalisation, modifiez le fichier :

```
/etc/zabbix/zabbix_java_gateway_logback.xml
```

et remplacez `info` dans `level="info"` par `debug` ou même `trace` (pour un dépannage approfondi) :

```
<configuration scan="true" scanPeriod="15 seconds">
[...]
      <root level="info">
              <appender-ref ref="FILE" />
      </root>

</configuration>
```

Surveillance JMX

Consultez la page Surveillance JMX pour plus de détails.

Installation à partir des sources

Vue d'ensemble

Si vous avez installé Zabbix à partir des sources, les informations suivantes vous aideront à configurer le Java gateway de Zabbix.

Présentation des fichiers

Si vous avez obtenu Java gateway à partir des sources, vous devriez avoir obtenu un ensemble de scripts shell, de fichiers JAR et de fichiers de configuration sous `$PREFIX/sbin/zabbix_java`. Le rôle de ces fichiers est résumé ci-dessous.

```
bin/zabbix-java-gateway-$VERSION.jar
```

Fichier JAR de Java gateway lui-même.

```
lib/logback-core-1.5.25.jar
lib/logback-classic-1.5.25.jar
lib/slf4j-api-2.0.16.jar
lib/android-json-4.3_r3.1.jar
```

Dépendances de Java gateway : bibliothèques Logback, SLF4J et Android JSON.

```
lib/logback.xml
lib/logback-console.xml
```

Fichiers de configuration pour Logback.

```
shutdown.sh
startup.sh
```

Scripts pratiques pour démarrer et arrêter Java gateway.

```
settings.sh
```

Fichier de configuration chargé par les scripts de démarrage et d'arrêt ci-dessus.

Configuration et exécution de Java gateway

Par défaut, Java gateway écoute sur le port 10052. Si vous prévoyez d'exécuter Java gateway sur un autre port, vous pouvez le spécifier dans le script `settings.sh`. Consultez la description du fichier de configuration de Java gateway pour savoir comment définir ce paramètre et d'autres options.

> **Warning:**
> Le port 10052 n'est pas enregistré auprès de l'IANA.

Une fois les paramètres configurés, vous pouvez démarrer Java gateway en exécutant le script `startup.sh` :

```
./startup.sh
```

De même, lorsque vous n'avez plus besoin de Java gateway, exécutez le script `shutdown.sh` pour l'arrêter :

```
./shutdown.sh
```

Notez que, contrairement au serveur ou au proxy, Java gateway est léger et n'a pas besoin de base de données.

Configuration du serveur pour une utilisation avec Java gateway

Une fois Java gateway démarrée et opérationnelle, vous devez indiquer au serveur Zabbix où trouver Zabbix Java gateway. Cela se fait en spécifiant les paramètres `JavaGateway` et `JavaGatewayPort` dans le fichier de configuration du serveur. Si l'hôte sur lequel l'application JMX s'exécute est surveillé par un proxy Zabbix, vous devez alors spécifier les paramètres de connexion dans le fichier de configuration du proxy à la place.

```
JavaGateway=192.168.3.14
JavaGatewayPort=10052
```

Par défaut, le serveur ne démarre aucun processus lié à la surveillance JMX. Si vous souhaitez toutefois l'utiliser, vous devez spécifier le nombre d'instances pré-forkées de collecteurs Java. Vous procédez de la même manière que pour spécifier les collecteurs et les trappeurs classiques.

```
StartJavaPollers=5
```

N'oubliez pas de redémarrer le serveur ou le proxy une fois la configuration terminée.

Débogage de Java gateway

En cas de problème avec Java gateway ou si le message d'erreur que vous voyez concernant un élément dans l'interface n'est pas suffisamment explicite, il peut être utile de consulter le fichier journal de Java gateway.

Par défaut, Java gateway enregistre ses activités dans le fichier `/tmp/zabbix_java.log` avec le niveau de journalisation "info". Parfois, ces informations ne suffisent pas et il est nécessaire d'obtenir des informations au niveau de journalisation "debug". Pour augmenter le niveau de journalisation, modifiez le fichier `lib/logback.xml` et changez l'attribut `level` de la balise `<root>` en `debug` :

```
<root level="debug">
  <appender-ref ref="FILE" />
</root>
```

Notez que, contrairement au serveur Zabbix ou au proxy Zabbix, il n'est pas nécessaire de redémarrer Zabbix Java gateway après avoir modifié le fichier `logback.xml` - les modifications apportées à `logback.xml` seront prises en compte automatiquement. Une fois le débogage terminé, vous pouvez remettre le niveau de journalisation à "info".

Si vous souhaitez journaliser dans un autre fichier ou dans un support complètement différent, comme une base de données, adaptez le fichier `logback.xml` à vos besoins. Consultez le manuel Logback pour plus de détails.

Parfois, à des fins de débogage, il est utile de démarrer Java gateway comme application console plutôt que comme démon. Pour ce faire, commentez la variable `PID_FILE` dans `settings.sh`. Si `PID_FILE` est omise, le script `startup.sh` démarre Java gateway comme application console et demande à Logback d'utiliser le fichier `lib/logback-console.xml` à la place, ce qui permet non seulement de journaliser dans la console, mais active également le niveau de journalisation "debug".

Enfin, notez que, puisque Java gateway utilise SLF4J pour la journalisation, vous pouvez remplacer Logback par le framework de votre choix en plaçant un fichier JAR approprié dans le répertoire lib. Consultez le manuel SLF4J pour plus de détails.

Surveillance JMX

Consultez la page Surveillance JMX pour plus de détails.

Expéditeur

Présentation générale Zabbix sender est un utilitaire en ligne de commande qui peut être utilisé pour envoyer des données de performance au serveur ou au proxy Zabbix afin qu'elles soient traitées.

L'utilitaire est généralement utilisé dans des scripts utilisateur exécutés sur une longue durée pour envoyer périodiquement des données de disponibilité et de performance.

Pour envoyer directement les résultats au serveur ou au proxy Zabbix, un élément de type *Zabbix trapper* doit être configuré.

Voir également :

- Protocole Zabbix sender
- Bibliothèque de liens dynamiques Zabbix sender pour Windows
- Bibliothèque Python pour Zabbix, qui dispose d'une fonctionnalité intégrée permettant de fonctionner comme Zabbix sender

Exécution de Zabbix sender Exemple d'exécution de Zabbix sender UNIX :

```
cd bin
./zabbix_sender -z zabbix -s "Linux DB3" -k db.connections -o 43
```

où :

- `z` - hôte du serveur Zabbix (l'adresse IP peut également être utilisée) ;
- `s` - nom technique de l'hôte surveillé (tel qu'il est enregistré dans l'interface Zabbix) ;
- `k` - clé de l'élément ;
- `o` - valeur à envoyer.

> **Attention:**
> Les options contenant des espaces doivent être placées entre guillemets doubles.

Zabbix sender peut être utilisé pour envoyer plusieurs valeurs à partir d'un fichier d'entrée. Consultez la page de manuel de Zabbix sender pour plus d'informations.

Si un fichier de configuration est spécifié, Zabbix sender utilise toutes les adresses définies dans le paramètre de configuration `ServerActive` de l'agent pour envoyer les données. Si l'envoi vers une adresse échoue, sender tente d'envoyer les données vers les autres adresses. Si l'envoi d'un lot de données échoue vers une adresse, les lots suivants ne sont pas envoyés à cette adresse.

Zabbix sender accepte les chaînes encodées en UTF-8 (pour les systèmes de type Unix et Windows) sans marque d'ordre des octets (BOM) au début du fichier.

Sous Windows, Zabbix sender peut être exécuté de manière similaire :

```
zabbix_sender.exe [options]
```

Les scénarios d'envoi en temps réel de `zabbix_sender` regroupent plusieurs valeurs qui lui sont transmises successivement et les envoient au serveur au moyen d'une seule connexion. Une valeur dont l'intervalle avec la valeur précédente ne dépasse pas 0,2 seconde peut être placée dans la même pile, mais le temps d'interrogation maximal reste d'une seconde.

> **Note:**
> Zabbix sender s'arrête si une entrée de paramètre non valide (ne respectant pas la notation `parameter=value`) est présente dans le fichier de configuration spécifié.

Exécution de Zabbix sender avec la découverte bas niveau Exemple d'exécution de Zabbix sender pour envoyer une valeur au format JSON pour la découverte bas niveau :

```
./zabbix_sender -z 192.168.1.113 -s "Zabbix server" -k trapper.discovery.item -o '[{"{#FSNAME}":"/","{#FSI
```

Pour que cela fonctionne, la règle de découverte bas niveau doit avoir le type d'élément *Zabbix trapper* (dans cet exemple, avec la clé `trapper.discovery.item`).

Protocole Zabbix sender

Le serveur Zabbix et le proxy Zabbix utilisent un protocole de communication basé sur JSON pour recevoir des données de Zabbix sender. Les données peuvent être reçues à l'aide d'un élément trapper, ou d'un élément agent HTTP avec la capture activée.

Les messages de requête et de réponse doivent commencer par l'en-tête et la longueur des données.

Requête Zabbix sender

```
{
    "request": "données sender",
    "data": [
        {
            "host": "<hostname>",
            "key": "trap",
            "value": "test value"
        }
    ]
}
```

Réponse du serveur Zabbix

```
{
    "response": "success",
    "info": "processed: 1; failed: 0; total: 1; seconds spent: 0.060753"
}
```

Requête Zabbix sender avec un horodatage

Sinon, Zabbix sender peut envoyer une requête avec un horodatage et des nanosecondes.

```
{
    "request": "sender data",
    "data": [
        {
            "host": "<hostname>",
            "key": "trap",
            "value": "test value",
            "clock": 1516710794,
            "ns": 592397170
        },
        {
            "host": "<hostname>",
            "key": "trap",
            "value": "test value",
            "clock": 1516710795,
            "ns": 192399456
        }
    ],
    "clock": 1516712029,
    "ns": 873386094
}
```

Réponse du serveur Zabbix

```
{
    "response": "success",
    "info": "processed: 2; failed: 0; total: 2; seconds spent: 0.060904"
}
```

Bibliothèque de liaison dynamique Zabbix sender pour Windows

Dans un environnement Windows, les applications peuvent envoyer des données au serveur/proxy Zabbix en utilisant la bibliothèque de liaison dynamique Zabbix sender (`zabbix_sender.dll`) au lieu de devoir lancer un processus externe (`zabbix_sender.exe`).

zabbix_sender.h et zabbix_sender.lib sont requis pour compiler des applications utilisateur avec zabbix_sender.dll.

Obtention

Il existe deux façons d'obtenir zabbix_sender.dll.

1. Téléchargez les fichiers zabbix_sender.h, zabbix_sender.lib et zabbix_sender.dll sous forme d'archive ZIP.

Lors du choix des options de téléchargement, assurez-vous de sélectionner "No encryption" sous *Encryption* et "Archive" sous *Packaging*. Téléchargez ensuite Zabbix agent (et non Zabbix agent 2).

Les fichiers zabbix_sender.h, zabbix_sender.lib et zabbix_sender.dll se trouveront dans l'archive ZIP téléchargée, dans le répertoire `bin\dev`. Décompressez les fichiers à l'emplacement souhaité.

2. Compilez zabbix_sender.dll à partir des sources (voir les instructions).

La bibliothèque de liaison dynamique avec les fichiers de développement se trouvera dans le répertoire `bin\winXX\dev`. Pour l'utiliser, incluez le fichier d'en-tête zabbix_sender.h et liez la bibliothèque zabbix_sender.lib.

Voir aussi

- exemple d'un utilitaire Zabbix sender simple implémenté avec la bibliothèque de liaison dynamique Zabbix sender pour illustrer l'utilisation de la bibliothèque ;
- fichier zabbix_sender.h pour les fonctions de l'interface de la bibliothèque de liaison dynamique Zabbix sender. Ce fichier contient une documentation expliquant l'objectif de chaque fonction de l'interface, ses arguments et sa valeur de retour.

Obtenir

Vue d'ensemble Zabbix get est un utilitaire en ligne de commande qui peut être utilisé pour communiquer avec l'agent Zabbix et récupérer les informations requises depuis l'agent.

L'utilitaire est généralement utilisé pour le dépannage des agents Zabbix.

Voir aussi la bibliothèque Python pour Zabbix, qui dispose d'une fonctionnalité intégrée permettant de fonctionner comme Zabbix get.

Exécution de Zabbix get Voici un exemple d'exécution de Zabbix get sous UNIX pour obtenir la valeur de charge du processeur à partir de l'agent :

```
cd bin
./zabbix_get -s 127.0.0.1 -p 10050 -k system.cpu.load[all,avg1]
```

Autre exemple d'exécution de Zabbix get pour capturer une chaîne depuis un site web :

```
cd bin
./zabbix_get -s 192.168.1.1 -p 10050 -k "web.page.regexp[www.example.com,,,\"USA: ([a-zA-Z0-9.-]+)\",,\1]"
```

Notez que la clé d'élément contient ici un espace, donc des guillemets sont utilisés pour indiquer la clé d'élément au shell. Les guillemets ne font pas partie de la clé d'élément ; ils seront supprimés par le shell et ne seront pas transmis à l'agent Zabbix.

Si une clé d'élément n'est pas prise en charge, Zabbix get renverra le code de sortie 1.

Zabbix get accepte les paramètres de ligne de commande suivants :

```
-s --host <host name or IP>             Spécifie le nom d'hôte ou l'adresse IP d'un hôte
-p --port <port number>                 Spécifie le numéro de port de l'agent exécuté sur l'hôte (par défau
-I --source-address <IP address>        Spécifie l'adresse IP source
-t --timeout <seconds>                  Spécifie le délai d'attente. Plage valide : 1-600 secondes (par dé
-k --key <item key>                     Spécifie la clé de l'élément dont il faut récupérer la valeur
-P --protocol <value>                   Protocole utilisé pour communiquer avec l'agent. Valeurs :
                                            auto - se connecte en utilisant le protocole JSON, avec repli
                                            json - se connecte en utilisant le protocole JSON
                                            plaintext - se connecte en utilisant le protocole texte brut,
-h --help                               Affiche ce message d'aide
-V --version                            Affiche le numéro de version

--tls-connect <value>                   Mode de connexion à l'agent. Valeurs :
                                            unencrypted - connexion sans chiffrement (par défaut)
                                            psk - connexion en utilisant TLS et une clé prépartagée
                                            cert - connexion en utilisant TLS et un certificat
--tls-ca-file <CA file>                 Chemin complet d'un fichier contenant les certificats de l'AC racin
--tls-crl-file <CRL file>               Chemin complet d'un fichier contenant les certificats révoqués
--tls-agent-cert-issuer <cert issuer>   Émetteur de certificat d'agent autorisé
--tls-agent-cert-subject <cert subject> Sujet de certificat d'agent autorisé
--tls-cert-file <cert file>             Chemin complet d'un fichier contenant le certificat ou la chaîne de
--tls-key-file <key file>               Chemin complet d'un fichier contenant la clé privée
--tls-psk-identity <PSK-identity>       Chaîne unique, sensible à la casse, utilisée pour identifier la cl
```

```
--tls-psk-file <PSK-file>               Chemin complet d'un fichier contenant la clé prépartagée
--tls-cipher13 <cipher-string>          Chaîne de chiffrement pour OpenSSL 1.1.1 ou version ultérieure pou
--tls-cipher <cipher-string>            Chaîne de priorité GnuTLS (pour TLS 1.2 et versions ultérieures) c
```

Voir aussi la page de manuel Zabbix get pour plus d'informations.

Zabbix get sous Windows peut être exécuté de manière similaire :

```
zabbix_get.exe [options]
```

JS

Vue d'ensemble zabbix_js est un utilitaire en ligne de commande qui peut être utilisé pour tester des scripts intégrés.

Cet utilitaire exécutera un script utilisateur avec un paramètre de chaîne et affichera le résultat. Les scripts sont exécutés à l'aide du moteur de script Zabbix intégré.

En cas d'erreurs de compilation ou d'exécution, zabbix_js affichera l'erreur sur stderr et quittera avec le code 1.

```
zabbix_js -s script-file -p input-param [-l log-level] [-t timeout]
zabbix_js -s script-file -i input-file [-l log-level] [-t timeout]
zabbix_js -h
zabbix_js -V
```

Utilisation zabbix_js accepte les paramètres de ligne de commande suivants :

```
-s, --script script-file          Spécifie le nom du fichier du script à exécuter
                                  Si "-" est spécifié comme nom de fichier, le script sera lu depuis stdin
-i, --input input-file            Spécifie le nom du fichier contenant les informations d'entrée
                                  Le contenu sera transmis au script comme valeur
                                  Mutuellement exclusif avec -p
                                  Si "-" est spécifié comme nom de fichier, le script sera lu depuis stdin
-p, --param input-param           Spécifie le paramètre d'entrée : la variable à transmettre au script com
                                  Mutuellement exclusif avec -i
-l, --loglevel log-level          Spécifie le niveau de journalisation
                                  Plage : 0-5
-t, --timeout timeout             Spécifie le délai d'attente en secondes
                                  Plage valide : 1-600 secondes (par défaut : 10 secondes)
-h, --help                        Affiche les informations d'aide
-V, --version                     Affiche le numéro de version
-w <webdriver url>                Active la surveillance du navigateur
```

Exemples : Exemple 1 : Exécution d'un script avec une variable contenant une valeur

Le script (script-file.js) :

```
return value;
```

Utilisation :

```
zabbix_js -s script-file.js -p example_value
```

Sortie : example_value

Exemple 2 : Exécution d'un script avec un fichier contenant des informations d'entrée

Le fichier contenant le script (script-file.js) :

```
return value;
```

Le fichier contenant les informations d'entrée (example.txt) :

```
Example of input information from the file
```

Utilisation :

```
zabbix_js -s script-file.js -i example.txt
```

Sortie : `Example of input information from the file`

Exemple 3 : Exécution d'un script avec une entrée lue depuis stdin

Le fichier contenant le script (`script-file.js`) :

```
return value;
```

Utilisation :

```
zabbix_js -s script-file.js -i -
```

Entrée (stdin) :

```
Example of input from stdin
```

Sortie : `Example of input from stdin`

Service web

Vue d'ensemble Le service web Zabbix est un processus utilisé pour la communication avec des services web externes. Actuellement, le service web Zabbix est utilisé pour générer et envoyer des rapports planifiés, avec l'objectif d'ajouter d'autres fonctionnalités à l'avenir.

Le serveur Zabbix se connecte au service web via HTTP(S). Le service web Zabbix nécessite que Google Chrome soit installé sur le même hôte ; sur certaines distributions, le service peut également fonctionner avec Chromium (voir problèmes connus).

Installation Le paquet officiel zabbix-web-service est disponible dans le dépôt Zabbix.

Pour compiler le service web Zabbix à partir des sources, spécifiez l'option de configuration `--enable-webservice`.

Pour configurer le service web Zabbix, mettez à jour les paramètres du fichier de configuration du service web.

> **Attention:**
> Il est fortement recommandé de configurer le chiffrement entre le serveur Zabbix et le service web Zabbix à l'aide de certificats. Par défaut, les données transmises entre le serveur Zabbix et le service web Zabbix ne sont pas chiffrées, ce qui peut entraîner un accès non autorisé.

Paramètres de configuration: Zabbix web service

Vue d'ensemble

Le service web Zabbix est un processus utilisé pour la communication avec des services web externes.

Les paramètres pris en charge par le fichier de configuration du service web Zabbix (`zabbix_web_service.conf`) sont répertoriés dans cette section.

Les paramètres sont listés sans informations supplémentaires. Cliquez sur le paramètre pour voir les détails complets.

Parameter	Description
AllowedIP	Une liste d'adresses IP séparées par des virgules, éventuellement en notation CIDR, ou des noms DNS des serveurs Zabbix et des proxies Zabbix.
DebugLevel	Le niveau de débogage.
IgnoreURLCertErrors	Spécifie la gestion des erreurs de validation du certificat TLS lors de l'accès à l'URL de l'interface.
Include	Vous pouvez inclure des fichiers individuels ou tous les fichiers d'un répertoire dans le fichier de configuration.
ListenPort	Le service écoutera sur ce port les connexions provenant du serveur.
LogFile	Le nom du fichier journal.
LogFileSize	La taille maximale du fichier journal.
LogType	Le type de sortie du journal.
Timeout	Le temps maximal (en secondes) consacré au formatage du rapport PDF d'un tableau de bord.
TLSAccept	Les connexions entrantes à accepter.

Parameter	Description
TLSCAFile	Le chemin complet d'un fichier contenant les certificats des AC racines pour la vérification du certificat du pair, utilisé pour les communications chiffrées entre les composants Zabbix.
TLSCertFile	Le chemin complet d'un fichier contenant le certificat du service ou la chaîne de certificats, utilisé pour les communications chiffrées entre les composants Zabbix.
TLSKeyFile	Le chemin complet d'un fichier contenant la clé privée du service, utilisé pour les communications chiffrées entre les composants Zabbix.

Tous les paramètres sont facultatifs, sauf indication explicite qu'un paramètre est obligatoire.

Notez que :

- Les valeurs par défaut reflètent les valeurs par défaut du processus, et non celles des fichiers de configuration fournis ;
- Les valeurs prennent en charge les variables d'environnement ;
- Zabbix prend en charge les fichiers de configuration uniquement en encodage UTF-8 sans BOM ;
- Les commentaires commençant par # ne sont pris en charge qu'au début de la ligne.

Détails du paramètre

AllowedIP

Une liste d'adresses IP séparées par des virgules, éventuellement en notation CIDR, ou de noms DNS des serveurs Zabbix et des proxies Zabbix. Les connexions entrantes ne seront acceptées que depuis les hôtes répertoriés ici.
Si la prise en charge d'IPv6 est activée, alors `127.0.0.1`, `::127.0.0.1`, `::ffff:127.0.0.1` sont traités de la même manière et `::/0` autorisera toute adresse IPv4 ou IPv6. `0.0.0.0/0` peut être utilisé pour autoriser toute adresse IPv4.

Exemple :

```
127.0.0.1,192.168.1.0/24,::1,2001:db8::/32,zabbix.example.com
```

Obligatoire : oui

DebugLevel

Spécifiez le niveau de débogage : - *0* - informations de base sur le démarrage et l'arrêt des processus Zabbix ; - *1* - informations critiques ; - *2* - informations d'erreur ; - *3* - avertissements ; - *4* - pour le débogage (génère beaucoup d'informations) ; - *5* - débogage étendu (génère encore plus d'informations).

Par défaut : 3
 Plage : 0-5

IgnoreURLCertErrors

Spécifie la gestion des erreurs de validation du certificat TLS lors de l'accès à l'URL de l'interface :
0 - ne pas ignorer les erreurs de certificat ;
1 - ignorer les erreurs de certificat.

Par défaut : 0
 Plage : 0-1

Inclure

Spécifiez un répertoire ou des fichiers individuels à inclure dans le fichier de configuration.
 Lors de l'installation de Zabbix, ce paramètre est défini sur `/usr/local/etc`, sauf si vous le modifiez au moment de la compilation.

Règles générales pour le paramètre `Include` :

- Le chemin peut être relatif à l'emplacement du fichier `zabbix_web_service.conf`.
- Vous pouvez spécifier plusieurs paramètres `Include` dans un même fichier de configuration.
- Zabbix traite le paramètre `Include` dès qu'il le lit, donc envisagez de le placer à la fin du fichier de configuration.

Règles pour les fichiers inclus :

- Chaque fichier doit être lisible.
- Chaque fichier doit avoir une syntaxe correcte. Sinon, le composant ne démarrera pas.

Règles pour l'inclusion de répertoires ou de modèles de fichiers :

- Ne supposez aucun ordre d'inclusion (par exemple, les fichiers ne sont pas inclus par ordre alphabétique).
- Ne définissez pas le même paramètre dans plus d'un fichier inclus (par exemple, pour remplacer un réglage général par un réglage spécifique), car vous ne pouvez pas prévoir quelle valeur sera utilisée.
- Faites attention aux copies de sauvegarde que certains éditeurs de texte créent automatiquement (par exemple, la modification de `include/my_specific.conf` peut créer `include/my_specific.conf.BAK`), car les deux fichiers seront inclus.

Pour inclure uniquement les fichiers pertinents d'un répertoire, utilisez le caractère générique (*) pour la correspondance de motifs.

Exemple :

```
Include=/absolute/path/to/config/files/*.conf
```

ListenPort

Le service écoutera sur ce port pour les connexions provenant du serveur.

Par défaut: 10053
 Plage: 1024-32767

LogFile

Le nom du fichier journal.

Exemple :

`/tmp/zabbix_web_service.log`

Obligatoire : Oui, si LogType est défini sur `file` ; sinon non

LogFileSize

La taille maximale d'un fichier journal en MB.
0 - désactive la rotation automatique des journaux.
Remarque : si la limite de taille du fichier journal est atteinte et que la rotation du fichier échoue, quelle qu'en soit la raison, le fichier journal existant est tronqué et recréé.

Par défaut : 1
 Plage : 0-1024

LogType

Le type de sortie du journal : - *file* - écrit le journal dans le fichier spécifié par le paramètre LogFile ; - *system* - écrit le journal dans syslog ; - *console* - écrit le journal sur la sortie standard.

Par défaut : `file`

Délai d'attente

Temps maximal (en secondes) consacré au formatage du PDF rapport d'un tableau de bord.

Par défaut: 10
 Plage: 1-30

TLSAccept

Quelles connexions entrantes accepter : - *unencrypted* - accepter les connexions sans chiffrement (par défaut) ; - *cert* - accepter les connexions avec TLS et un certificat.

Par défaut : `unencrypted`

TLSCAFile

Le chemin complet du fichier contenant le ou les certificats de l'autorité de certification racine utilisés pour la vérification du certificat du pair, employé pour les communications chiffrées entre les composants Zabbix.

TLSCertFile

Le chemin complet du fichier contenant le certificat du service ou la chaîne de certificats, utilisé pour les communications chiffrées avec les composants Zabbix.

TLSKeyFile

Le chemin d'accès complet du fichier contenant la clé privée du service, utilisé pour les communications chiffrées entre les composants Zabbix.

Adaptateur bridge

L'adaptateur bridge Zabbix permet à votre serveur Zabbix de communiquer avec l'application mobile Zabbix.

L'application mobile Zabbix ne se connecte pas directement à votre serveur Zabbix. À la place, deux composants échangent des données entre eux :

- Zabbix Bridge - un service hébergé de manière centralisée qui échange des données avec l'application mobile Zabbix sur votre appareil mobile.
- Adaptateur bridge Zabbix - un composant que vous installez et exécutez. Il connecte votre serveur Zabbix à Zabbix Bridge.

L'adaptateur bridge peut être installé sur le même système que votre serveur Zabbix ou sur un autre système, à l'aide de packages.

Pour configurer l'adaptateur bridge Zabbix, mettez à jour son fichier de configuration (chemin par défaut : `/etc/zabbix/zabbix_bridge_ad`

Pour sécuriser l'adaptateur bridge Zabbix, consultez Accès mobile sécurisé.

Flux de données Lorsque vous utilisez l'application mobile Zabbix, les données circulent comme suit :

1. Vous effectuez une action dans l'application mobile Zabbix, qui envoie ensuite une requête à Zabbix Bridge.
2. Zabbix Bridge transmet la requête à l'adaptateur Zabbix bridge sous forme de requête JSON-RPC.
3. L'adaptateur Zabbix bridge vérifie si la méthode de l'API Zabbix demandée est autorisée. Si la méthode n'est pas autorisée, l'adaptateur bridge rejette la requête. Si la méthode est autorisée, l'adaptateur bridge envoie la requête à l'API Zabbix.
4. L'API Zabbix traite la requête et renvoie une réponse à l'adaptateur Zabbix bridge.
5. L'adaptateur Zabbix bridge renvoie la réponse à Zabbix Bridge.
6. Zabbix Bridge transmet la réponse à l'application mobile Zabbix, qui affiche les données sur votre appareil mobile.

Paramètres de configuration : adaptateur de pont Zabbix

Vue d'ensemble

Les paramètres de configuration permettent de personnaliser le comportement de l'adaptateur de pont Zabbix.

Indiquez les valeurs des paramètres dans le fichier de configuration (`zabbix_bridge_adapter.conf`). Tous les paramètres sont **facultatifs**, sauf indication explicite contraire.

Cette page est structurée comme suit :

- Vue d'ensemble des paramètres (pour afficher tous les détails d'un paramètre, cliquez sur son nom)
- Détails des paramètres

Paramètre	Description
AllowedAPIMethods	Indiquer les méthodes de l'API Zabbix pouvant être appelées via l'adaptateur de pont Zabbix.
DataDir	Indiquer le répertoire dans lequel le fichier d'état de l'adaptateur de pont et les clés de chiffrement sont créés et stockés.
DebugLevel	Définir le niveau de débogage.
Include	Indiquer un répertoire ou des fichiers individuels à inclure dans le fichier de configuration.
ListenIP	Indiquer l'adresse IP d'écoute de l'adaptateur de pont pour les connexions HTTP(S) entrantes depuis le serveur Zabbix.
ListenHealthPort	Indiquer le port d'écoute de l'adaptateur de pont pour les connexions HTTP entrantes de vérification de l'état.
ListenPort	Indiquer le port d'écoute de l'adaptateur de pont pour les connexions HTTP(S) entrantes depuis le serveur Zabbix.
LogFile	Indiquer le chemin complet du fichier journal de l'adaptateur de pont.
LogFileSize	Définir la taille maximale du fichier journal (en Mo).
LogType	Indiquer le type de sortie du journal.
PidFile	Indiquer le fichier PID.
TLSConfig.AllowedCN	Indiquer une liste séparée par des virgules de noms communs client autorisés (`mTLS` uniquement).
TLSConfig.CAFile	Indiquer le chemin complet d'un fichier contenant le ou les certificats CA de niveau supérieur utilisés pour vérifier les certificats client en mode `mTLS`.
TLSConfig.CertFile	Indiquer le chemin complet d'un fichier contenant le certificat ou la chaîne de certificats.
TLSConfig.KeyFile	Indiquer le chemin complet d'un fichier contenant la clé privée.
TLSConfig.Mode	Définir le mode TLS pour les connexions entrantes sur `ListenPort`.
ServerId	Indiquer l'identifiant unique du serveur Zabbix. **Obligatoire**.
ZabbixAPITLSCAFile	Indiquer le chemin complet d'un fichier contenant le ou les certificats CA de niveau supérieur utilisés par l'adaptateur de pont pour vérifier le certificat de l'API Zabbix.
ZabbixAPITLSConnect	Définir le mode TLS pour les connexions sortantes vers l'API Zabbix.
ZabbixAPIURL	Indiquer l'URL du point de terminaison de l'API Zabbix. **Obligatoire**.
ZabbixBridgeURL	Indiquer l'URL du pont Zabbix pour récupérer la configuration du pont Zabbix (récupérée au démarrage de l'adaptateur de pont).

Notez que :

- Les valeurs par défaut reflètent les valeurs par défaut du processus, et non celles des fichiers de configuration fournis.
- Les valeurs prennent en charge les variables d'environnement.
- Zabbix prend uniquement en charge les fichiers de configuration encodés en UTF-8 sans BOM ;
- Les commentaires commençant par # sont pris en charge uniquement au début de la ligne.

Détails des paramètres

AllowedAPIMethods

Indiquez quelles méthodes de l'API Zabbix peuvent être appelées via l'adaptateur Zabbix bridge.
 Pour plusieurs méthodes, répétez le paramètre, à raison d'une méthode par ligne.

Chaque valeur de paramètre doit être soit un nom de méthode exact (par exemple, `problem.get`), soit un modèle générique qui remplace uniquement le nom complet de l'API (par exemple, `*.get`) ou le nom complet de la méthode (par exemple, `problem.*`).

Toute autre valeur n'est pas valide et empêche le démarrage de l'adaptateur Zabbix bridge. Cela inclut un caractère générique seul (`*`), un modèle générique partiel (`host*` ou `host*.*`) ou une valeur vide (`AllowedAPIMethods=`).

Exemple :

```
AllowedAPIMethods=apiinfo.version
AllowedAPIMethods=problem.*
AllowedAPIMethods=*.get
```

Ensemble minimal de méthodes requises par l'application mobile Zabbix :

```
AllowedAPIMethods=apiinfo.version
AllowedAPIMethods=device.onboard
AllowedAPIMethods=device.offboard
AllowedAPIMethods=user.get
AllowedAPIMethods=role.get
AllowedAPIMethods=problem.get
AllowedAPIMethods=event.get
AllowedAPIMethods=event.acknowledge
AllowedAPIMethods=trigger.get
AllowedAPIMethods=alert.get
AllowedAPIMethods=host.get
AllowedAPIMethods=hostgroup.get
```

Par défaut : non défini (toutes les méthodes de l'API Zabbix sont refusées)

DataDir

Spécifiez le répertoire dans lequel le fichier d'état de l'adaptateur bridge et les clés de chiffrement sont créés et stockés.
 Le répertoire doit disposer des permissions 0700, et tous les fichiers qui y sont stockés doivent disposer des permissions 0600.

Par défaut : `/var/lib/zabbix-bridge-adapter`

DebugLevel

Définir le niveau de débogage :

- 0 - informations de base sur le démarrage et l'arrêt des processus Zabbix
- 1 - informations critiques
- 2 - informations sur les erreurs
- 3 - avertissements
- 4 - pour le débogage (génère de nombreuses informations)
- 5 - débogage étendu (génère encore plus d'informations)

Par défaut : 3
 Plage : 0-5

Inclure

Indiquez un répertoire ou des fichiers individuels à inclure dans le fichier de configuration.
 Lors de l'installation de Zabbix, ce paramètre est défini sur `/usr/local/etc`, sauf si vous modifiez cette valeur lors de la compilation.

Règles générales pour le paramètre `Include` :

- Le chemin peut être relatif à l'emplacement du fichier `zabbix_bridge_adapter.conf`.
- Vous pouvez spécifier plusieurs paramètres `Include` dans un même fichier de configuration.
- Zabbix traite le paramètre `Include` dès qu'il le lit. Il est donc recommandé de le placer à la fin du fichier de configuration.

Règles applicables aux fichiers inclus :

- Chaque fichier doit être lisible.
- Chaque fichier doit avoir une syntaxe correcte. Dans le cas contraire, le composant ne démarrera pas.

Règles applicables à l'inclusion de répertoires ou de modèles de fichiers :

- Ne partez pas du principe qu'un ordre d'inclusion est appliqué (par exemple, les fichiers ne sont pas inclus dans l'ordre alphabétique).
- Ne définissez pas le même paramètre dans plusieurs fichiers inclus (par exemple, pour remplacer un paramètre général par un paramètre spécifique), car vous ne pouvez pas prévoir quelle valeur sera utilisée.
- Faites attention aux copies de sauvegarde que certains éditeurs de texte créent automatiquement (par exemple, la modification de `include/my_specific.conf` peut créer `include/my_specific.conf.BAK`), car les deux fichiers seront inclus.

Pour inclure uniquement les fichiers pertinents d'un répertoire, utilisez le caractère générique (*) pour la correspondance de modèles.

Exemple :

```
Include=/absolute/path/to/config/files/*.conf
```

ListenIP

Spécifiez l'adresse IP d'écoute de l'adaptateur bridge pour les connexions HTTP(S) entrantes du serveur Zabbix.
 Si elle n'est pas spécifiée, l'adaptateur bridge écoute sur toutes les adresses IP.

ListenHealthPort

Spécifiez le port d'écoute de l'adaptateur bridge pour les connexions HTTP entrantes de vérification de l'état.
 Fournit les points de terminaison `/health` et `/ready`.
 S'il n'est pas spécifié, le service d'état n'est pas démarré.

Valeur par défaut : 8081
 Plage : 1024-32767

ListenPort

Spécifiez le port d'écoute de l'adaptateur bridge pour les connexions HTTP(S) entrantes du serveur Zabbix.

Valeur par défaut : 10054
 Plage : 1024-32767

LogFile

Indiquez le chemin complet vers le fichier journal de l'adaptateur bridge.

Obligatoire : **Oui** si LogType est défini sur `file`

Par défaut : `/tmp/zabbix_bridge_adapter.log`

LogFileSize

Définir la taille maximale du fichier journal (en Mo).
 0 - désactiver la rotation automatique des journaux.

Valeur par défaut : 1
 Plage : 0-1024

LogType

Spécifiez le type de sortie du journal :

- `file` - écrit le journal dans le fichier spécifié par le paramètre LogFile.
- `system` - écrit le journal dans syslog.
- `console` - écrit le journal sur la sortie standard.

Valeur par défaut : `file`

PidFile

Spécifiez le fichier PID.

Par défaut : `/tmp/zabbix_bridge_adapter.pid`

ServerId

Spécifiez l'identifiant unique du serveur Zabbix.
 Disponible dans *Rapports* > *Informations système* dans l'interface utilisateur de Zabbix.

Obligatoire : **Oui**

TLSConfig.AllowedCN

Spécifiez une liste de noms communs (Common Names) des clients autorisés, séparés par des virgules (mTLS uniquement).
 Si cette option n'est pas spécifiée, l'adaptateur bridge ne vérifie pas le nom commun du client.

TLSConfig.CAFile

Spécifiez le chemin complet vers un fichier contenant le ou les certificats CA racine utilisés pour vérifier les certificats client en mode mTLS.

TLSConfig.CertFile

Spécifiez le chemin complet vers un fichier contenant le certificat ou la chaîne de certificats.

TLSConfig.KeyFile

Spécifiez le chemin complet vers un fichier contenant la clé privée.

TLSConfig.Mode

Définissez le mode TLS pour les connexions entrantes sur ListenPort :

- none - accepte les connexions sans chiffrement.
- tls - TLS standard avec v1.2 comme version minimale.
- mtls - étend TLS avec une authentification mutuelle.

Valeur par défaut : none

ZabbixAPITLSCAFile

Spécifiez le chemin complet d'un fichier contenant le ou les certificats CA de niveau supérieur utilisés par l'adaptateur de pont pour vérifier le certificat de l'API Zabbix.

Obligatoire : **Oui** si ZabbixAPITLSConnect est défini sur verify_ca ou verify_full.

ZabbixAPITLSConnect

Définissez le mode TLS pour les connexions sortantes vers l'API Zabbix :

- disabled (ou vide) - utilise HTTP en clair ; ZabbixAPIURL doit utiliser http://.
- required - utilise HTTPS sans vérification du certificat (uniquement pour les tests).
- verify_ca - utilise HTTPS et vérifie le certificat de l'API Zabbix.
- verify_full - utilise HTTPS et vérifie le certificat ainsi que le nom d'hôte de l'API Zabbix.

ZabbixAPIURL

Spécifiez l'URL du point de terminaison de l'API Zabbix.

Obligatoire : **Oui**

Exemple :

```
ZabbixAPIURL=https://example.com/zabbix/api_jsonrpc.php
```

ZabbixBridgeURL

Spécifiez l'URL du bridge Zabbix pour récupérer la configuration du bridge Zabbix (récupérée au démarrage de l'adaptateur du bridge).

Par défaut : ZabbixBridgeURL=https://test.aws.zabbix-lab.win/.well-known/bridge-configuration

serveur MCP

Présentation générale Le serveur MCP Zabbix est un processus qui permet aux agents IA d'utiliser Zabbix avec le Model Context Protocol (MCP). Le serveur MCP communique avec les clients MCP et l'API Zabbix.

Avec le serveur MCP, un agent IA peut :

- Récupérer les groupes de hôtes, les groupes de modèles, les hôtes, les modèles, les éléments, les règles LLD, les interfaces, les macros, les déclencheurs, les problèmes, les périodes de maintenance et les données d'historique des éléments (à l'exception des éléments binaires) ;
- Effectuer des mises à jour des problèmes : ajouter des commentaires, acquitter ou annuler l'acquittement, modifier la gravité, supprimer ou rétablir la suppression, ou fermer les problèmes ;
- Créer, mettre à jour et supprimer des périodes de maintenance.

Les jetons d'API Zabbix sont utilisés pour l'autorisation ; les jetons sont envoyés avec chaque requête. L'accès est limité par les permissions du propriétaire du jeton : seuls les données des groupes de hôtes accessibles à cet utilisateur sont visibles, et les actions que l'utilisateur n'est pas autorisé à effectuer sont rejetées.

Le serveur MCP est sans état : aucune session n'est conservée entre les requêtes.

> **Note:**
> Bien que le serveur MCP Zabbix puisse fonctionner avec des versions antérieures ou ultérieures de Zabbix, sa compatibilité est officiellement prise en charge uniquement avec la même version de Zabbix que celle du serveur MCP lui-même.

Voir également :

- Paramètres de configuration : serveur MCP
- Outils et ressources

Installation Le paquet officiel zabbix-mcp-server est disponible dans le dépôt Zabbix.

Pour installer le serveur MCP Zabbix à partir des sources, compilez-le avec `make`.

Pour configurer le serveur MCP Zabbix, mettez à jour les paramètres du fichier de configuration du serveur MCP. Les paramètres `FrontendURL` et `AllowedIP` doivent au minimum être définis.

Le fichier de configuration par défaut (`zabbix_mcp_server.conf`) se trouve dans `/etc/zabbix/` ; il interdit les outils qui exposent le plus souvent des données sensibles. Pour activer un outil, supprimez la ligne `DenyTool` correspondante.

Voir également : Bonnes pratiques de sécurité

Exécution Pour démarrer le serveur MCP, exécutez :

```
zabbix_mcp_server -c /etc/zabbix/zabbix_mcp_server.conf
```

Pour vérifier le fichier de configuration et quitter sans démarrer le serveur, utilisez l'option `-T` :

```
zabbix_mcp_server -c /etc/zabbix/zabbix_mcp_server.conf -T
```

Les options de ligne de commande suivantes sont prises en charge :

Option	Description
`-c <file>`	Chemin vers le fichier de configuration. La valeur par défaut est `/etc/zabbix/zabbix_mcp_server.conf`.
`-T`	Valider le fichier de configuration et quitter.
`-h`	Afficher les informations d'aide.
`-V`	Afficher le numéro de version.

Lorsque le serveur MCP est installé à partir des sources, une unité systemd doit être créée manuellement, par exemple :

```
[Unit]
Description=Serveur MCP Zabbix
After=network-online.target

[Service]
User=zabbix
ExecStart=/usr/local/sbin/zabbix_mcp_server -c /etc/zabbix/zabbix_mcp_server.conf
Restart=on-failure

[Install]
WantedBy=multi-user.target
```

Connexion d'un client MCP Tout client MCP prenant en charge le transport HTTP Streamable peut être connecté au serveur MCP. Les paramètres suivants sont requis :

Paramètre	Valeur
Transport	HTTP Streamable
URL	`https://<host>:<ListenPort>/mcp`
En-tête	`Authorization: Bearer <Zabbix API token>`, envoyé avec chaque requête

Chaque utilisateur doit configurer son propre token API.

Pour vérifier la configuration, envoyez une seule requête au serveur MCP :

```
curl -sS https://mcp.example.com:8443/mcp \
  -H "Authorization: Bearer $ZABBIX_API_TOKEN" \
  -H "Content-Type: application/json" \
  -H "Accept: application/json, text/event-stream" \
  -d '{"jsonrpc":"2.0","id":1,"method":"tools/list"}'
```

Le code d'état HTTP 200 et une liste des outils disponibles dans la réponse confirment que le serveur MCP est accessible et accepte le token ; la liste indique les outils exposés. Les autres codes d'état indiquent les problèmes suivants :

- 400 - l'en-tête `Accept` n'inclut pas les deux types de média requis par le transport ;
- 401 - l'en-tête `Authorization` contenant le token est manquant ;
- 403 - l'adresse du client ne figure pas dans le paramètre `AllowedIP`.

Approbation et accès aux outils Si le client MCP prend en charge la sollicitation via formulaire, chaque modification doit être approuvée par l'utilisateur avant d'être envoyée au serveur Zabbix. Le client MCP affiche à l'utilisateur les modifications qui seront effectuées, et la modification est appliquée uniquement si l'utilisateur la confirme ; si l'utilisateur rejette ou ignore la demande, aucune modification n'est effectuée dans Zabbix. Une approbation est valide pendant cinq minutes et uniquement pour la modification exacte qui a été affichée ; pendant cette période, l'approbation peut être réutilisée.

Si le client MCP ne prend pas en charge la sollicitation via formulaire (y compris les clients qui prennent uniquement en charge la sollicitation via URL), les modifications sont appliquées immédiatement, sans approbation. Dans ce cas, limitez les outils d'écriture et les autorisations de l'API Zabbix.

L'approbation est particulièrement importante, car la sortie des outils peut contenir du texte non fiable provenant de systèmes surveillés, susceptible d'influencer un agent.

Utilisez AllowTool, DenyTool, AllowToolRegexp et DenyToolRegexp pour contrôler les outils exposés par le serveur MCP. Les outils refusés ne peuvent être ni appelés ni répertoriés, et les ressources qui leur sont associées ne sont pas accessibles.

Warning:
Les règles AllowTool, DenyTool, AllowToolRegexp et DenyToolRegexp limitent l'accès MCP, mais ne modifient pas les autorisations du jeton Zabbix. Si l'adresse de l'interface est exposée à un agent IA, celui-ci peut tenter d'envoyer des appels directs à l'API. Gardez l'emplacement de l'interface Zabbix et la configuration du serveur MCP inaccessibles à l'agent afin que les requêtes doivent passer par le serveur MCP et ses contrôles de sécurité.

Bonnes pratiques de sécurité Jetons API

L'authentification et l'autorisation sont entièrement gérées par Zabbix. Le client MCP envoie un jeton API Zabbix avec chaque requête ; le serveur MCP le transmet sans le stocker ni l'enregistrer dans les journaux, tandis que Zabbix valide le jeton, applique les autorisations et consigne les actions dans le journal d'audit.

Utilisez un jeton distinct, avec le moins de privilèges possible et une date d'expiration, pour chaque agent et chaque utilisateur. Stockez les jetons dans un emplacement sécurisé et renouvelez régulièrement les jetons API.

Chiffrement TLS

Pour empêcher l'envoi des jetons en clair, définissez `TLSAccept=cert` et spécifiez le certificat et la clé (`TLSCertFile` et `TLSKeyFile`), ou exécutez le serveur MCP derrière un proxy inverse de terminaison TLS de confiance et définissez `TLSAccept=unencrypted`. Le protocole HTTP non chiffré n'est acceptable que sur un réseau de confiance.

Restrictions d'accès

Limitez le paramètre `AllowedIP` aux adresses des clients MCP de confiance et utilisez les permissions Zabbix pour limiter les actions que les jetons d'API peuvent effectuer.

Le serveur MCP ne fournit pas de limitation du débit des requêtes. Chaque requête MCP entraîne normalement un appel à l'API Zabbix, bien qu'un agent exécuté en boucle puisse tout de même générer une charge importante.

Protection des données sensibles

Les clés d'éléments peuvent contenir des identifiants (par exemple, dans les éléments SSH, JMX, ODBC et agent HTTP), les valeurs des macros peuvent contenir des mots de passe et des chaînes de communauté, les interfaces révèlent des adresses internes, et les valeurs des éléments de journal ou de texte peuvent contenir n'importe quelles données. Pour cette raison, les outils `item_get`, `lld_get`, `interface_get`, `macro_get` et `history_get` sont refusés par défaut dans le fichier de configuration fourni.

Pour éviter d'exposer les valeurs sensibles des macros au fournisseur LLM de l'agent connecté, utilisez les macros secrètes ; leurs valeurs ne sont jamais renvoyées par l'API.

Outils et ressources

Vue d'ensemble

Le serveur MCP Zabbix expose les fonctionnalités de Zabbix aux clients MCP sous forme d'*outils* et de *ressources*.

Les noms des outils décrivent l'activité. Si le paramètre ToolPrefix est défini, le préfixe est ajouté à chaque nom d'outil ; les règles d'accès aux outils correspondent aux noms simples répertoriés ci-dessous.

Outils de lecture

Outil	Méthode de l'API Zabbix	Arguments	Trié par	Ressource
`hostgroup_get`	hostgroup.get	`groupids`, `names`, `search`, `limit`	name	`zabbix://hostgroup/<i`
`templategroup_get`	templategroup.get	`groupids`, `names`, `search`, `limit`	name	`zabbix://templategrou`
`host_get`	host.get	`hostids`, `groupids`, `search`, `limit`	name	`zabbix://host/<id>`
`template_get`	template.get	`templateids`, `groupids`, `search`, `limit`	name	`zabbix://template/<id`
`item_get`	item.get	`hostids`, `itemids`, `search`, `limit`	name	`zabbix://item/<id>`
`lld_get`	discoveryrule.get	`hostids`, `ruleids`, `limit`	name	-
`interface_get`	hostinterface.get	`hostids`, `interfaceids`, `limit`	ID de l'interface	-
`macro_get`	usermacro.get	`hostids` ou `global`, `macro_names`, `limit`	macro	-
`problem_get`	problem.get	`eventids`, `groupids`, `hostids`, `severities`, `acknowledged`, `suppressed`, `recent`, `limit`	ID d'événement, décroissant	-
`maintenance_get`	maintenance.get	`maintenanceids`, `groupids`, `hostids`, `limit`	name	-
`history_get`	history.get	`itemids`, `time_from`, `time_till`, `type`, `limit`	clock, décroissant	-
`trigger_get`	trigger.get	`triggerids`, `hostids`, `search`, `limit`	description	`zabbix://trigger/<id>`

Notez que :

- `host_get` et `template_get` renvoient également les déclencheurs de chaque objet (un ensemble limité de champs), les étiquettes et les modèles liés. Les éléments, les règles de découverte, les interfaces et les macros sont renvoyés par les outils correspondants.
- `history_get` nécessite une plage temporelle ne dépassant pas 7 jours et ne commençant pas plus tôt qu'il y a 7 jours. L'argument `type` doit être l'un des suivants : `float`, `unsigned`, `character`, `text`, `log` ou `json`, et doit correspondre au type de valeur des éléments ; si le type est incorrect, un résultat vide est renvoyé, ce qui ne peut pas être distingué d'un élément sans données. L'historique des éléments binaires (type de valeur 5) ne peut pas être récupéré via le serveur MCP.
- Les outils de lecture demandent une liste fixe de champs ; tous les champs (`output: extend`) ne sont jamais demandés.
- Les résultats des outils qui disposent d'une ressource contiennent le champ `uri`, afin qu'un agent puisse passer d'une liste à une entité unique.

Format de la réponse

Tous les outils de lecture renvoient la réponse au même format :

```
{
  "results": [ ... ],
  "count": 2,
  "truncated": true,
  "truncated_by": "count",
  "limit": 2,
  "note": "The result count limit was reached ..."
}
```

Une réponse est limitée par l'argument `limit` (valeur par défaut : 100, maximum : 1000) et par la taille sérialisée de la réponse (MaxResponseSize, valeur par défaut : 64K). La réponse s'arrête dès que l'une de ces limites est atteinte. La réponse est tronquée uniquement entre les résultats, jamais au sein d'un résultat, et une réponse tronquée indique quelle limite a été atteinte. Chaque outil utilise un ordre de tri fixe ; une réponse tronquée correspond donc toujours au début de l'ensemble complet des résultats. La pagination n'est pas prise en charge ; pour obtenir d'autres résultats, affinez la requête.

Outils de problème

Tous les outils de problème appellent event.acknowledge avec les indicateurs d'action correspondants.
L'argument `eventids` est obligatoire ; les ID d'événements peuvent être récupérés avec `problem_get`.

Un message (commentaire) peut être ajouté avec n'importe quel outil de problème.
L'argument `message` est facultatif, sauf pour `problem_annotate`, où il est obligatoire ; s'il est spécifié, le message est enregistré dans le même appel de l'API Zabbix.

Tool	Effect	Additional arguments
`problem_acknowledge`	Accuser réception du problème.	-
`problem_unacknowledge`	Supprimer l'accusé de réception.	-
`problem_close`	Fermer manuellement le problème (uniquement pour les déclencheurs qui autorisent la fermeture manuelle).	-
`problem_annotate`	Enregistrer un message.	`message` (obligatoire)
`problem_change_severity`	Modifier la gravité du problème.	`severity` (0-5)
`problem_suppress`	Supprimer le problème jusqu'à une heure spécifiée ou indéfiniment.	`suppress_until`
`problem_unsuppress`	Supprimer la suppression.	-

Les problèmes ne peuvent pas être créés ou supprimés ; tous les outils de problème mettent à jour les problèmes existants.

Outils de maintenance

Outil	Méthode de l'API Zabbix	Arguments
`maintenance_create`	maintenance.create	`name`, `groupids` et/ou `hostids`, `active_since`, `active_till`, `timeperiods`, `description`, `tags`, `tags_evaltype`
`maintenance_update`	maintenance.update	`maintenanceid` et n'importe lequel des arguments ci-dessus. Les arguments de type liste remplacent l'ensemble des valeurs ; une liste vide efface l'ensemble.
`maintenance_delete`	maintenance.delete	`maintenanceids`

Lorsqu'une période de maintenance est mise à jour, les arguments de type liste (tels que `hostids`, `groupids` et `timeperiods`) remplacent les valeurs existantes au lieu de leur ajouter de nouvelles valeurs. Par exemple, si une période de maintenance couvre les hôtes A et B et que `maintenance_update` est appelé avec `hostids` contenant uniquement l'hôte C, la période de maintenance couvrira uniquement l'hôte C.

`maintenance_create` est la seule opération qui n'est pas idempotente, et `maintenance_delete` est la seule opération qui ne peut pas être annulée. Pour empêcher leur utilisation, refusez ces outils à l'aide du paramètre DenyTool.

Comportement des outils d'écriture

Les outils de problème et de maintenance sont des *outils d'écriture*. Ils fonctionnent comme suit :

- Les arguments sont validés avant tout appel à l'API Zabbix ; si la validation échoue, l'erreur indique la contrainte qui n'a pas été respectée.
- Si le client le prend en charge, la modification est appliquée uniquement après avoir été approuvée par l'utilisateur.
- La modification est effectuée par Zabbix à l'aide du token de l'appelant. Zabbix vérifie les autorisations et consigne la modification dans le journal d'audit en l'associant au propriétaire du token.
- Les erreurs Zabbix, y compris les erreurs d'autorisation, sont renvoyées sans modification sous forme d'objet d'erreur avec les champs `code`, `message` et `data`.
- Les outils d'écriture renvoient directement le résultat Zabbix ; les limites de réponse des outils de lecture ne s'appliquent pas.
- Les appels d'écriture ne sont jamais réessayés par le serveur MCP.

Ressources

Les ressources utilisent le schéma d'URI `zabbix://<kind>/<id>`. Seule la ressource `zabbix://about` est incluse dans la réponse `resources/list` ; les ID d'entités ne peuvent pas être énumérés et sont obtenus à partir des résultats des outils. La réponse `resources/templates/list` répertorie les types d'entités.

URI	Contenu
`zabbix://about`	L'URL de l'API, l'URL de l'interface, la version de Zabbix, l'identité à laquelle correspond le token de l'appelant et la liste des outils exposés.
`zabbix://host/<hostid>`	L'enregistrement du hôte sans les éléments, les déclencheurs, les interfaces et les macros.
`zabbix://hostgroup/<groupid>`	L'enregistrement du groupe de hôtes sans la liste des membres.

URI	Contenu
`zabbix://template/<templateid>`	L'enregistrement du modèle sans les éléments, les déclencheurs et les règles de découverte.
`zabbix://templategroup/<groupid>`	L'enregistrement du groupe de modèles sans la liste des membres.
`zabbix://item/<itemid>`	La définition de l'élément sans sa valeur actuelle.
`zabbix://trigger/<triggerid>`	La définition du déclencheur sans les hôtes, les dépendances, les fonctions et les tags.

Une ressource dont l'outil correspondant est refusé n'existe pas. Une entité inexistante et une entité à laquelle le propriétaire du token n'a pas accès sont toutes deux signalées comme *resource not found* ; ces cas sont volontairement impossibles à distinguer. La ressource `zabbix://about` n'est pas affectée par les règles d'accès aux outils.

Paramètres de configuration : serveur MCP Zabbix

Présentation générale

Le serveur MCP Zabbix est un processus qui permet aux agents d'IA d'utiliser Zabbix avec le protocole Model Context Protocol (MCP).

Les paramètres pris en charge par le fichier de configuration du serveur MCP Zabbix (`zabbix_mcp_server.conf`) sont répertoriés dans cette section.

Les paramètres sont répertoriés sans informations supplémentaires. Cliquez sur un paramètre pour afficher tous les détails.

Parameter	Description
AllowedIP	Une liste d'adresses IP séparées par des virgules, éventuellement au format CIDR, ou de noms DNS des clients MCP.
AllowTool	Une règle générique qui autorise un outil.
AllowToolRegexp	Une règle d'expression régulière qui autorise des outils.
DebugLevel	Le niveau de débogage.
DenyTool	Une règle générique qui refuse un outil.
DenyToolRegexp	Une règle d'expression régulière qui refuse des outils.
FrontendURL	Le point de terminaison de l'API Zabbix.
IgnoreURLCertErrors	Indique comment gérer les erreurs de validation des certificats TLS lors de l'accès à l'API Zabbix.
ListenIP	L'adresse IP sur laquelle écouter.
ListenPort	Le port sur lequel écouter.
LogFile	Le nom du fichier journal.
LogFileSize	La taille maximale du fichier journal.
LogType	Le type de sortie du journal.
MaxResponseSize	La taille maximale d'une réponse renvoyée par un outil de lecture.
PidFile	Le nom du fichier PID.
Timeout	Le délai maximal (en secondes) d'attente d'une réponse de l'API Zabbix.
TLSAccept	Les connexions entrantes à accepter.
TLSCertFile	Le chemin complet d'un fichier contenant le certificat du serveur ou la chaîne de certificats.
TLSKeyFile	Le chemin complet d'un fichier contenant la clé privée du serveur.
ToolPrefix	Un préfixe ajouté aux noms de tous les outils.

Tous les paramètres sont facultatifs, sauf indication explicite du contraire.

Notez que :

- Les valeurs par défaut correspondent aux valeurs par défaut du processus, et non aux valeurs des fichiers de configuration fournis ;
- Les valeurs prennent en charge les variables d'environnement au format `${NAME}` ;
- Zabbix prend uniquement en charge les fichiers de configuration encodés en UTF-8 sans BOM ;
- Les commentaires commençant par # sont pris en charge uniquement au début de la ligne.

Détails des paramètres

AllowedIP

Liste d'adresses IP séparées par des virgules, éventuellement au format CIDR, ou de noms DNS des clients MCP. Les connexions entrantes seront acceptées uniquement depuis les hôtes répertoriés ici.

Exemple :

```
AllowedIP=127.0.0.1,192.168.1.0/24,::1,2001:db8::/32,mcp-client.example.com
```

Obligatoire : oui

AllowTool

Une règle générique qui autorise un outil. Le paramètre peut être spécifié plusieurs fois.
 Les modèles génériques utilisent la syntaxe des clés d'élément, où * correspond à n'importe quels caractères. La règle correspond au nom simple de l'outil et ignore ToolPrefix.
 Toutes les règles `AllowTool`, `DenyTool`, `AllowToolRegexp` et `DenyToolRegexp` forment une seule liste, dans l'ordre dans lequel elles sont spécifiées ; la première règle correspondante est déterminante. Un outil qui ne correspond à aucune règle est autorisé.

Exemple :

```
AllowTool=host_get
```

AllowToolRegexp

Une règle d'expression régulière qui autorise les outils. Le paramètre peut être spécifié plusieurs fois.
 Les expressions régulières utilisent la syntaxe RE2 et correspondent à toute partie du nom d'outil simple, sauf si elles sont ancrées.

Exemple :

```
AllowToolRegexp=^(host|hostgroup|trigger|problem)_get$
```

DebugLevel

Spécifiez le niveau de débogage : - *0* - informations de base sur le démarrage et l'arrêt des processus Zabbix ; - *1* - informations critiques ; - *2* - informations sur les erreurs ; - *3* - avertissements ; - *4* - pour le débogage (génère beaucoup d'informations) ; - *5* - débogage étendu (génère encore plus d'informations).

Valeur par défaut : 3
 Plage : 0-5

DenyTool

Une règle avec caractère générique qui interdit un outil. Le paramètre peut être spécifié plusieurs fois.
 Un outil interdit n'est pas inclus dans la liste des outils et ne peut pas être appelé. Voir AllowTool pour plus de détails.

Exemple :

```
DenyTool=maintenance_delete
```

DenyToolRegexp

Une règle d'expression régulière qui interdit des outils. Le paramètre peut être spécifié plusieurs fois.
 Voir AllowToolRegexp pour plus de détails.

Exemple :

```
DenyToolRegexp=^problem_(close|suppress)$
```

FrontendURL

Le point de terminaison de l'API Zabbix : l'URL de l'interface Zabbix se terminant par `api_jsonrpc.php`.

Exemple :

```
FrontendURL=https://zabbix.example.com/api_jsonrpc.php
```

Obligatoire : oui

IgnoreURLCertErrors

Spécifie la gestion des erreurs de validation des certificats TLS lors de l'accès à l'API Zabbix :
 false - ne pas ignorer les erreurs de certificat ;
 true - ignorer les erreurs de certificat (à utiliser uniquement dans un environnement de développement).

Valeur par défaut : `false`

ListenIP

L'adresse IP sur laquelle écouter.

Par défaut : `0.0.0.0`

ListenPort

Le port d'écoute. Le chemin du point de terminaison MCP est `/mcp`.

Valeur par défaut : 8443
 Plage : 1024-32767

LogFile

Nom du fichier journal.

Par défaut : `/tmp/zabbix_mcp_server.log`

Obligatoire : Non

LogFileSize

La taille maximale d'un fichier journal en Mo.
0 - désactive la rotation automatique des journaux.

Valeur par défaut : 1
 Plage : 0-1024

LogType

Le type de sortie du journal : - *file* - écrit le journal dans le fichier spécifié par le paramètre LogFile ; - *system* - écrit le journal dans syslog ; - *console* - écrit le journal dans la sortie standard.

Par défaut : `file`

MaxResponseSize

La taille maximale d'une réponse renvoyée par un outil de lecture. La première limite atteinte entre cette limite et l'argument `limit` de l'outil arrête la réponse.
 Prend en charge les suffixes d'unités.

Valeur par défaut : 64K
 Plage : 16K-256K

PidFile

Nom du fichier PID.

Par défaut : `/tmp/zabbix_mcp_server.pid`

Délai d'attente

Le délai maximal (en secondes) à attendre pour obtenir une réponse de l'API Zabbix.

Valeur par défaut : 10
 Plage : 3-30

TLSAccept

Connexions entrantes à accepter : - *non chiffrées* - accepter les connexions sans chiffrement (par défaut) ; - *certificat* - accepter les connexions avec TLS et un certificat.

Par défaut : `unencrypted`

TLSCertFile

Le chemin complet du fichier contenant le certificat du serveur ou la chaîne de certificats.

Obligatoire : Oui, si `TLSAccept` est défini sur `cert` ; sinon, non

TLSKeyFile

Le chemin complet du fichier contenant la clé privée du serveur.

Obligatoire : Oui, si `TLSAccept` est défini sur `cert` ; sinon, non

ToolPrefix

Préfixe ajouté aux noms de tous les outils, par exemple pour éviter les conflits de noms avec les outils d'autres serveurs MCP. Par exemple, avec `ToolPrefix=zbx_`, l'outil `host_get` est exposé sous le nom `zbx_host_get`.
 Caractères autorisés : lettres, chiffres, trait de soulignement (`_`), trait d'union (`-`) et point (`.`).
 Les règles d'accès aux outils correspondent aux noms des outils sans le préfixe.

Exemple :

`ToolPrefix=zbx_`

Variables d'environnement

Vue d'ensemble Les variables d'environnement permettent de configurer les composants Zabbix sans coder en dur les valeurs dans les fichiers de configuration. Cela facilite la gestion des configurations dans des environnements dynamiques, tels que Docker, où des variables peuvent être transmises au moment de l'exécution afin d'adapter le système à différentes configurations.

Dans le cas le plus simple, en définissant la valeur du paramètre de configuration DebugLevel du serveur Zabbix sur une variable d'environnement, vous pouvez ensuite l'utiliser pour configurer le serveur au démarrage :

```
### Fichier de configuration du serveur Zabbix :
DebugLevel=${NEW_DEBUG_LEVEL}

### Démarrage du serveur Zabbix :
NEW_DEBUG_LEVEL=5 /usr/sbin/zabbix_server
```

Les variables d'environnement sont prises en charge par les composants Zabbix suivants :

- Serveur
- Proxy
- Agent (UNIX ou Windows)
- Agent 2 (UNIX ou Windows), y compris les plugins
- Service web
- Zabbix sender (lors de l'utilisation de l'option -c, --config)

Notes importantes

- Lorsqu'un paramètre de configuration est défini sur une variable d'environnement, qui n'est pas spécifiée lors de l'exécution du composant, la valeur par défaut du paramètre est utilisée.
- Lors de l'utilisation de commandes d'exécution (par exemple, pour augmenter le niveau de journalisation de l'agent), toutes les variables d'environnement précédemment utilisées doivent être spécifiées. En effet, les composants Zabbix utilisent leur fichier de configuration pour exécuter les commandes d'exécution ; si des variables d'environnement sont omises, les valeurs par défaut des paramètres de configuration seront utilisées. Voir Exemples.
- La commande d'exécution `userparameter_reload` commande d'exécution ne prend pas en charge le rechargement des variables d'environnement. Lors du rechargement, les variables sont ignorées, et seuls les paramètres avec des valeurs normales sont rechargés.
- Les variables d'environnement actuelles du processus, qui ont été utilisées dans les fichiers de configuration, sont effacées après le démarrage du composant Zabbix. Cela garantit que les processus enfants (par exemple, les scripts distants exécutés par Zabbix) ne peuvent pas accéder à ces variables. Toutefois, notez que les variables initiales du processus peuvent toujours être récupérées (par exemple, via le fichier `/proc/<PID>/environ`).

Syntaxe Les variables d'environnement doivent utiliser la syntaxe suivante : `${alphanumerics/underscores}`.

Le nom de la variable peut uniquement inclure des lettres (a-z, A-Z), des underscores (_), et des chiffres (0-9), et ne doit pas commencer par un chiffre.

Les variables qui ne correspondent pas à la syntaxe requise ou qui sont combinées avec une valeur normale seront traitées comme des valeurs normales, ce qui peut provoquer des erreurs.

Syntaxe correcte des variables :

```
DebugLevel=${NEW_DEBUG_LEVEL}
Hostname=${ZBX_HOSTNAME}
LogFile=${LogFile_001}
```

Syntaxe incorrecte des variables :

```
DebugLevel=${5_DebugLevel}
Hostname=${ZBX.HOSTNAME 1}
LogFile=/${HOME}/zabbix/zabbix_server.log
```

> **Note:**
> Sous Windows, les noms des variables d'environnement ne sont pas sensibles à la casse.

Exemples Les exemples suivants montrent comment configurer et utiliser des variables d'environnement avec les composants Zabbix.

Exemple 1 : Configuration et test de l'agent Zabbix

1. Définissez les variables d'environnement dans le fichier de configuration de l'agent :

```
Hostname=${ZBX_HOSTNAME}
ServerActive=${ServerActive}
```

2. Testez le fichier de configuration :

```
ZBX_HOSTNAME="New Zabbix agent" ServerActive=127.0.0.1 /usr/sbin/zabbix_agentd -c /etc/zabbix/zabbix_agent
```

3. Démarrez l'agent avec les variables d'environnement :

```
ZBX_HOSTNAME="New Zabbix agent" ServerActive=127.0.0.1 /usr/sbin/zabbix_agentd -c /etc/zabbix/zabbix_agent
```

Lorsque vous utilisez des commandes d'exécution (par exemple, pour augmenter le niveau de journalisation de l'agent), toutes les variables d'environnement utilisées précédemment doivent être spécifiées :

```
ZBX_HOSTNAME="New Zabbix agent" ServerActive=127.0.0.1 /usr/sbin/zabbix_agentd -c /etc/zabbix/zabbix_agent
```

Cela s'explique par le fait que l'agent utilise son fichier de configuration pour exécuter les commandes d'exécution ; si les variables d'environnement sont omises, les valeurs par défaut des paramètres de configuration seront utilisées.

Vous pouvez également, après avoir défini les variables d'environnement dans le fichier de configuration de l'agent, les rendre disponibles pour les processus (par exemple, en utilisant la commande `export`). Cela réduit le risque de comportement inattendu dû à des variables manquantes ou définies incorrectement.

```
export ZBX_HOSTNAME="New Zabbix agent"
export ServerActive=127.0.0.1
/usr/sbin/zabbix_agentd -c /etc/zabbix/zabbix_agentd.conf --test-config
/usr/sbin/zabbix_agentd -c /etc/zabbix/zabbix_agentd.conf
/usr/sbin/zabbix_agentd -c /etc/zabbix/zabbix_agentd.conf -R log_level_increase
```

Exemple 2 : Configuration de l'agent Zabbix pour un conteneur

Si vous créez et configurez votre propre image personnalisée pour les composants Zabbix (par exemple, l'agent Zabbix), vous pouvez définir les paramètres de configuration à l'aide de variables d'environnement, puis démarrer le conteneur avec ces variables.

1. Lors de la préparation de l'image, définissez les variables d'environnement dans le fichier de configuration de l'agent :

```
Hostname=${ZBX_HOSTNAME}
BufferSize=${BUFSZ}
ListenPort=${LISTENPORT}
UserParameter=${_UsrPar01}
UserParameter=${_UsrPar02}
```

2. Après avoir créé l'image du conteneur, démarrez le conteneur de l'agent (par exemple, Docker) avec les variables d'environnement :

```
docker run --name my-zabbix-agent -e ZBX_HOSTNAME="new-hostname" -e BUFSZ=1000 -e LISTENPORT=20050 -e _Usr
```

3. Lorsque vous utilisez des commandes d'exécution (par exemple, pour augmenter le niveau de journalisation de l'agent), accédez au shell du conteneur et exécutez la commande d'exécution :

```
docker exec -it <containerid> sh
/usr/sbin/zabbix_agentd -R log_level_increase
```

> **Attention:**
> La commande d'exécution `userparameter_reload` ne prend pas en charge le rechargement des variables d'environnement. Lors du rechargement, les variables sont ignorées et seuls les paramètres ayant des valeurs normales sont rechargés.

Configuration

Vue d'ensemble La configuration dans Zabbix consiste à spécifier quels hôtes et systèmes surveiller, à définir quelles données collecter et à établir comment les notifications sont envoyées lorsque des problèmes surviennent.

Utilisez la barre latérale pour naviguer vers les aspects de configuration pertinents.

Tâches de configuration clés

- Configurer un nouvel hôte ou un groupe d'hôtes → Hosts and host groups
- Définir quelles données collecter à partir des hôtes → Items
- Créer des règles pour détecter les problèmes → Problem detection with triggers
- Configurer la manière dont les notifications sont envoyées → Notifications upon events
- Afficher les données de supervision via des tableaux de bord, des graphiques et des cartes → Visualization
- Réutiliser des configurations sur plusieurs hôtes → Templates and template groups, Templates out of the box
- Contrôler l'accès des utilisateurs et les autorisations → Users and user groups

Bonnes pratiques de configuration Les choix de configuration ont un impact sur les performances et la maintenabilité. Bien que chaque installation Zabbix soit différente, ces bonnes pratiques peuvent servir de base et être adaptées selon les besoins.

Hôte distinct pour chaque instance de surveillance

Un hôte dans Zabbix est une entité logique plutôt qu'une machine ou un périphérique physique. Créez un hôte distinct pour chaque instance indépendante, telle qu'une machine virtuelle, une base de données, un conteneur ou un commutateur réseau.

Cette approche offre les avantages suivants :

1. Des données de surveillance plus claires, avec des éléments, des déclencheurs et des notifications d'alerte séparés pour chaque hôte.
2. Un contrôle d'accès utilisateur fin via user-roles, aligné sur le principe du moindre privilège.

Utiliser des modèles et la mise à jour en masse pour des hôtes similaires

Pour des hôtes similaires comme *Network switch 1* et *Network switch 2*, vous pouvez créer un modèle avec les métriques requises (*Network switch template*) et regrouper les hôtes dans un groupe d'hôtes (*Network switches*).

Lors de l'ajout de nouveaux commutateurs, vous pouvez utiliser la mise à jour en masse dans *Data Collection* > *Hosts* pour lier le modèle au lieu de cloner les hôtes manuellement.

Configurer des éléments maîtres et dépendants pour minimiser les requêtes

Pour minimiser les requêtes vers l'entité cible, configurez un élément maître pour collecter les données dans une seule requête, puis utilisez des éléments dépendants avec prétraitement pour extraire des valeurs spécifiques.

Par exemple, l'élément maître peut collecter une réponse JSON ou XML contenant plusieurs métriques, ou exécuter une requête de base de données renvoyant plusieurs colonnes (par exemple, le nombre de connexions ouvertes, les connexions interrompues, le nombre maximal de connexions simultanées autorisées et le total cumulé des connexions depuis le démarrage), et les éléments dépendants analyseront et stockeront chaque valeur séparément.

Supprimez l'historique de l'élément maître après la collecte et ne conservez que les données des éléments dépendants.

Déployer un proxy Zabbix lorsque cela est nécessaire

Si tous vos hôtes se trouvent sur le même réseau local que le serveur Zabbix et qu'aucun problème d'évolutivité ou de performances ne se pose, vous n'avez peut-être pas besoin d'un proxy. Dans les environnements plus vastes ou plus complexes, un proxy permet de mieux répartir la charge.

Ajoutez un proxy Zabbix lorsque :

1. Vous supervisez plusieurs hôtes à l'aide de différentes méthodes de collecte des métriques derrière un pare-feu. Le proxy collecte les données des hôtes et les transmet au serveur Zabbix, ce qui réduit le nombre de ports ouverts dans le pare-feu.
2. Vous supervisez des sites distants, des succursales ou des réseaux. Les proxys Zabbix situés sur des sites distants continuent à collecter les données pendant les interruptions réseau et les transmettent lorsque la connexion est rétablie.
3. Vous gérez des déploiements à grande échelle. Si vous avez de nombreux hôtes et/ou collectez un grand nombre de valeurs par seconde, déployez un proxy afin de réduire la charge du serveur et d'améliorer les performances.

Configurer la récupération indépendante des macros secrètes

Vous pouvez utiliser les macros utilisateur secret soit comme texte secret, soit comme macros de coffre-fort secret.

Par défaut, les valeurs des macros secrètes sont récupérées par le serveur Zabbix et propagées aux proxies. Pour renforcer la sécurité avec les macros de coffre-fort secret, configurez les valeurs des macros afin qu'elles soient récupérées indépendamment par le serveur Zabbix et les proxies.

Configuration d'un modèle

Aperçu

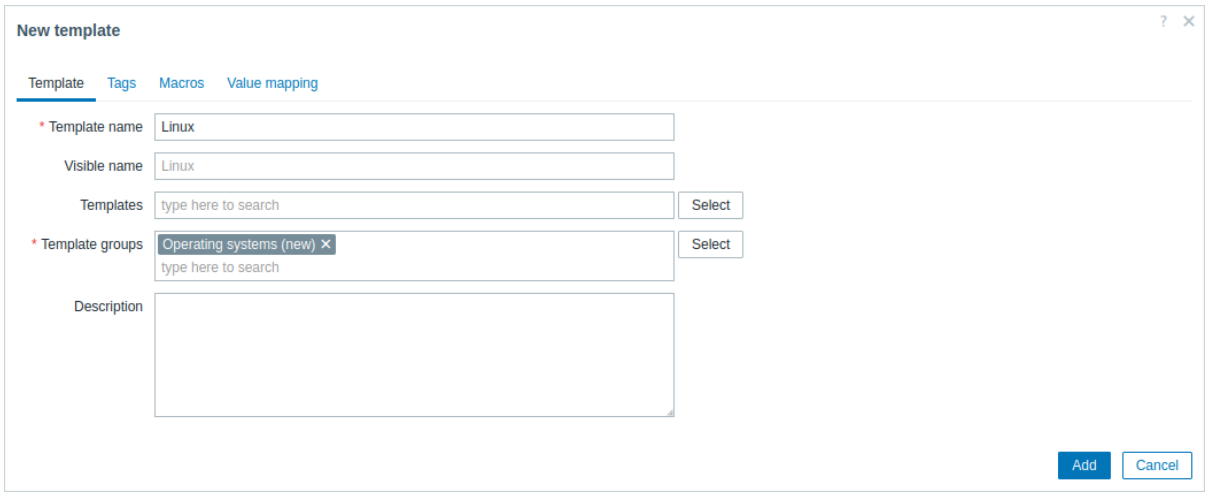
La configuration d'un modèle nécessite d'abord de créer un modèle en définissant ses paramètres généraux, puis d'y ajouter des entités (éléments, déclencheurs, graphiques, etc.).

Création d'un modèle

Pour créer un modèle, procédez comme suit :

1. Accédez à *Collecte de données* > *Modèles*.
2. Cliquez sur *Créer un modèle*.
3. Modifiez les attributs du modèle.

L'onglet **Modèle** contient les attributs généraux du modèle.



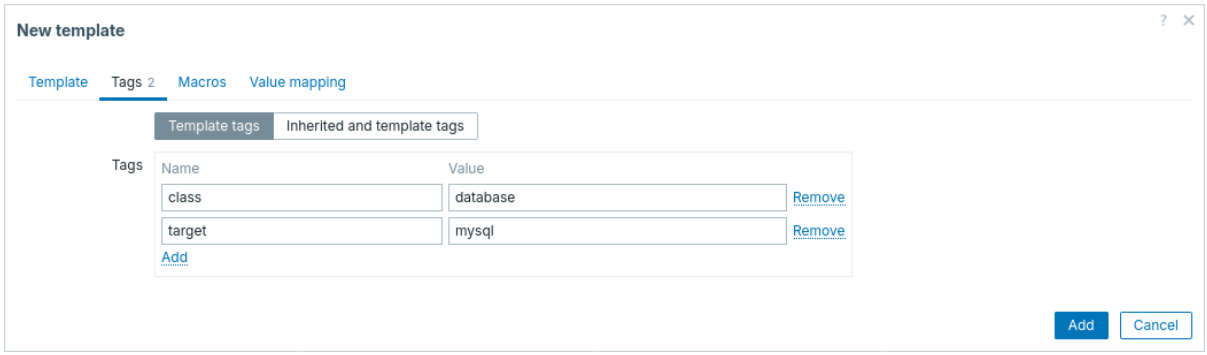
Tous les champs obligatoires sont marqués d'un astérisque rouge.

Attributs du modèle :

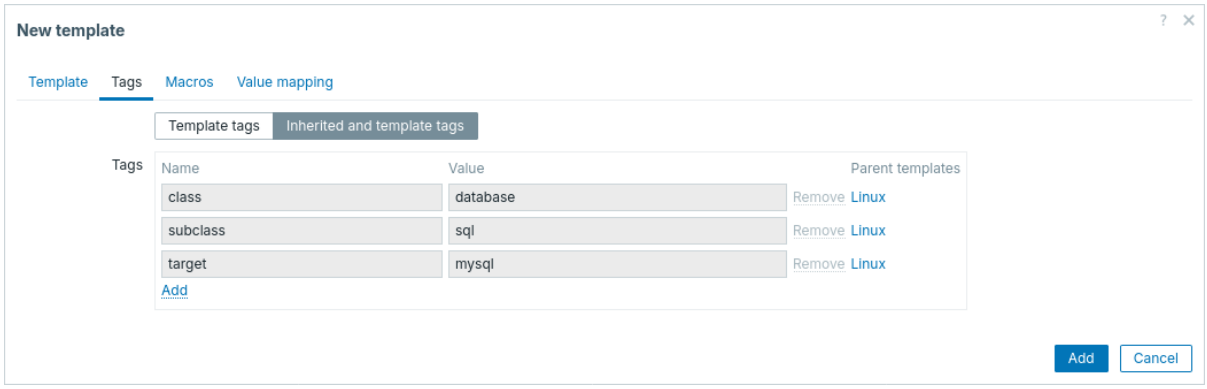
Paramètre	Description
Nom du modèle	Nom unique du modèle. Les caractères alphanumériques, les espaces, les points, les tirets et les traits de soulignement sont autorisés. Les espaces en début et en fin ne sont pas autorisés.
Nom visible	Si vous définissez ce nom, c'est celui qui sera visible dans les listes, les cartes, etc.
Modèles	Liez un ou plusieurs modèles à ce modèle. Toutes les entités (éléments, déclencheurs, etc.) seront héritées des modèles liés. Pour lier un nouveau modèle, saisissez son nom dans le champ *Modèles* ; une liste déroulante des modèles correspondants s'affichera. Vous pouvez également cliquer sur *Sélectionner* pour ouvrir la fenêtre contextuelle *Modèles*, qui n'affiche initialement aucun modèle. Pour remplir la liste, saisissez un nom de groupe de modèles dans le champ *Groupe de modèles* (puis sélectionnez-en un dans la liste suggérée) ou cliquez sur le bouton *Sélectionner* situé à côté du champ *Groupe de modèles* pour ouvrir une fenêtre contextuelle *Groupes de modèles*. Une fois un groupe de modèles choisi, la fenêtre contextuelle *Modèles* est mise à jour et affiche les modèles appartenant à ce groupe. Les modèles sélectionnés seront liés lors de l'enregistrement ou de la mise à jour de la configuration. Pour délier un modèle, utilisez l'une des deux options du bloc *Modèles* : *Délier* - délie le modèle, mais conserve ses entités (éléments, déclencheurs, etc.) ; *Délier et effacer* - délie le modèle et supprime toutes ses entités (éléments, déclencheurs, etc.).
Groupes de modèles	Groupes de modèles auxquels appartient le modèle.
Description	Description du modèle.

Paramètre	Description
Fournisseur et version	Fournisseur et version du modèle ; affichés uniquement lors de la mise à jour de modèles existants (modèles prêts à l'emploi fournis par Zabbix, modèles importés ou modèles modifiés via l'API Template) si la configuration du modèle contient ces informations. Ne peut pas être modifié dans l'interface Zabbix. Pour les modèles prêts à l'emploi, la version est affichée comme suit : version majeure de Zabbix, séparateur (« - »), numéro de révision (incrémenté à chaque nouvelle version du modèle et réinitialisé à chaque version majeure de Zabbix). Par exemple : 7.0-0, 7.0-5, 8.0-0, 8.0-3.

L'onglet **Étiquettes** permet de définir des étiquettes au niveau du modèle. Tous les problèmes des hôtes liés à ce modèle seront associés aux valeurs saisies ici.



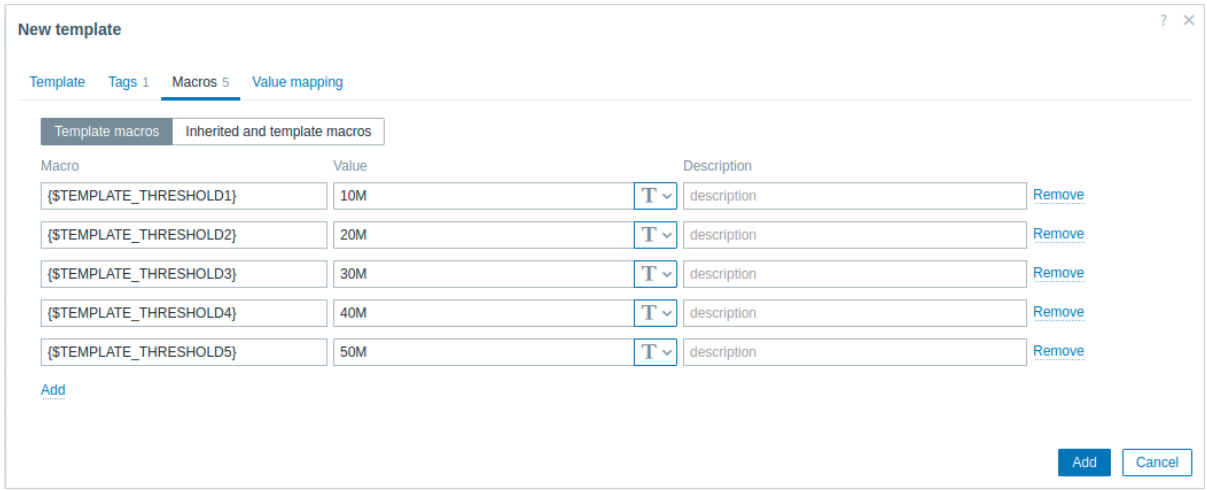
Si vous sélectionnez l'option *Étiquettes héritées et du modèle*, vous pouvez également afficher les étiquettes des modèles liés que le modèle héritera.



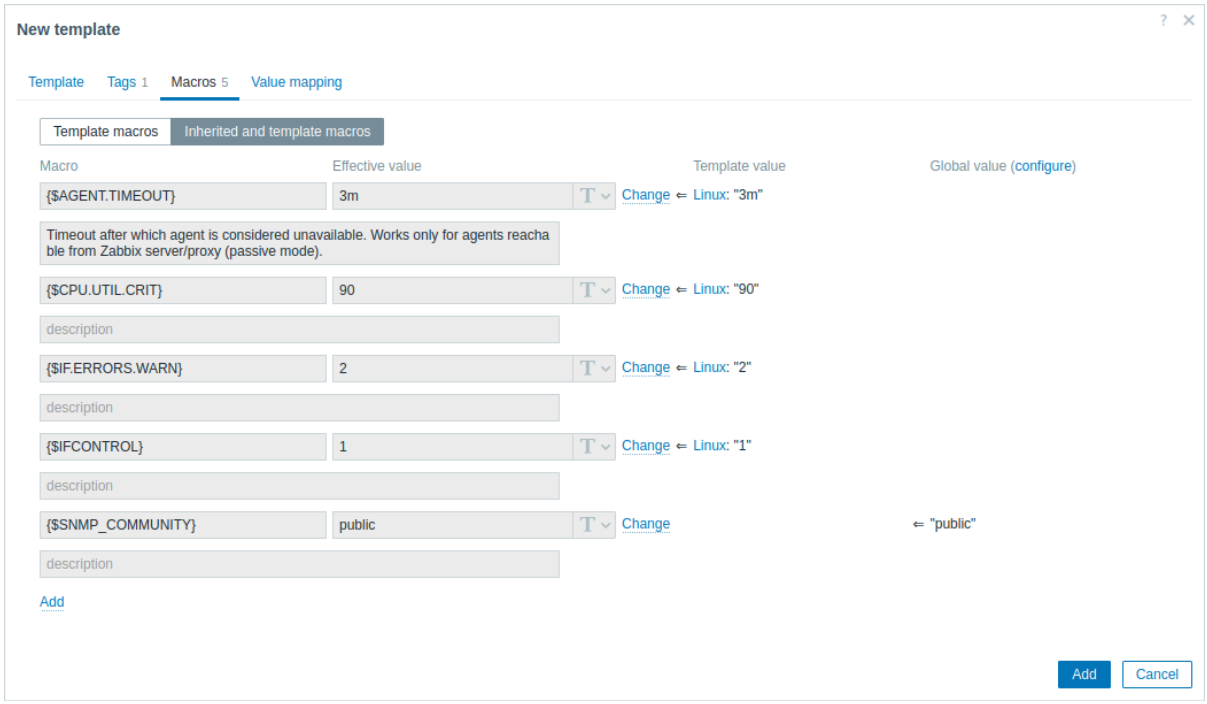
Pour plus de commodité, des liens vers les modèles correspondants sont fournis.

Les macros utilisateur, les macros {INVENTORY.*}, {HOST.HOST}, {HOST.NAME}, {HOST.CONN}, {HOST.DNS}, {HOST.IP}, {HOST.PORT} et {HOST.ID} sont prises en charge dans les étiquettes.

L'onglet **Macros** permet de définir des macros utilisateur au niveau du modèle sous forme de paires nom-valeur. Notez que les valeurs des macros peuvent être conservées sous forme de texte brut, de texte secret ou de secret Vault. Il est également possible d'ajouter une description.



Si vous sélectionnez l'option *Macros héritées et du modèle*, vous pouvez également afficher les macros des modèles liés et les macros globales que le modèle héritera, ainsi que les valeurs auxquelles les macros seront résolues.



Pour plus de commodité, des liens vers les modèles correspondants ainsi qu'un lien vers la configuration des macros globales sont fournis. Il est également possible de modifier une macro d'un modèle lié ou une macro globale au niveau du modèle, ce qui crée effectivement une copie de la macro sur le modèle.

L'onglet **Mappage des valeurs** permet de configurer une représentation conviviale des données des éléments dans les mappages de valeurs.

Boutons :

Add	Ajoute le modèle. Le modèle ajouté doit apparaître dans la liste.
Update	Met à jour les propriétés d'un modèle existant.
Clone	Crée un autre modèle à partir des propriétés du modèle actuel. Cela **inclut** les entités (éléments, déclencheurs, etc.) héritées des modèles liés ainsi que celles directement associées au modèle actuel, mais **exclut** le fournisseur et la version du modèle actuel afin que le modèle cloné puisse être distingué de l'original.
Delete	Supprime le modèle ; les entités du modèle (éléments, déclencheurs, etc.) restent associées aux hôtes liés.
Delete and clear	Supprime le modèle et toutes ses entités des hôtes liés.
Cancel	Annule la modification des propriétés du modèle.

Ajout d'éléments, de déclencheurs, de graphiques

> **Attention:**
> Les éléments doivent d'abord être ajoutés à un modèle. Les déclencheurs et les graphiques ne peuvent pas être ajoutés sans l'élément correspondant.

Vous pouvez ajouter des éléments existants à un modèle ou créer de nouveaux éléments.

Pour ajouter des éléments au modèle :

1. Accédez à *Collecte de données* > *Hôtes* (ou *Modèles*).
2. Cliquez sur *Éléments* dans la ligne de l'hôte/modèle requis.
3. Cochez les cases des éléments que vous souhaitez ajouter au modèle.
4. Cliquez sur *Copier* sous la liste des éléments.
5. Sélectionnez le modèle (ou le groupe de modèles) vers lequel les éléments doivent être copiés, puis cliquez sur *Copier*.
 Tous les éléments sélectionnés doivent être copiés vers le modèle.

L'ajout de déclencheurs et de graphiques se fait de manière similaire (à partir de la liste des déclencheurs et des graphiques respectivement), en gardant à l'esprit qu'ils ne peuvent être ajoutés que si les éléments requis ont d'abord été ajoutés.

Ajout de tableaux de bord

Pour ajouter des tableaux de bord à un modèle dans *Collecte de données* > *Modèles*, procédez comme suit :

1. Cliquez sur *Tableaux de bord* dans la ligne du modèle.
2. Configurez un tableau de bord en suivant les recommandations de configuration des tableaux de bord.

Attention:
Lors de la configuration des widgets sur un tableau de bord de modèle (au lieu d'un tableau de bord global), les paramètres liés à l'hôte ne sont pas disponibles, et certains paramètres ont une étiquette différente. Cela s'explique par le fait que les tableaux de bord de modèle affichent uniquement les données de l'hôte auquel le modèle est lié. Par exemple, les paramètres *Host groups*, *Exclude host groups* et *Hosts* du widget *Problems* ne sont pas disponibles, le paramètre *Host groups* du widget *Host availability* n'est pas disponible, et le paramètre *Show hosts in maintenance* est renommé *Show data in maintenance*, etc. Pour plus d'informations sur la disponibilité des paramètres dans les widgets de tableau de bord de modèle, consultez les paramètres spécifiques de chaque widget de tableau de bord.

Note:
Pour plus de détails sur l'accès aux tableaux de bord d'hôte créés à partir de tableaux de bord de modèle, consultez la section tableaux de bord d'hôte.

Configuration des règles de découverte de bas niveau

Consultez la section découverte de bas niveau du manuel.

Ajout de scénarios web

Pour ajouter des scénarios web à un modèle dans *Collecte de données* > *Modèles*, procédez comme suit :

1. Cliquez sur *Web* dans la ligne du modèle.
2. Configurez un scénario web en suivant la méthode habituelle de configuration des scénarios web.

Configuration d'un groupe de modèles

Aperçu

Les groupes de modèles sont utilisés pour le regroupement logique des modèles et pour l'attribution des permissions des utilisateurs sur ceux-ci.

Chaque modèle doit avoir au moins un groupe de modèles attribué. Un modèle peut appartenir à plusieurs groupes de modèles, et chaque groupe de modèles peut contenir plusieurs modèles.

Notez que, dans Zabbix, toutes les permissions sont basées sur des groupes : groupes d'utilisateurs, groupes d'hôtes et groupes de modèles. Ainsi, même si un seul utilisateur a besoin d'accéder à un seul modèle, cet accès est accordé en ajoutant l'utilisateur à un groupe d'utilisateurs et en attribuant à ce groupe d'utilisateurs des permissions sur le groupe de modèles contenant ce modèle.

Configuration

Attention:
Seuls les utilisateurs Super admin peuvent créer des groupes de modèles.

Il existe deux options pour créer un groupe de modèles dans l'interface Zabbix.

Première option :

1. Accédez à *Collecte de données* > *Groupes de modèles*.
2. Cliquez sur *Créer un groupe de modèles* dans le coin supérieur droit de l'écran.
3. Saisissez le nom du groupe dans le formulaire.



Deuxième option : lors de la configuration d'un modèle, saisissez un nom de groupe inexistant dans le champ de saisie *Groupes de modèles*.

Une fois le groupe de modèles créé, vous pouvez cliquer sur le nom du modèle dans la liste sous *Collecte de données* > *Groupes de modèles* pour modifier le nom du groupe, cloner le groupe ou supprimer le groupe.

La suppression d'un groupe de modèles supprime uniquement le groupe logique, et non les modèles qu'il contient. Il n'est pas possible de supprimer un groupe de modèles qui est le seul groupe d'un modèle existant.

Création de sous-groupes de modèles

Un sous-groupe de modèles (ou groupe de modèles imbriqué) est un enfant du groupe de modèles parent qui le contient.

Un sous-groupe est créé en utilisant la barre oblique « / » dans le champ de saisie du nom du groupe afin d'indiquer sa relation avec le ou les groupes parents. Par exemple :

- saisir `Linux servers/Databases` crée le sous-groupe `Linux servers/Databases` du groupe parent `Linux servers`.
- saisir `Linux servers/Databases/MySQL/Tokyo` crée le sous-groupe correspondant dans les groupes parents imbriqués `Linux servers`, `Linux servers/Databases`, `Linux servers/Databases/MySQL`.

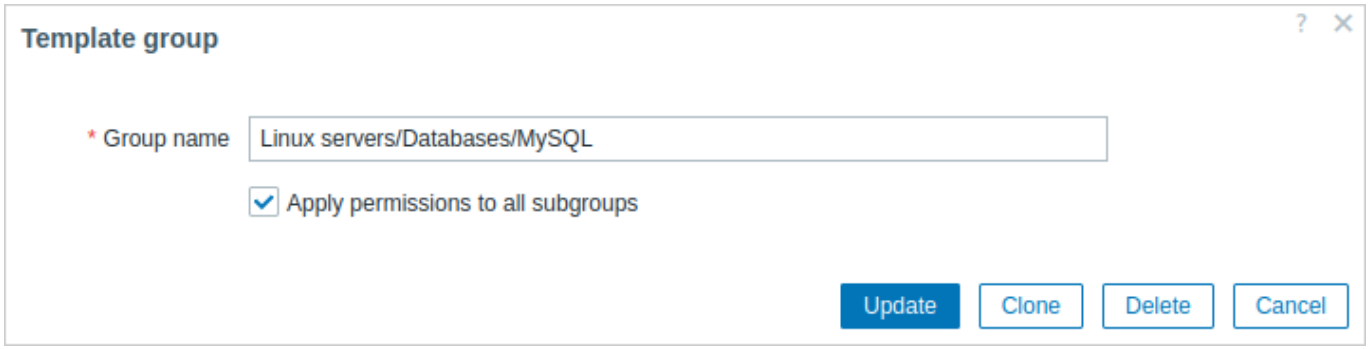
Lors de la création d'un sous-groupe, l'utilisation de barres obliques au début ou à la fin, ou de plusieurs barres obliques consécutives, n'est pas autorisée. L'échappement de « / » n'est pas pris en charge.

Il n'est pas nécessaire de créer un ou plusieurs groupes de modèles parents avant de créer un sous-groupe. Vous pouvez choisir de commencer par créer un sous-groupe (par exemple, `Linux servers/Databases`) ou un ou plusieurs groupes de modèles parents (dans notre exemple, `Linux servers`). Si vous commencez par créer un sous-groupe, le ou les groupes de modèles parents ne seront **pas** créés automatiquement.

Permissions pour les groupes de modèles

- Lors de la création d'un sous-groupe à partir d'un groupe de modèles parent existant (par exemple, la création de `Linux servers/Databases` alors que `Linux servers` existe déjà), les permissions du groupe d'utilisateurs sur le sous-groupe sont héritées du parent.
- Lors de la création d'un groupe de modèles parent à partir d'un sous-groupe existant (par exemple, la création de `Linux servers` alors que `Linux servers/Databases` existe déjà), aucune permission n'est définie pour le parent.

Lors de la modification de n'importe quel groupe de modèles, vous pouvez également définir une option supplémentaire, *Appliquer les permissions à tous les sous-groupes*.



En cochant cette case et en cliquant sur *Mettre à jour*, le même niveau de permissions sera appliqué à tous les sous-groupes actuels et futurs du groupe de modèles en cours de modification.

Ainsi, si des groupes d'utilisateurs se sont vu attribuer des permissions différentes pour les sous-groupes du groupe de modèles en cours de modification, le fait de cocher la case accordera à tous les sous-groupes actuels et futurs les mêmes permissions utilisateur que celles du groupe en cours de modification.

Notez que cette option n'est pas enregistrée dans la base de données et qu'elle remplacera les permissions existantes. Toute modification effectuée via cette option ne peut être annulée que manuellement.

Liaison/déliaison

Aperçu

La liaison est un processus par lequel des modèles sont appliqués aux hôtes, tandis que la dissociation supprime l'association du modèle avec un hôte.

Lier un modèle

Pour lier un modèle à l'hôte, procédez comme suit :

1. Accédez à *Collecte de données* > *Hôtes*.
2. Cliquez sur l'hôte requis.
3. Commencez à saisir le nom du modèle dans le champ *Modèles*. Une liste de modèles correspondants s'affiche ; faites-la défiler pour effectuer votre sélection. Vous pouvez également cliquer sur *Sélectionner* à côté du champ, puis sélectionner un ou plusieurs modèles dans la liste d'une fenêtre contextuelle.
4. Cliquez sur *Ajouter/Mettre à jour* dans le formulaire des attributs de l'hôte.

L'hôte dispose désormais de toutes les entités du modèle. Cela inclut les éléments, les déclencheurs, les graphiques, les règles de découverte de bas niveau, les scénarios web ainsi que les tableaux de bord.

Attention:
La liaison de plusieurs modèles au même hôte échouera si ces modèles contiennent des éléments ayant la même clé d'élément. De plus, comme les déclencheurs et les graphiques utilisent des éléments, ils ne peuvent pas non plus être liés à un même hôte depuis plusieurs modèles si des clés d'élément identiques sont utilisées.

Lorsque des entités (éléments, déclencheurs, etc.) sont ajoutées depuis le modèle :

- les entités identiques qui existaient déjà sur l'hôte sont mises à jour en tant qu'entités du modèle, et **toute personnalisation existante de l'entité au niveau de l'hôte est perdue** ;
- les entités du modèle sont ajoutées ;
- les entités liées directement qui, avant la liaison du modèle, existaient uniquement sur l'hôte restent inchangées.

Dans les listes, toutes les entités provenant du modèle sont désormais précédées du nom du modèle, indiquant qu'elles appartiennent à ce modèle particulier. Le nom du modèle lui-même (affiché en gris) est un lien qui permet d'accéder à la liste de ces entités au niveau du modèle.

Note:
Pour certains éléments, tels que les vérifications externes, les vérifications par agent HTTP, les vérifications simples, les vérifications SSH et les vérifications Telnet, une interface d'hôte est facultative. Si, au moment de la liaison d'un modèle, aucune interface n'est définie pour l'hôte, ces éléments seront ajoutés sans interface. Si vous ajoutez ultérieurement une interface à l'hôte, elle ne sera pas automatiquement attribuée aux éléments déjà existants. Pour attribuer la nouvelle interface de l'hôte à tous les éléments du modèle en une seule fois, dissociez le modèle de l'hôte, puis liez-le à nouveau. Pour conserver l'historique des éléments, utilisez l'option *Dissocier* et non *Dissocier et effacer*.

Si une entité n'est pas précédée du nom du modèle, cela signifie qu'elle existait sur l'hôte avant et qu'elle n'a pas été ajoutée par le modèle.

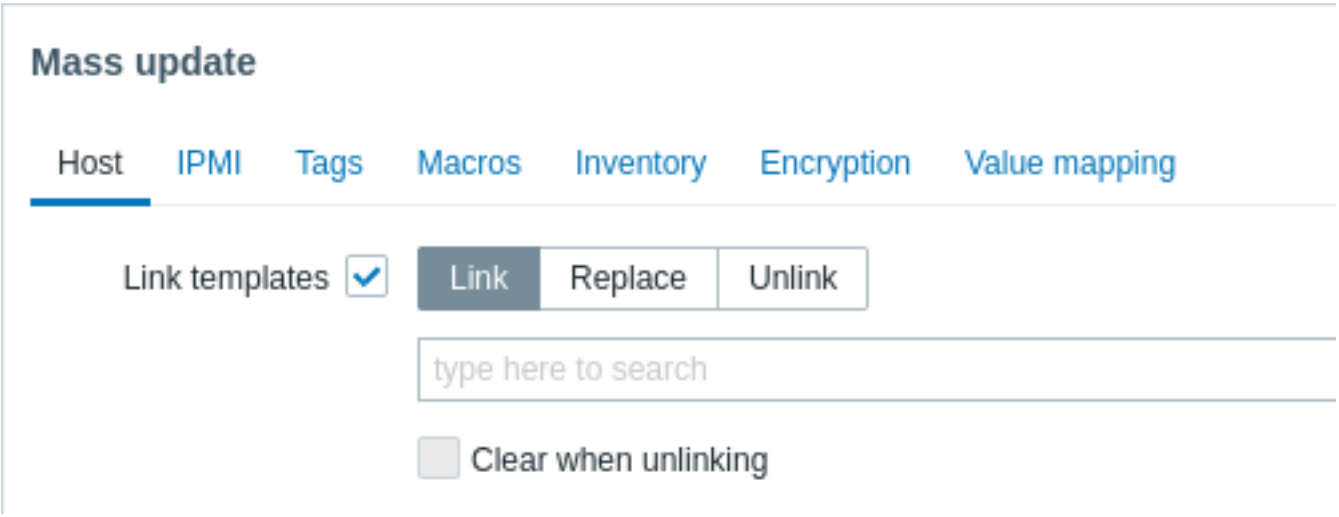
Critères d'unicité des entités

Lors de l'ajout d'entités (éléments, déclencheurs, etc.) à partir d'un modèle, il est important de savoir lesquelles de ces entités existent déjà sur l'hôte et doivent être mises à jour, et lesquelles sont différentes. Les critères d'unicité permettant de déterminer si les entités sont identiques ou différentes sont les suivants :

- pour les éléments - la clé de l'élément ;
- pour les déclencheurs - le nom du déclencheur et l'expression ;
- pour les graphiques personnalisés - le nom du graphique et ses éléments.

Liaison de modèles à plusieurs hôtes

Pour mettre à jour la liaison de modèles de nombreux hôtes, accédez à *Collecte de données* > *Hôtes*, sélectionnez certains hôtes en cochant leurs cases, puis cliquez sur **Mise à jour en masse** sous la liste et sélectionnez ensuite *Lier des modèles* :



Pour lier des modèles supplémentaires, commencez à saisir le nom du modèle dans le champ de saisie semi-automatique jusqu'à ce qu'une liste déroulante proposant les modèles correspondants apparaisse. Faites simplement défiler la liste pour sélectionner le modèle à lier.

L'option *Remplacer* vous permet de lier un nouveau modèle tout en déliant les modèles précédemment liés aux hôtes. L'option *Délier* vous permet de spécifier les modèles à délier. L'option *Effacer lors du déliement* vous permet non seulement de délier les modèles précédemment liés, mais également de supprimer toutes les entités qui en sont héritées (éléments, déclencheurs, etc.).

Note:
Zabbix propose un ensemble conséquent de modèles prédéfinis. Vous pouvez les utiliser comme référence, mais évitez de les utiliser sans modification en production, car ils peuvent contenir trop d'éléments et interroger les données trop fréquemment. Si vous souhaitez les utiliser, ajustez-les pour répondre à vos besoins réels.

Modification des entités liées

Si vous essayez de modifier un élément ou un déclencheur lié depuis le modèle, vous remarquerez peut-être que de nombreuses options clés sont désactivées pour la modification. Cela est logique, car l'idée des modèles est de permettre une modification centralisée au niveau du modèle. Cependant, vous pouvez tout de même, par exemple, activer/désactiver un élément sur des hôtes individuels et définir l'intervalle de mise à jour, la durée de conservation de l'historique ainsi que certains autres paramètres.

Si vous souhaitez modifier entièrement l'entité, vous devez la modifier au niveau du modèle (le raccourci vers le niveau du modèle est affiché dans le nom du formulaire), en gardant à l'esprit que ces modifications affecteront tous les hôtes auxquels ce modèle est lié.

Attention:
Toute personnalisation des entités effectuée au niveau du modèle remplacera les personnalisations précédentes des entités au niveau de l'hôte.

Dissocier un modèle

Pour dissocier un modèle d'un hôte, procédez comme suit :

1. Accédez à *Collecte de données* > *Hôtes*.
2. Cliquez sur l'hôte requis et repérez le champ *Modèles*.
3. Cliquez sur *Dissocier* ou *Dissocier et vider* à côté du modèle pour le dissocier.
4. Cliquez sur *Mettre à jour* dans le formulaire des attributs de l'hôte.

Le choix de l'option *Dissocier* supprimera simplement l'association avec le modèle, tout en conservant sur l'hôte toutes les entités qui en dépendent. Cela inclut les éléments, les déclencheurs, les graphiques, les règles de découverte de bas niveau et les scénarios web, mais exclut les tableaux de bord. Notez que les correspondances de valeurs et les balises héritées des modèles liés seront également supprimées.

Le choix de l'option *Dissocier et vider* supprimera à la fois l'association avec le modèle et toutes ses entités (éléments, déclencheurs, etc.).

Imbrication

Vue d'ensemble

L'imbrication est une méthode permettant à un modèle d'en englober un ou plusieurs autres.

Comme il est logique de séparer les entités dans des modèles individuels pour différents services, applications, etc., vous pouvez vous retrouver avec un grand nombre de modèles, qui devront tous être liés à de nombreux hôtes. Pour simplifier la situation, il est possible de lier plusieurs modèles entre eux dans un seul modèle.

L'avantage de l'imbrication est que vous n'avez à lier qu'un seul modèle à l'hôte, et l'hôte héritera automatiquement de toutes les entités des modèles liés à ce modèle. Par exemple, si nous lions *T1* et *T2* à *T3*, nous complétons *T3* avec toutes les entités de *T1* et *T2*, mais pas l'inverse. Si nous lions *T1* à *T2* et *T3*, nous complétons *T2* et *T3* avec les entités de *T1*.

Configuration des modèles imbriqués

Pour lier des modèles, vous devez prendre un modèle existant (ou en créer un nouveau), puis :

1. Ouvrez le formulaire de configuration du modèle.
2. Repérez le champ *Modèles*.
3. Cliquez sur *Sélectionner* pour ouvrir la fenêtre contextuelle *Modèles*.
4. Dans la fenêtre contextuelle, choisissez les modèles requis, puis cliquez sur *Sélectionner* pour ajouter les modèles à la liste.
5. Cliquez sur *Ajouter* ou *Mettre à jour* dans le formulaire de configuration du modèle.

Ainsi, toutes les entités du modèle configuré, ainsi que toutes les entités des modèles liés, apparaîtront désormais dans la configuration du modèle. Cela inclut les éléments, les déclencheurs, les graphiques, les règles de découverte de bas niveau et les scénarios web, mais exclut les tableaux de bord. Cependant, les tableaux de bord des modèles liés seront néanmoins hérités par les hôtes.

Pour dissocier l'un des modèles liés, cliquez sur *Dissocier* ou *Dissocier et effacer* dans le formulaire de configuration du modèle, puis cliquez sur *Mettre à jour*.

L'option *Dissocier* supprimera simplement l'association avec le modèle lié, sans supprimer aucune de ses entités (éléments, déclencheurs, etc.).

L'option *Dissocier et effacer* supprimera à la fois l'association avec le modèle lié et toutes ses entités (éléments, déclencheurs, etc.).

Mise à jour en masse

Aperçu

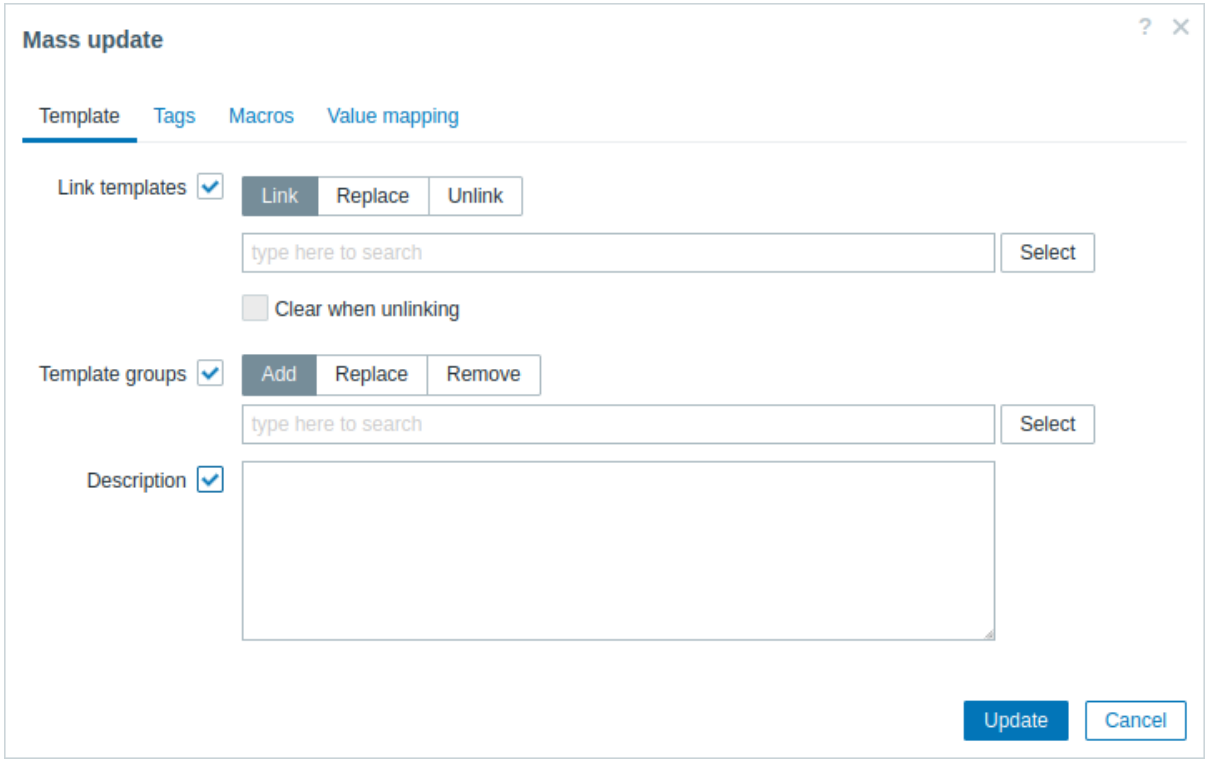
Parfois, vous souhaiterez peut-être modifier un attribut pour plusieurs modèles à la fois. Au lieu d'ouvrir chaque modèle individuel pour le modifier, vous pouvez utiliser la fonction de mise à jour en masse pour cela.

Utilisation de la mise à jour en masse

Pour mettre à jour en masse certains modèles, procédez comme suit :

1. Cochez les cases devant les modèles que vous souhaitez mettre à jour dans la liste des modèles.
2. Cliquez sur *Mise à jour en masse* sous la liste.
3. Accédez à l'onglet contenant les attributs requis (*Modèle*, *Tags*, *Macros* ou *Mappage de valeurs*).
4. Cochez les cases des attributs à mettre à jour et saisissez une nouvelle valeur pour chacun d'eux.

L'onglet **Modèle** contient les options générales de mise à jour en masse des modèles.



Les options suivantes sont disponibles lors de la sélection du bouton correspondant pour la mise à jour des *modèles liés* :

- *Lier* - indiquez les modèles supplémentaires à lier ;
- *Remplacer* - indiquez les modèles à lier tout en déliant les modèles précédemment liés ;
- *Délier* - indiquez les modèles à délier.

Pour indiquer les modèles à lier/délier, commencez à saisir le nom du modèle dans le champ de saisie semi-automatique jusqu'à ce qu'une liste déroulante proposant les modèles correspondants apparaisse. Faites simplement défiler la liste pour sélectionner les modèles requis.

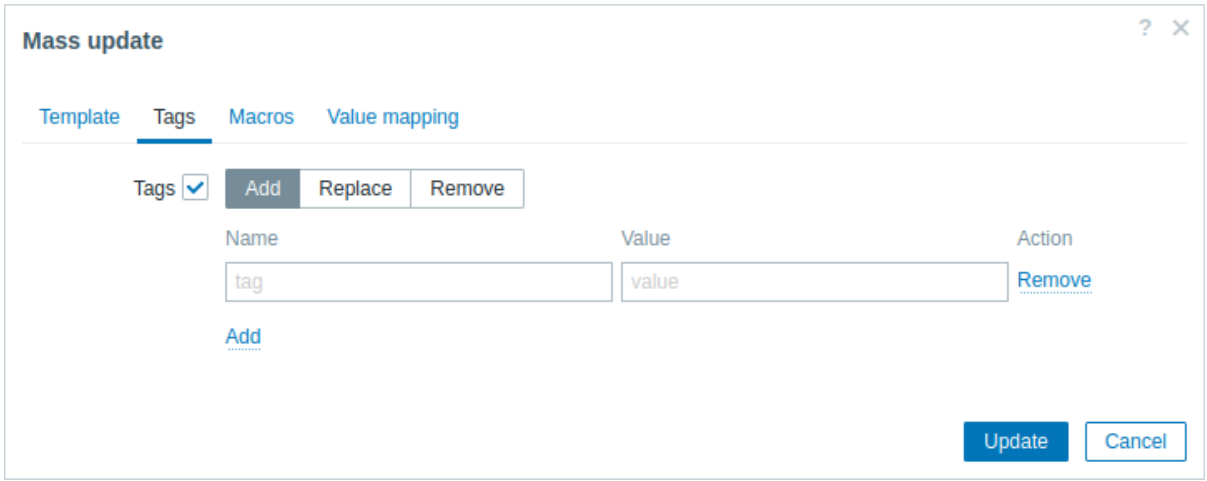
L'option *Effacer lors du déliement* vous permet non seulement de délier les modèles précédemment liés, mais également de supprimer toutes les entités héritées de ces modèles (éléments, déclencheurs, graphiques, etc.).

Les options suivantes sont disponibles lors de la sélection du bouton correspondant pour la mise à jour des *groupes de modèles* :

- *Ajouter* - indiquez des groupes de modèles supplémentaires parmi ceux existants ou saisissez des groupes de modèles entièrement nouveaux pour les modèles ;
- *Remplacer* - supprimez le modèle de tous les groupes de modèles existants et remplacez-les par le ou les groupes indiqués dans ce champ (groupes de modèles existants ou nouveaux) ;
- *Supprimer* - supprimez des groupes de modèles spécifiques des modèles.

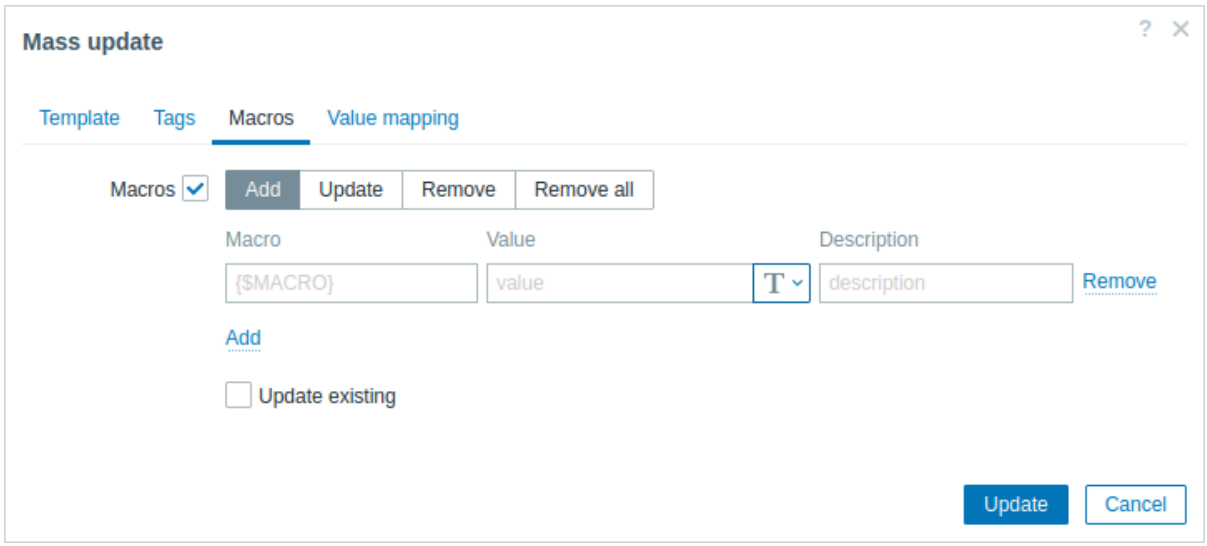
Ces champs proposent une saisie semi-automatique : commencez à y saisir du texte pour afficher une liste déroulante des groupes de modèles correspondants. Si le groupe de modèles est nouveau, il apparaît également dans la liste déroulante et est indiqué par *(nouveau)* après le texte. Faites simplement défiler la liste pour le sélectionner.

L'onglet **Tags** vous permet de mettre à jour en masse les tags au niveau du modèle.



Les macros utilisateur, les macros {INVENTORY.*}, {HOST.HOST}, {HOST.NAME}, {HOST.CONN}, {HOST.DNS}, {HOST.IP}, {HOST.PORT} et {HOST.ID} sont prises en charge dans les tags. Notez que les tags portant le même nom, mais des valeurs différentes, ne sont pas considérés comme des « doublons » et peuvent être ajoutés au même modèle.

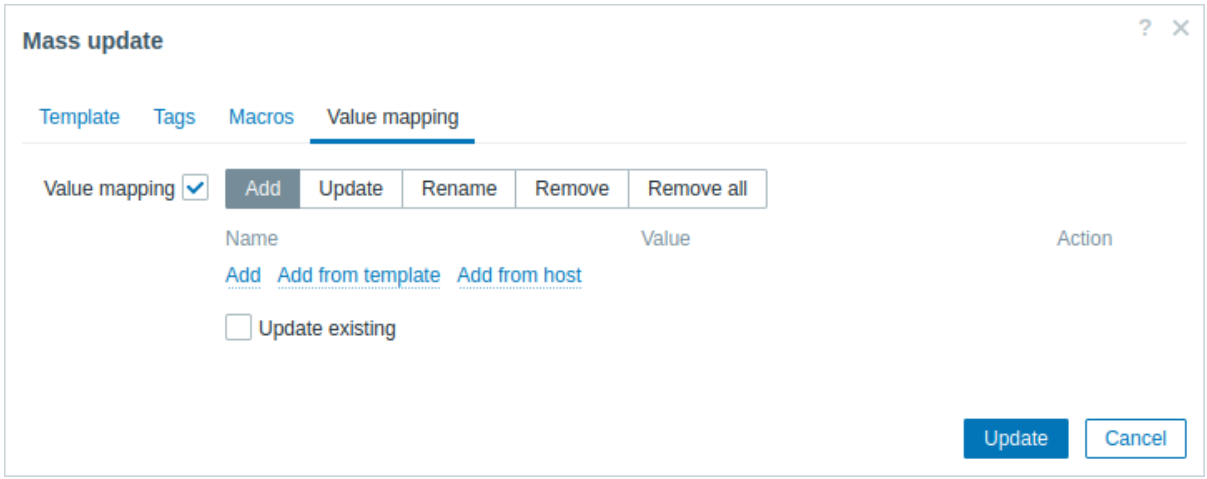
L'onglet **Macros** permet de mettre à jour en masse les macros au niveau des modèles.



Les options suivantes sont disponibles lors de la sélection du bouton correspondant pour la mise à jour des macros :

- *Ajouter* - spécifier des macros utilisateur supplémentaires pour les modèles. Si la case *Mettre à jour les éléments existants* est cochée, la valeur, le type et la description du nom de macro spécifié seront mis à jour. Si elle n'est pas cochée, si une macro portant ce nom existe déjà sur le ou les modèles, elle ne sera pas mise à jour.
- *Mettre à jour* - remplacer les valeurs, les types et les descriptions des macros spécifiées dans cette liste. Si la case *Ajouter les éléments manquants* est cochée, la macro qui n'existait pas auparavant sur un modèle sera ajoutée en tant que nouvelle macro. Si elle n'est pas cochée, seules les macros qui existent déjà sur un modèle seront mises à jour.
- *Supprimer* - supprimer les macros spécifiées des modèles. Si la case *À l'exception des éléments sélectionnés* est cochée, toutes les macros à l'exception de celles spécifiées dans la liste seront supprimées. Si elle n'est pas cochée, seules les macros spécifiées dans la liste seront supprimées.
- *Tout supprimer* - supprimer toutes les macros utilisateur des modèles. Si la case *Je confirme la suppression de toutes les macros* n'est pas cochée, une nouvelle fenêtre contextuelle s'ouvrira pour demander de confirmer la suppression de toutes les macros.

L'onglet **Mappage des valeurs** permet de mettre à jour en masse les mappages de valeurs.



Les boutons proposant les options suivantes sont disponibles pour mettre à jour les mappages de valeurs :

- *Ajouter* - ajouter des mappages de valeurs aux modèles. Si vous cochez *Mettre à jour les éléments existants*, toutes les propriétés du mappage de valeurs portant ce nom seront mises à jour. Sinon, si un mappage de valeurs portant ce nom existe déjà, il ne sera pas mis à jour.
- *Mettre à jour* - mettre à jour les mappages de valeurs existants. Si vous cochez *Ajouter les éléments manquants*, un mappage de valeurs qui n'existait pas auparavant sur un modèle sera ajouté en tant que nouveau mappage de valeurs. Sinon, seuls les mappages de valeurs qui existent déjà sur un modèle seront mis à jour.
- *Renommer* - attribuer un nouveau nom à un mappage de valeurs existant.

- *Supprimer* - supprimer les mappages de valeurs spécifiés des modèles. Si vous cochez *Sauf ceux sélectionnés*, tous les mappages de valeurs seront supprimés **à l'exception** de ceux qui sont spécifiés.
- *Tout supprimer* - supprimer tous les mappages de valeurs des modèles. Si la case *Je confirme la suppression de tous les mappages de valeurs* n'est pas cochée, une nouvelle fenêtre contextuelle s'ouvrira pour demander de confirmer la suppression.

Les options *Ajouter depuis un modèle* et *Ajouter depuis un hôte* sont disponibles pour les opérations d'ajout et de mise à jour des mappages de valeurs.
Elles permettent de sélectionner des mappages de valeurs depuis un modèle ou un hôte, respectivement.

Lorsque toutes les modifications requises ont été effectuées, cliquez sur *Update*. Les attributs seront mis à jour en conséquence pour tous les modèles sélectionnés.

Hôtes et groupes d'hôtes

Qu'est-ce qu'un « hôte » ?

Dans Zabbix, un « hôte » désigne tout dispositif physique ou virtuel, application, service ou toute autre collection de paramètres surveillés logiquement liée.

La création d'hôtes est l'une des premières tâches de supervision dans Zabbix. Par exemple, si vous souhaitez surveiller certains paramètres sur un serveur « x », vous devez d'abord créer un hôte appelé, par exemple, « Server X », puis vous pourrez y ajouter des éléments de supervision.

Les hôtes sont organisés en groupes d'hôtes.

Passez à :

- Assistant hôte pour une interface guidée, étape par étape, permettant de créer et de configurer un hôte nouveau ou existant.
- Configuration d'un hôte pour une approche classique de la création et de la configuration d'un hôte.

Assistant de configuration de l'hôte

Vue d'ensemble

L'assistant d'hôte est une interface guidée, étape par étape, pour configurer votre cible de surveillance (périphérique, application, service, etc.) dans Zabbix. Il vous guide à travers les étapes suivantes :

- Sélection d'un modèle
- Création ou sélection d'un hôte
- Installation de l'agent Zabbix ou d'agent 2
- Ajout d'une interface d'hôte
- Application d'une configuration supplémentaire à votre cible de surveillance ou à l'hôte Zabbix (si requis par le modèle)

Pour accéder à l'assistant d'hôte dans l'interface Zabbix, procédez comme suit :

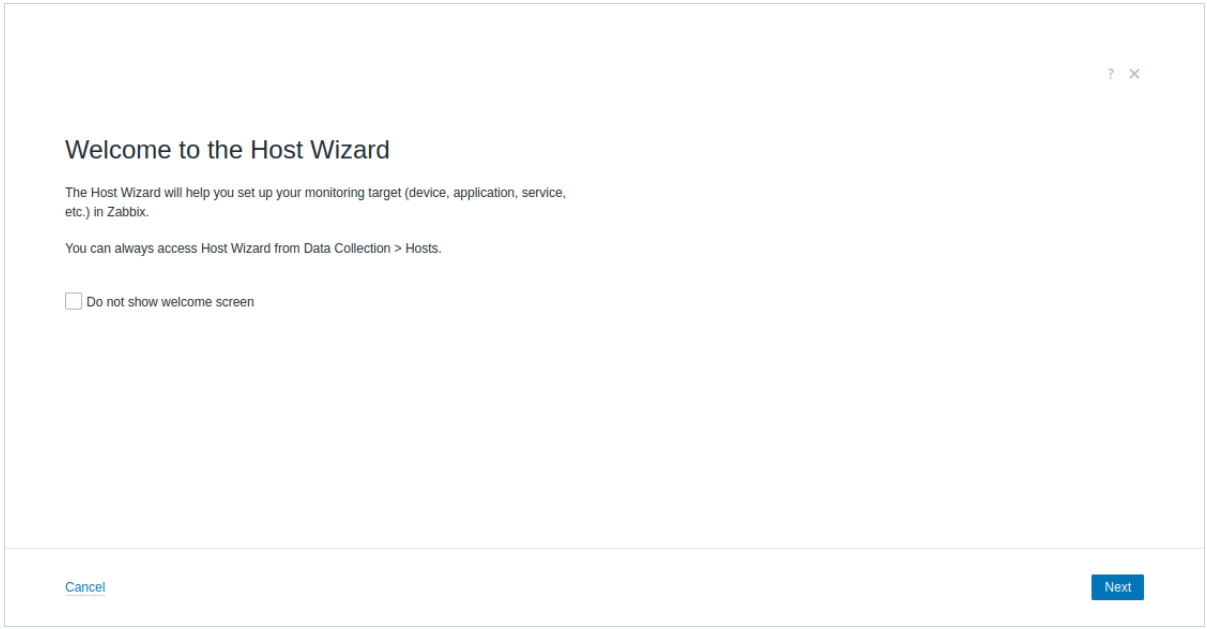
1. Allez à : *Collecte de données* > *Hôtes*.
2. Cliquez sur *Assistant d'hôte* dans le coin supérieur droit de l'écran.
3. Suivez les instructions de l'assistant.

Vous pouvez également modifier un hôte existant en cliquant sur son nom (ou sur l'icône à trois points à côté) dans différentes sections de l'interface afin d'accéder à l'assistant d'hôte depuis le menu de l'hôte. Notez que certaines étapes sont ignorées lors de la modification d'un hôte existant s'il inclut déjà la configuration requise par le modèle sélectionné.

Vous pouvez aussi utiliser l'approche classique pour créer et configurer un hôte.

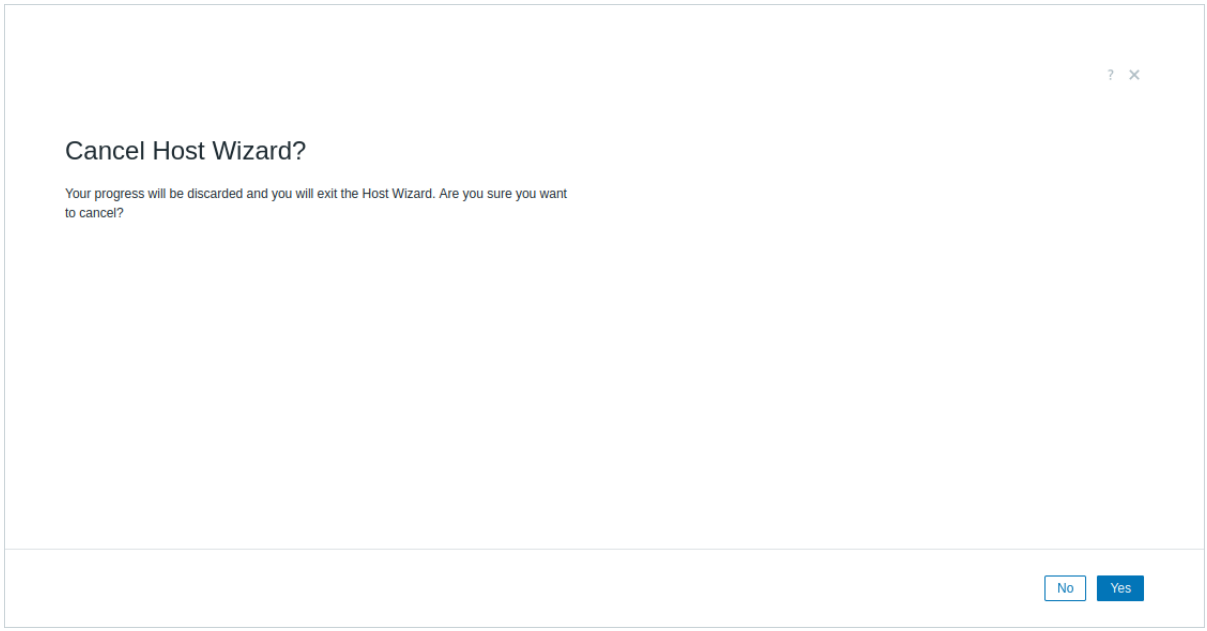
Bienvenue dans l'assistant d'hôte

Lorsque vous démarrez l'assistant d'hôte, un écran de bienvenue s'affiche :



Pour ignorer cet écran lors des prochaines sessions, vous pouvez cocher la case *Ne pas afficher l'écran de bienvenue* et cliquer sur *Suivant*.

Si vous commencez à configurer des paramètres dans les étapes suivantes de l'assistant d'hôte et tentez de fermer l'assistant en cliquant sur l'icône de fermeture ou en appuyant sur Esc, un écran de confirmation s'affichera :



Cliquer sur *Oui* quittera l'assistant d'hôte sans enregistrer votre progression. Cliquer sur *Non*, Esc ou sur l'icône de fermeture vous ramènera à la dernière étape.

Sélectionner un modèle

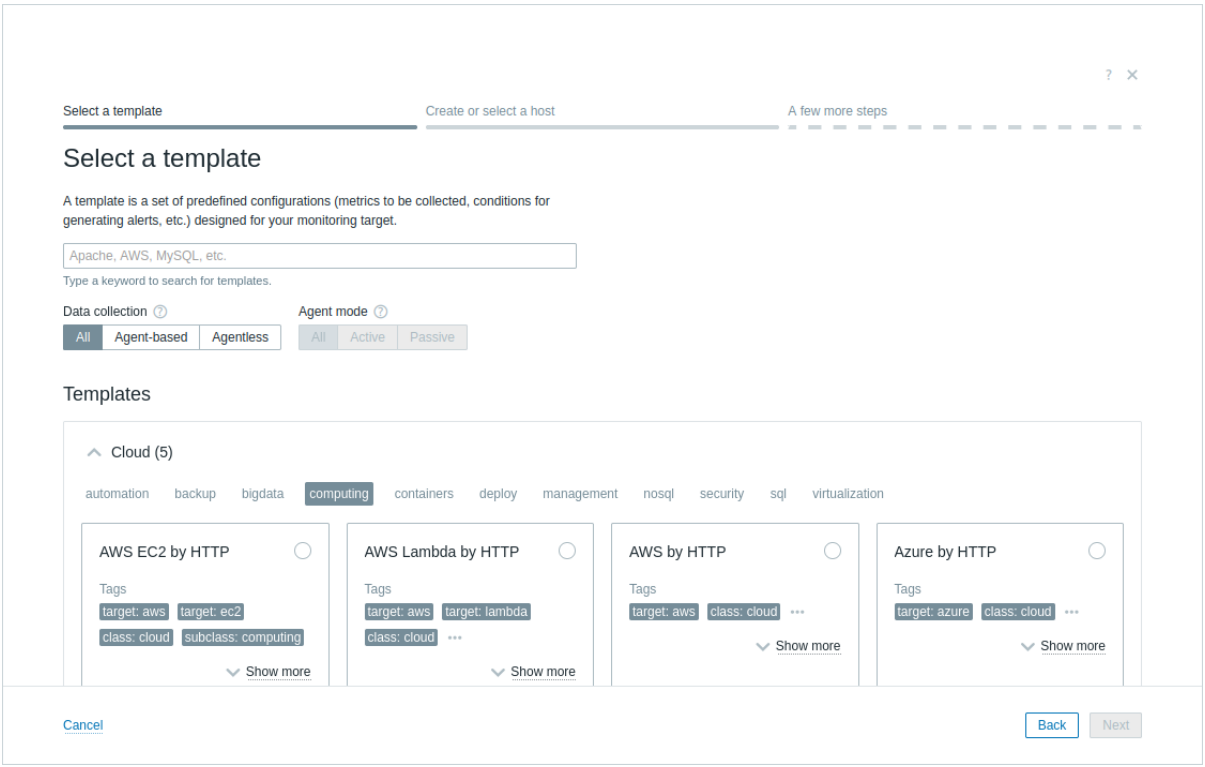
La première étape de la configuration de votre cible de supervision consiste à sélectionner un modèle, c'est-à-dire un ensemble de configurations prédéfinies (métriques à collecter, conditions de génération des alertes, etc.) conçues pour votre cible de supervision.

Les modèles prêts à l'emploi de Zabbix proposent diverses configurations de supervision prédéfinies pour les systèmes d'exploitation, les applications, les bases de données, les périphériques réseau, les services, etc.

Lors de cette étape, vous pouvez :

- Parcourir les modèles par classe (Cloud, Base de données, Réseau, etc.) et sous-classe (par exemple, automatisation, sauvegarde), qui sont basées sur les étiquettes des modèles.
- Rechercher des modèles à l'aide de mots-clés présents dans le nom du modèle ou dans les noms et valeurs de ses étiquettes.
- Filtrer les modèles selon la méthode de collecte des données (avec agent ou sans agent ; les modèles avec agent incluent au moins un élément Zabbix agent).

- Filtrer les modèles selon le mode de l'agent (actif ou passif ; les modèles actifs incluent au moins un élément Zabbix agent (actif), tandis que les modèles passifs incluent au moins un élément passif ; pour plus de détails, consultez Vérifications passives et actives de l'agent).
- Lors de l'utilisation de l'assistant d'hôte pour configurer un hôte existant, vous pouvez également filtrer les modèles déjà liés à l'hôte.



Attention:
Lors de la sélection d'un modèle, le message suivant peut s'afficher : *Certains modèles (n) sont incompatibles avec l'assistant d'hôte. Découvrez comment les mettre à jour. Les modèles personnalisés ne sont pas pris en charge.* Cela indique que certains modèles prêts à l'emploi ne sont pas encore compatibles avec l'assistant d'hôte et doivent être mis à niveau ; consultez les instructions de mise à niveau des modèles.

Le modèle que vous sélectionnez déterminera les étapes suivantes de l'assistant d'hôte. Par exemple, si vous sélectionnez le modèle *MySQL by Zabbix agent*, qui utilise Zabbix agent pour la collecte des données, l'assistant vous guidera tout au long du processus d'installation de l'agent.

Créer ou sélectionner un hôte

Un modèle doit être lié à un hôte, qui est une entité dans Zabbix représentant votre cible de supervision. Une fois lié, l'hôte reçoit toutes les entités du modèle, telles que les éléments (métriques à collecter) et les déclencheurs (conditions de génération des alertes), nécessaires à la supervision.

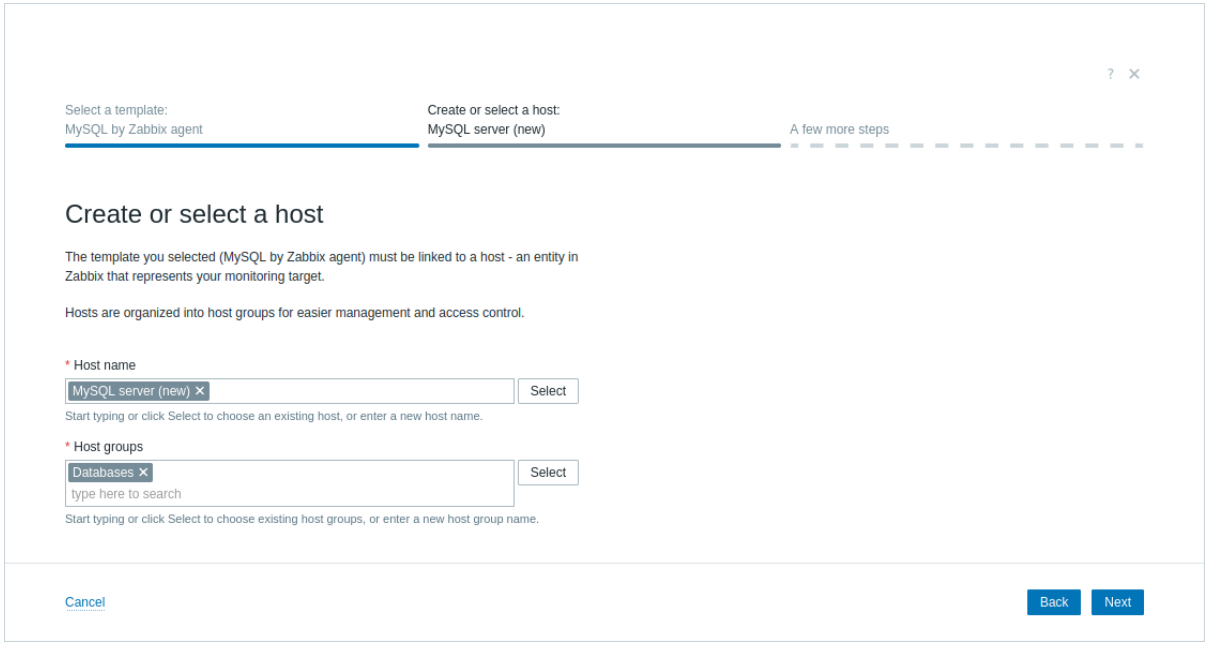
Chaque hôte doit également appartenir à au moins un groupe d'hôtes, utilisé pour organiser les hôtes et leur attribuer des permissions utilisateur.

À cette étape, vous pouvez :

- Créer un nouvel hôte et un groupe d'hôtes en saisissant leurs noms.
- Créer un nouvel hôte en saisissant son nom et sélectionner un groupe d'hôtes existant auquel le nouvel hôte sera affecté.
- Sélectionner un hôte existant ; vous pouvez également affecter cet hôte à des groupes d'hôtes supplémentaires sans le retirer de ses groupes actuels.

Note:
La sélection d'un hôte existant ne supprime pas et ne remplace pas sa configuration actuelle, sauf indication explicite dans les étapes suivantes. Les hôtes découverts ne peuvent pas être sélectionnés.

Par exemple, vous pouvez créer un nouvel hôte nommé *Serveur MySQL* pour représenter le serveur MySQL installé localement que vous souhaitez superviser. Le modèle *MySQL by Zabbix agent* précédemment sélectionné sera lié à cet hôte. Vous pouvez également sélectionner le groupe d'hôtes existant *Databases* (ou en créer un nouveau, par exemple *Serveurs MySQL*) afin d'organiser l'hôte avec d'autres hôtes liés aux bases de données et de simplifier ultérieurement la gestion des permissions.



Installer l'agent Zabbix

Les modèles basés sur un agent (tels que *MySQL by Zabbix agent*) nécessitent l'installation de l'agent Zabbix ou de l'agent 2 Zabbix - un processus déployé sur votre cible de supervision pour surveiller activement les ressources et applications locales (si le modèle l'exige).

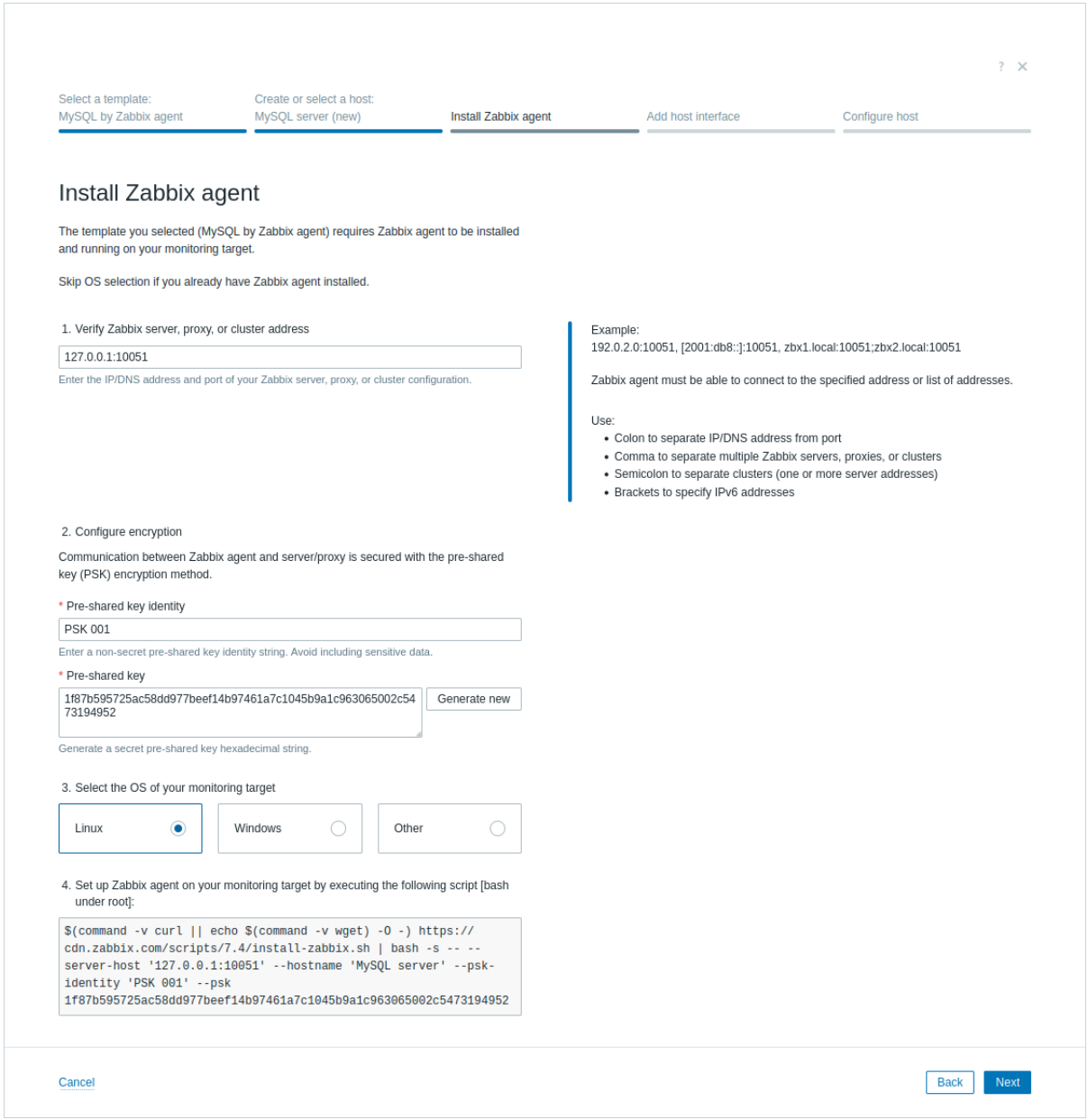
Dans cette étape :

- Vérifiez l'adresse du serveur Zabbix, du proxy ou du cluster (par exemple, 192.0.2.1:10051). Cette valeur sera utilisée pour configurer les paramètres Server et ServerActive de l'agent Zabbix.
- Saisissez une identité de clé prépartagée nouvelle, unique et non secrète (par exemple, *PSK 001* ou *mysql-agent-psk1*), puis générez la clé prépartagée. Ces valeurs seront utilisées pour configurer le chiffrement par clé prépartagée (PSK) du hôte Zabbix (par exemple, *serveur MySQL*) et de l'agent Zabbix.

Attention:
Lors de la modification d'un hôte existant, la configuration de la PSK remplacera tous les paramètres de chiffrement existants sur l'hôte.

- Sélectionnez le système d'exploitation de la machine qui héberge votre cible de supervision.
- Installez l'agent Zabbix sur ce système en exécutant le script fourni sur celui-ci ou en suivant les instructions d'installation.

Par exemple, vous pouvez définir l'*identité de la clé prépartagée* sur *PSK 001* et générer une nouvelle *clé prépartagée*. Vous pouvez ensuite sélectionner *Linux* comme système d'exploitation et exécuter le script fourni sur ce système. Après avoir installé l'agent Zabbix, retournez à l'assistant de création d'hôte.

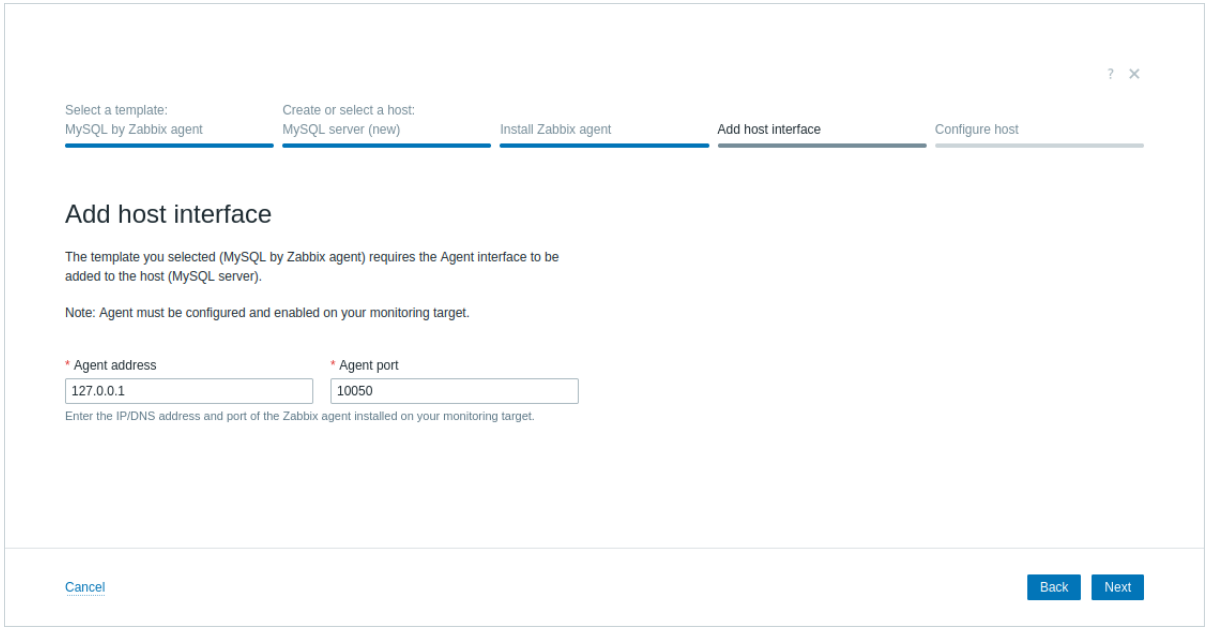


Ajouter une interface d'hôte

L'interface d'hôte définit comment le serveur Zabbix se connecte à votre cible de supervision via le réseau. Elle définit des paramètres de connexion tels que l'adresse IP, le nom DNS, le port et le type d'interface (Agent, SNMP, JMX ou IPMI), selon la méthode de collecte de données requise par le modèle sélectionné.

À cette étape, saisissez les détails de l'interface d'hôte requis par le modèle sélectionné.

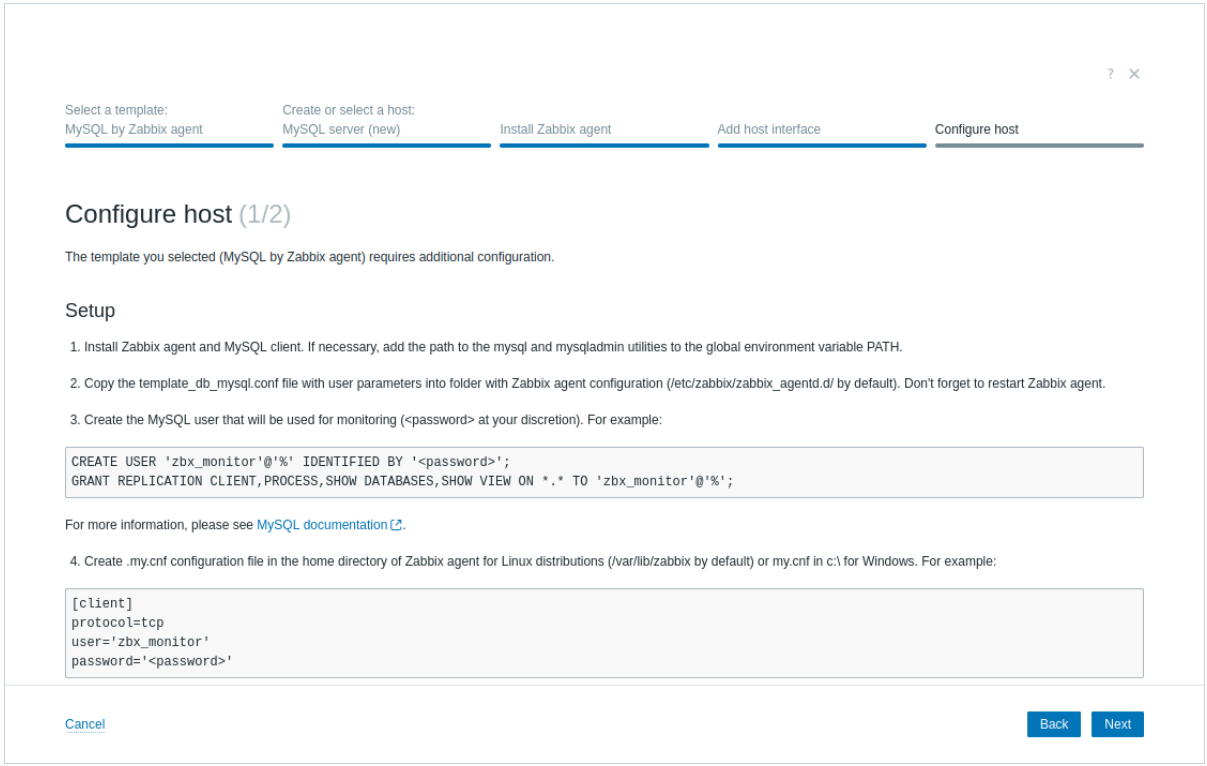
Par exemple, vous pouvez utiliser l'adresse agent par défaut (`127.0.0.1`) et le port (10050) si le serveur Zabbix, l'agent Zabbix et le serveur MySQL s'exécutent tous localement sur la même machine.



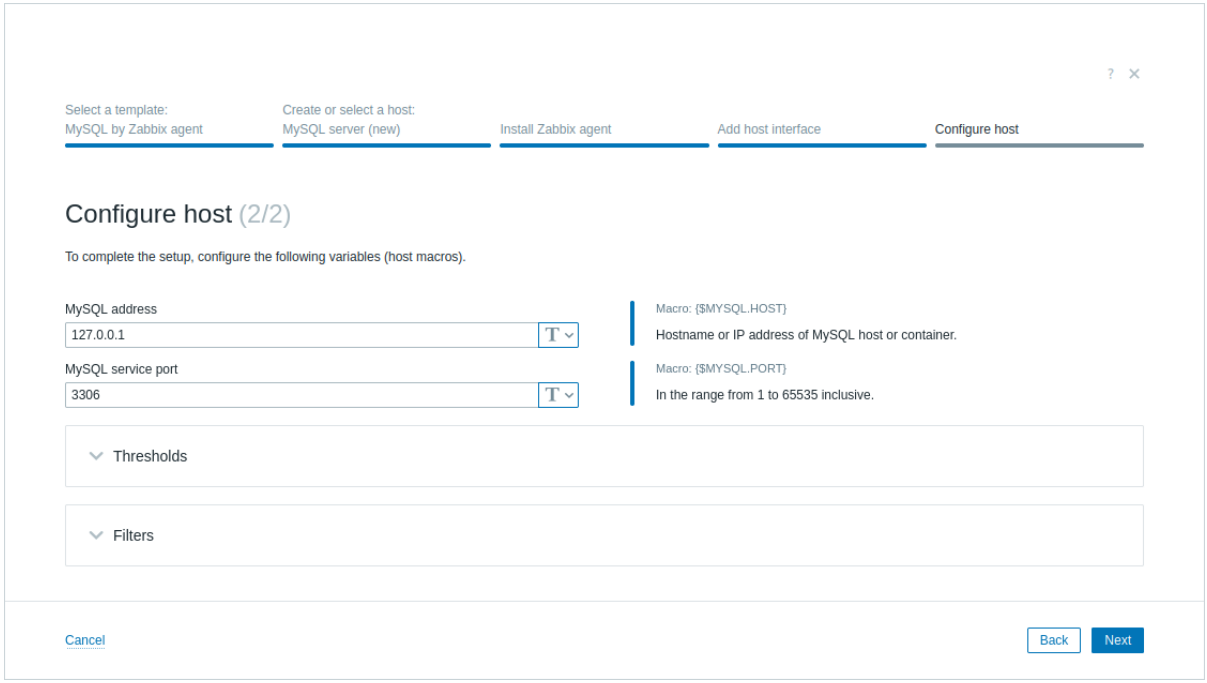
Configurer l'hôte

Certains modèles nécessitent une configuration supplémentaire avant de pouvoir créer l'hôte. Il peut notamment s'agir des opérations suivantes :

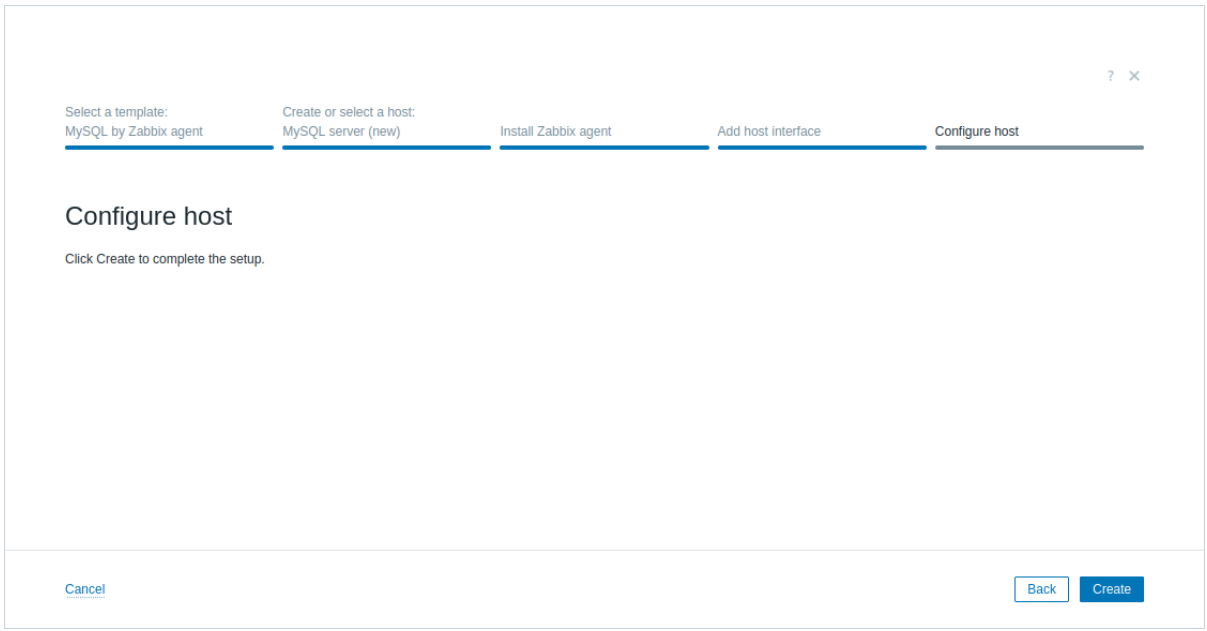
- Configuration manuelle de votre cible de supervision (par exemple, activation de services spécifiques, création d'utilisateurs de service ou attribution d'autorisations) :



- Définition des macros d'hôte - variables qui contrôlent le comportement des éléments, les paramètres de connexion, les identifiants d'authentification et d'autres paramètres :



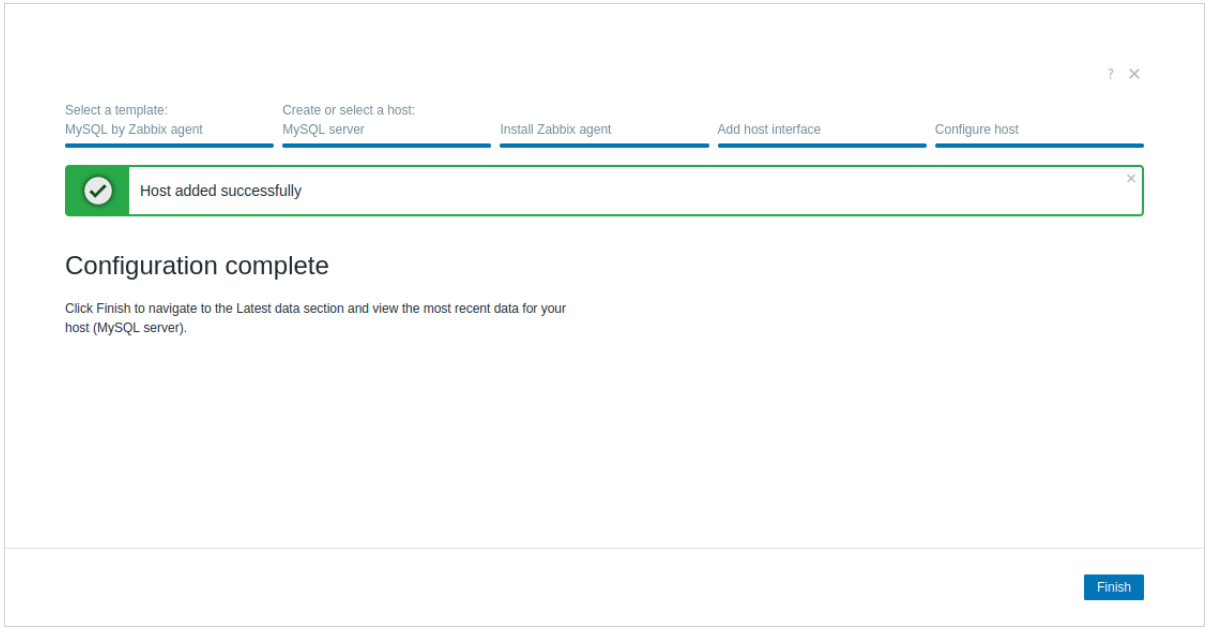
Après avoir effectué toutes les étapes de configuration requises, cliquez sur *Créer* pour ajouter l'hôte à Zabbix (ou sur *Mettre à jour* pour mettre à jour la configuration d'un hôte existant) :



Configuration terminée

À ce stade, Zabbix surveille déjà votre cible.

Pour les nouveaux hôtes, cliquez sur *Terminer* pour accéder à la section Dernières données et afficher les données les plus récentes de votre hôte. Pour les hôtes existants, cliquez sur *Terminer* pour fermer l'assistant d'hôte et revenir à l'écran depuis lequel il a été ouvert.



Configuration d'un hôte

Vue d'ensemble

Pour configurer un hôte dans l'interface Zabbix, procédez comme suit :

1. Accédez à *Collecte de données* > *Hôtes* ou *Supervision* > *Hôtes*.
2. Cliquez sur *Créer un hôte* dans le coin supérieur droit de l'écran (ou sur le nom de l'hôte pour modifier un hôte existant).
3. Saisissez les paramètres de l'hôte dans le formulaire.

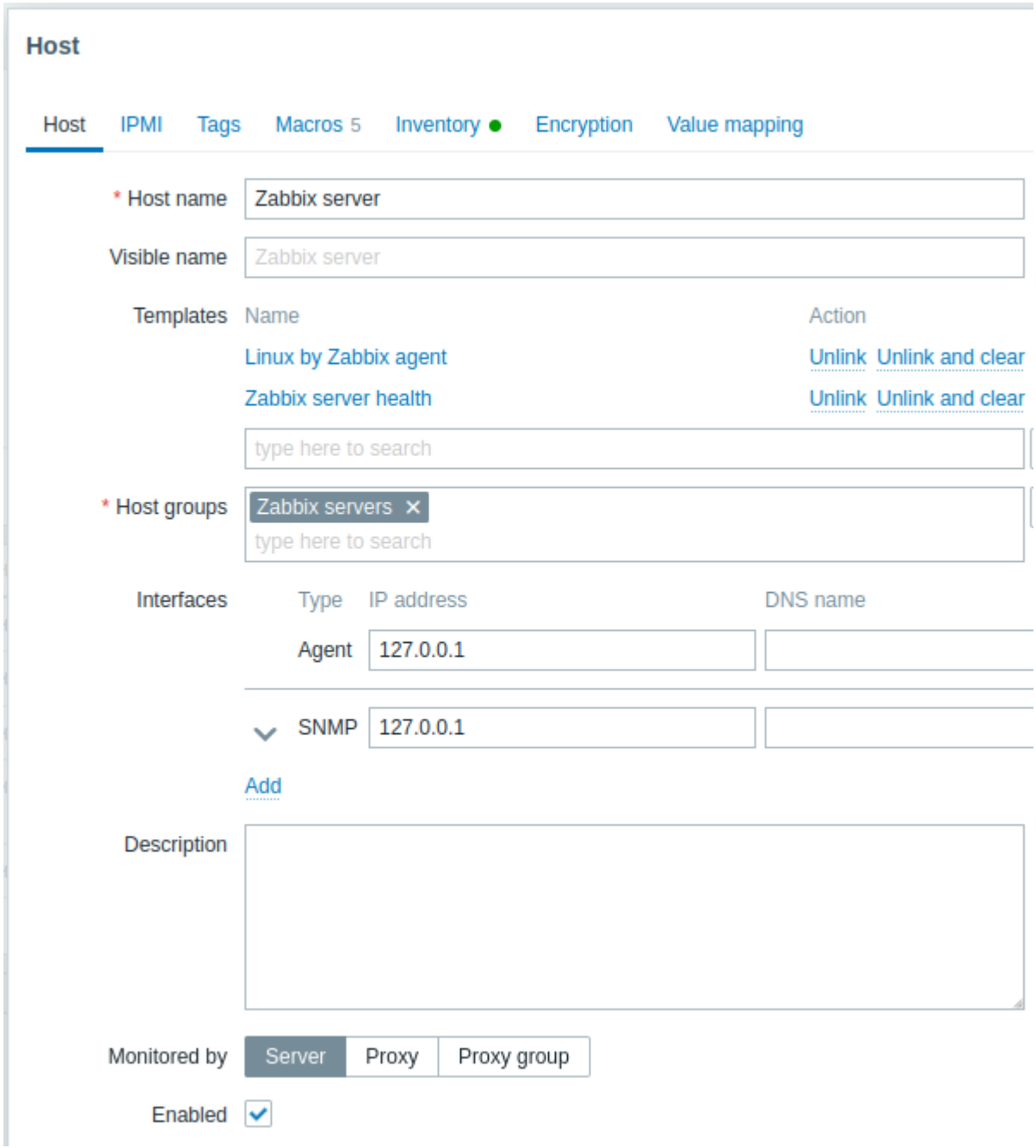
Vous pouvez également utiliser le bouton *Cloner* dans le formulaire de configuration d'un hôte existant pour créer un nouvel hôte. Cet hôte aura toutes les propriétés de l'hôte existant, y compris les modèles liés, les entités (éléments, déclencheurs, etc.) provenant de ces modèles, ainsi que les entités directement attachées à l'hôte existant.

Notez que lorsqu'un hôte est cloné, il conservera toutes les entités du modèle telles qu'elles sont initialement définies dans le modèle. Toute modification apportée à ces entités au niveau de l'hôte existant (comme un intervalle d'élément modifié, une expression régulière modifiée ou des prototypes ajoutés à la règle de découverte bas niveau) ne sera pas clonée vers le nouvel hôte ; elles resteront celles du modèle.

Vous pouvez également utiliser l'Assistant d'hôte pour configurer un hôte via une interface guidée, étape par étape.

Configuration

L'onglet **Hôte** contient les attributs généraux de l'hôte :



Tous les champs obligatoires sont marqués d'un astérisque rouge.

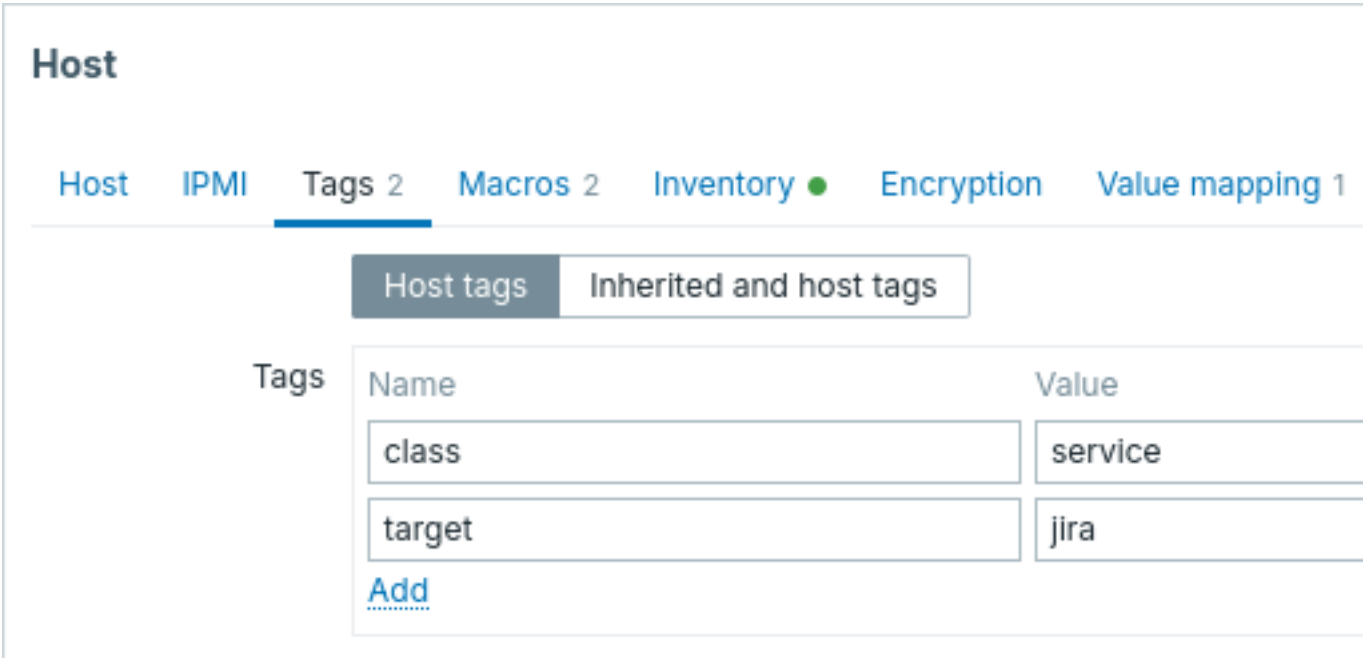
Paramètre	Description
Nom de l'hôte	Saisissez un nom d'hôte unique. Les caractères alphanumériques, les espaces, les points, les tirets et les traits de soulignement sont autorisés. Toutefois, les espaces en début et en fin sont interdits. *Remarque :* Si l'agent Zabbix s'exécute sur l'hôte que vous configurez, le paramètre *Hostname* du fichier de configuration de l'agent doit avoir la même valeur que le nom d'hôte saisi ici. Le nom défini dans ce paramètre est nécessaire au traitement des vérifications actives.
Nom visible	Saisissez un nom visible unique pour l'hôte. Si vous définissez ce nom, il sera affiché dans les listes, les cartes, etc., à la place du nom technique de l'hôte. Cet attribut prend en charge UTF-8.

Paramètre	Description
Modèles	Liez des modèles à l'hôte. Toutes les entités (éléments, déclencheurs, etc.) seront héritées du modèle. Pour lier un nouveau modèle, commencez à saisir son nom dans le champ de saisie. Une liste des modèles correspondants s'affiche ; faites-la défiler pour effectuer votre sélection. Vous pouvez également cliquer sur *Sélectionner* à côté du champ et sélectionner des modèles dans la liste d'une fenêtre contextuelle. Tous les modèles sélectionnés seront liés à l'hôte lors de l'enregistrement ou de la mise à jour du formulaire de configuration de l'hôte. Pour délier un modèle, utilisez l'une des deux options du bloc *Modèles liés* : *Délier* - délie le modèle, mais conserve ses entités (éléments, déclencheurs, etc.) ; *Délier et effacer* - délie le modèle et supprime toutes ses entités (éléments, déclencheurs, etc.). Les noms des modèles affichés sont des liens cliquables menant au formulaire de configuration du modèle.
Groupes d'hôtes	Sélectionnez les groupes d'hôtes auxquels appartient l'hôte. Un hôte doit appartenir à au moins un groupe d'hôtes. Vous pouvez créer un nouveau groupe et le lier à l'hôte en ajoutant le nom d'un groupe inexistant.
Interfaces	Plusieurs types d'interfaces d'hôte sont pris en charge : *Agent*, *SNMP*, *JMX* et *IPMI*. Aucune interface n'est définie par défaut. Pour ajouter une nouvelle interface, cliquez sur *Ajouter* dans le bloc *Interfaces*, sélectionnez le type d'interface et saisissez les informations *IP/DNS*, *Se connecter à* et *Port*. *Remarque :* Les interfaces utilisées dans des éléments ne peuvent pas être supprimées et le lien *Supprimer* est grisé pour ces interfaces. L'« IP » ou le « DNS » d'une interface SNMP est également utilisé pour les traps SNMP. Lors de la mise en correspondance, seule l'« IP » ou le « DNS » sélectionné dans l'interface de l'hôte est utilisé. Consultez Configuration de la supervision SNMP pour plus d'informations sur la configuration d'une interface SNMP (v1, v2 et v3).
Adresse IP	Adresse IP de l'hôte (facultatif).
Nom DNS	Nom DNS de l'hôte (facultatif).
Se connecter à	Cliquez sur le bouton correspondant pour indiquer au serveur Zabbix ce qu'il doit utiliser pour récupérer les données des agents : **IP** - Se connecter à l'adresse IP de l'hôte (recommandé) **DNS** - Se connecter au nom DNS de l'hôte
Port	Numéro de port TCP/UDP. Les valeurs par défaut sont les suivantes : 10050 pour l'agent Zabbix, 161 pour l'agent SNMP, 12345 pour JMX et 623 pour IPMI.
Par défaut	Cochez le bouton radio pour définir l'interface par défaut.
Description	Saisissez la description de l'hôte.
Supervisé par	Sélectionnez si l'hôte est supervisé par : **Serveur** - l'hôte est supervisé par le serveur Zabbix ; **Proxy** - l'hôte est supervisé par un proxy autonome ; **Groupe de proxies** - l'hôte est supervisé par un groupe de proxies.
Proxy	Le nom du proxy attribué est affiché (uniquement si le serveur Zabbix en a attribué un à partir du groupe de proxies sélectionné). Ce champ est affiché uniquement si l'hôte est supervisé par un groupe de proxies.
Activé	Lorsque la case est cochée, l'hôte est activé et prêt pour la supervision. Lorsque la case n'est pas cochée, l'hôte est désactivé et n'est pas supervisé : Pour les demandes de données passives initiées par le serveur Zabbix/proxy (par exemple, l'agent Zabbix, l'agent SNMP, les vérifications simples), la supervision est désactivée après la synchronisation de la configuration. Les déclencheurs et les actions liés à l'hôte sont également désactivés uniquement après le rechargement de la configuration. Pour les vérifications actives de l'agent Zabbix, la supervision s'arrête dans le délai (environ 5 secondes) nécessaire à l'agent Zabbix pour recevoir l'information indiquant que l'hôte a été désactivé. Pendant ce bref intervalle, l'hôte continue à collecter localement les données des vérifications actives et tente de les envoyer au serveur/proxy ; toutefois, puisque l'hôte est marqué comme *Désactivé*, le serveur/proxy rejette les données. Lorsque vous désactivez l'hôte, ses éléments sont immédiatement supprimés du cache de l'historique (à l'exception de leurs dernières valeurs, qui sont conservées pour les journaux).

L'onglet **IPMI** contient les attributs de gestion IPMI.

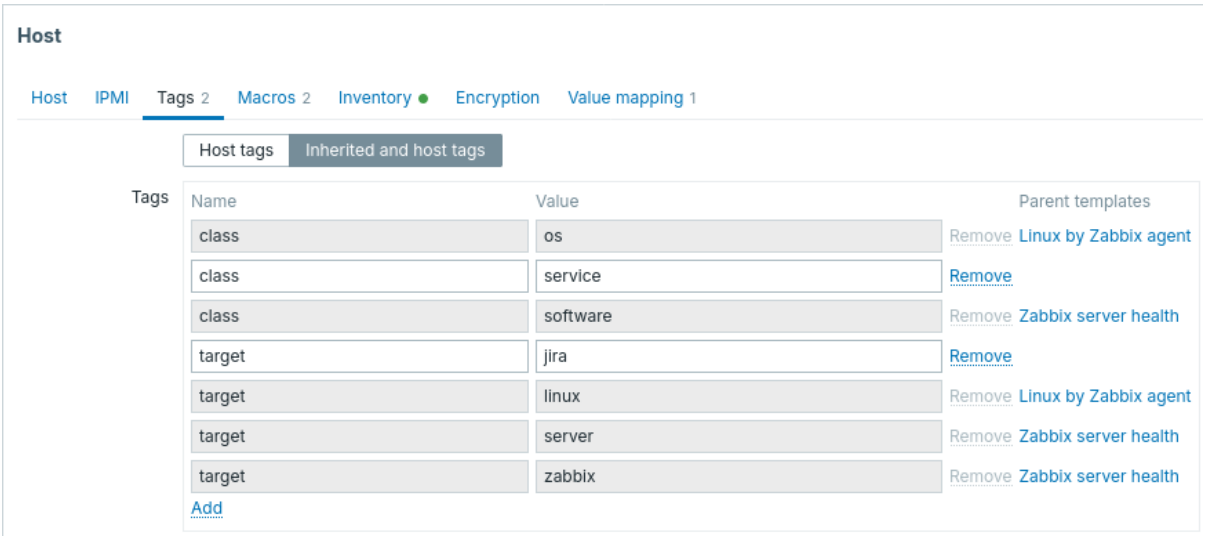
Paramètre	Description
Algorithme d'authentification	Sélectionnez l'algorithme d'authentification.
Niveau de privilège	Sélectionnez le niveau de privilège.
Nom d'utilisateur	Nom d'utilisateur utilisé pour l'authentification. Les macros utilisateur peuvent être utilisées.
Mot de passe	Mot de passe utilisé pour l'authentification. Les macros utilisateur peuvent être utilisées.

L'onglet **Étiquettes** vous permet de définir des étiquettes au niveau de l'hôte. Tous les problèmes de cet hôte seront associés aux valeurs saisies ici.



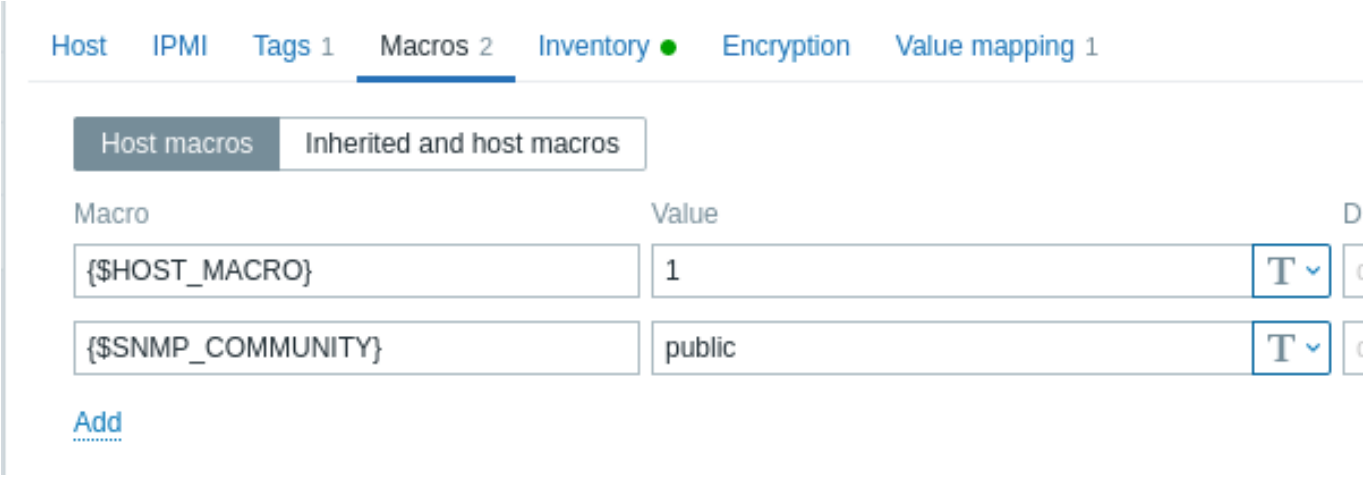
Les macros utilisateur, les macros `{INVENTORY.*}`, `{HOST.HOST}`, `{HOST.NAME}`, `{HOST.CONN}`, `{HOST.DNS}`, `{HOST.IP}`, `{HOST.PORT}` et `{HOST.ID}` sont prises en charge dans les étiquettes.

Vous pouvez également afficher ici les étiquettes au niveau des modèles si vous sélectionnez l'option *Étiquettes héritées et de l'hôte*. Toutes les étiquettes utilisateur définies pour l'hôte sont affichées, ainsi que leur origine.

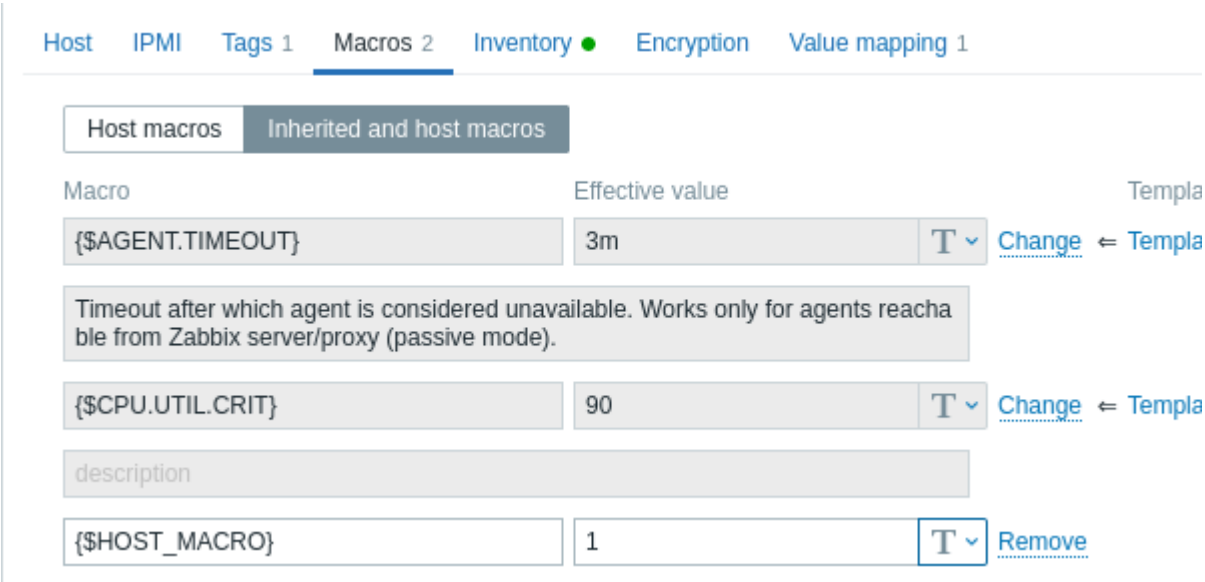


Pour plus de commodité, des liens vers les modèles correspondants sont fournis.

L'onglet **Macros** vous permet de définir des macros utilisateur au niveau de l'hôte sous forme de paires nom-valeur. Notez que les valeurs des macros peuvent être conservées sous forme de texte brut, de texte secret ou de secret Vault. Il est également possible d'ajouter une description.

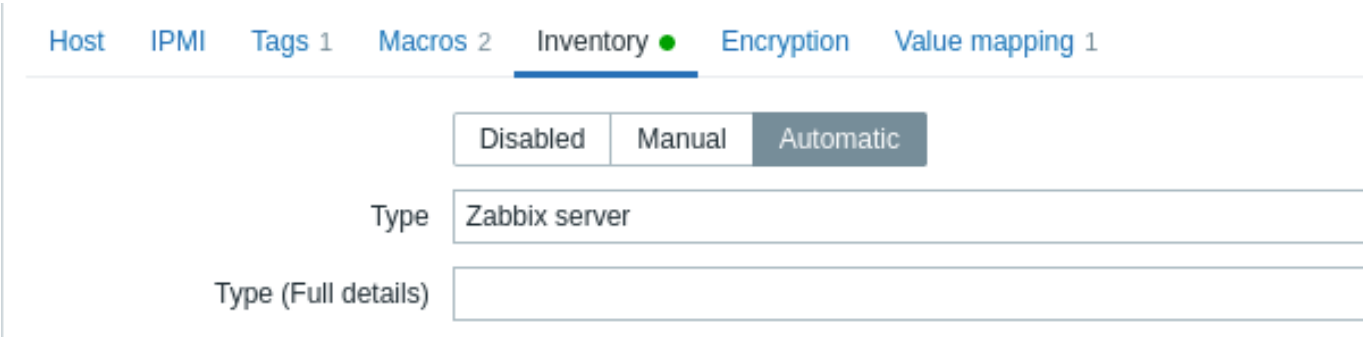


Vous pouvez également afficher ici les macros utilisateur au niveau des modèles et au niveau global si vous sélectionnez l'option *Macros héritées et de l'hôte*. Toutes les macros utilisateur définies pour l'hôte sont affichées, avec la valeur à laquelle elles sont résolues ainsi que leur origine.



Pour plus de commodité, des liens vers les modèles correspondants et la configuration des macros globales sont fournis. Il est également possible de modifier une macro de modèle ou une macro globale au niveau de l'hôte, ce qui crée effectivement une copie de la macro sur l'hôte.

L'onglet **Inventaire** vous permet de saisir manuellement les informations d'inventaire de l'hôte. Vous pouvez également choisir d'activer le remplissage *Automatique* de l'inventaire ou de désactiver le remplissage de l'inventaire pour cet hôte.



Si l'inventaire est activé (manuellement ou automatiquement), un point vert est affiché à côté du nom de l'onglet.

Chiffrement

L'onglet **Chiffrement** vous permet d'exiger des connexions chiffrées avec l'hôte.

Parameter	Description
Connections to host	Comment le serveur Zabbix ou le proxy se connecte à l'agent Zabbix sur un hôte: sans chiffrement (par défaut), en utilisant PSK (clé prépartagée) ou un certificat.
Connections from host	Sélectionnez le type de connexions autorisées depuis l'hôte (c.-à-d. depuis l'agent Zabbix et Zabbix sender). Plusieurs types de connexion peuvent être sélectionnés en même temps (utile pour les tests et pour basculer vers un autre type de connexion). La valeur par défaut est "No encryption".
Issuer	Émetteur autorisé du certificat. Le certificat est d'abord validé avec l'AC (autorité de certification). S'il est valide et signé par l'AC, le champ *Issuer* peut être utilisé pour restreindre davantage l'AC autorisée. Ce champ est destiné à être utilisé si votre installation Zabbix utilise des certificats provenant de plusieurs AC. Si ce champ est vide, toute AC est acceptée.
Subject	Sujet autorisé du certificat. Le certificat est d'abord validé avec l'AC. S'il est valide et signé par l'AC, le champ *Subject* peut être utilisé pour n'autoriser qu'une seule valeur de chaîne *Subject*. Si ce champ est vide, tout certificat valide signé par l'AC configurée est accepté.
PSK identity	Chaîne d'identité de la clé prépartagée. Ne mettez pas d'informations sensibles dans l'identité PSK, elle est transmise non chiffrée sur le réseau afin d'indiquer au destinataire quelle PSK utiliser.
PSK	Clé prépartagée (chaîne hexadécimale). Longueur maximale: 512 chiffres hexadécimaux (PSK de 256 octets) si Zabbix utilise la bibliothèque GnuTLS ou OpenSSL, 64 chiffres hexadécimaux (PSK de 32 octets) si Zabbix utilise la bibliothèque mbed TLS (PolarSSL). Exemple: 1f87b595725ac58dd977beef14b97461a7c1045b9a1c963065002c5473194952

Mappage des valeurs

L'onglet **Mappage des valeurs** vous permet de configurer une représentation conviviale des données des éléments dans les mappages de valeurs.

Configuration d'un groupe d'hôtes

Vue d'ensemble

Les groupes d'hôtes sont utilisés pour le regroupement logique des hôtes et l'attribution des permissions utilisateur sur ceux-ci.

Chaque hôte doit avoir au moins un groupe d'hôtes assigné. Un hôte peut appartenir à plusieurs groupes d'hôtes, et chaque groupe d'hôtes peut contenir plusieurs hôtes.

Notez que dans Zabbix, toutes les permissions sont basées sur des groupes : groupes d'utilisateurs, groupes d'hôtes et groupes de modèles. Ainsi, même si un seul utilisateur a besoin d'accéder à un seul hôte, cet accès est accordé en ajoutant l'utilisateur à un groupe d'utilisateurs et en attribuant à ce groupe d'utilisateurs des permissions sur le groupe d'hôtes contenant cet hôte.

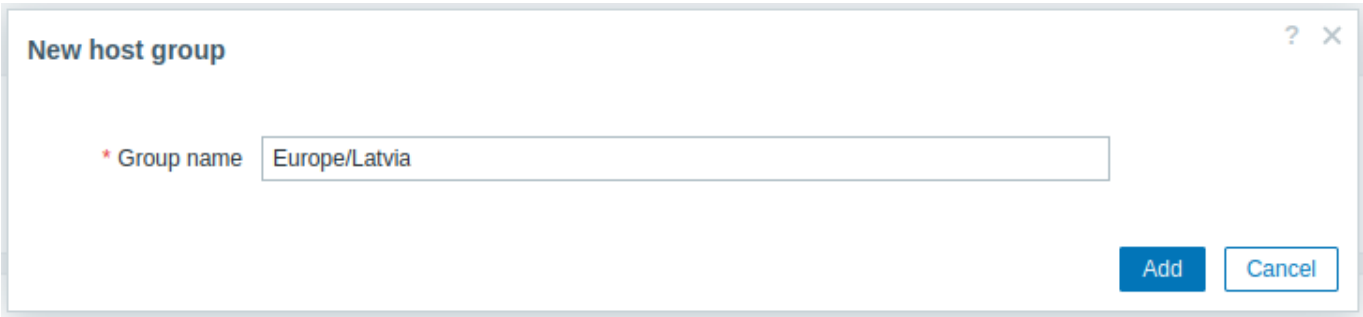
Configuration

> **Attention:**
> Seuls les utilisateurs Super admin peuvent créer des groupes d'hôtes.

Il existe deux options pour créer un groupe d'hôtes dans l'interface Zabbix.

Première option :

1. Accédez à *Collecte de données* > *Groupes d'hôtes*.
2. Cliquez sur *Créer un groupe d'hôtes* dans le coin supérieur droit de l'écran.
3. Saisissez le nom du groupe dans le formulaire.



Deuxième option : lors de la configuration d'un hôte, saisissez un nom de groupe inexistant dans le champ de saisie *Groupes d'hôtes*.

Une fois le groupe d'hôtes créé, vous pouvez cliquer sur le nom du groupe dans la liste sous *Collecte de données* > *Groupes d'hôtes* pour modifier le nom du groupe, cloner le groupe ou supprimer le groupe.

La suppression d'un groupe d'hôtes supprime uniquement le groupe logique, et non les hôtes du groupe. Il n'est pas possible de supprimer un groupe d'hôtes qui est le seul groupe d'un hôte existant.

Création de sous-groupes d'hôtes

Un sous-groupe d'hôtes (ou groupe d'hôtes imbriqué) est un enfant du groupe d'hôtes parent qui le contient.

Un sous-groupe est créé en utilisant la barre oblique « / » dans le champ de saisie du nom du groupe afin d'indiquer sa relation avec le ou les groupes parents. Par exemple :

- saisir `Europe/Latvia` crée le sous-groupe `Europe/Latvia` du groupe parent `Europe`.
- saisir `Europe/Latvia/Riga/Zabbix servers` crée le sous-groupe correspondant dans les groupes parents imbriqués `Europe`, `Europe/Latvia`, `Europe/Latvia/Riga`.

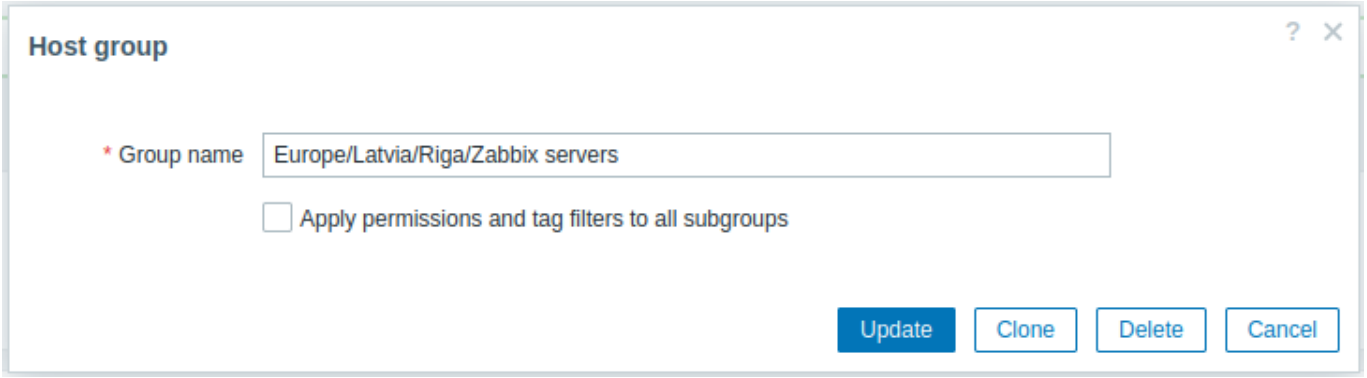
Lors de la création d'un sous-groupe, l'utilisation de barres obliques au début ou à la fin, ou de plusieurs barres obliques consécutives, n'est pas autorisée. L'échappement de « / » n'est pas pris en charge.

Il n'est pas nécessaire de créer un ou plusieurs groupes d'hôtes parents avant de créer un sous-groupe. Vous pouvez choisir de commencer par créer un sous-groupe (par exemple, `Europe/Latvia`) ou un ou plusieurs groupes d'hôtes parents (dans notre exemple, `Europe`). Si vous commencez par créer un sous-groupe, le ou les groupes d'hôtes parents **ne** seront **pas** créés automatiquement.

Autorisations pour les groupes d'hôtes

- Lors de la création d'un sous-groupe d'un groupe d'hôtes parent existant (par exemple, la création de `Europe/Latvia` alors que `Europe` existe déjà), les autorisations du groupe d'utilisateurs pour le sous-groupe sont héritées du parent.
- Lors de la création d'un groupe d'hôtes parent pour un sous-groupe existant (par exemple, la création de `Europe` alors que `Europe/Latvia` existe déjà), aucune autorisation n'est définie pour le parent.

Lors de la modification de n'importe quel groupe d'hôtes, vous pouvez également définir une option supplémentaire, *Appliquer les autorisations et les filtres de balises à tous les sous-groupes*.



En cochant cette case et en cliquant sur *Mettre à jour*, le même niveau d'autorisations et de filtres de balises sera appliqué à tous les sous-groupes actuels et futurs du groupe d'hôtes en cours de modification.

Ainsi, si des groupes d'utilisateurs se sont vu attribuer des autorisations différentes pour les sous-groupes du groupe d'hôtes en cours de modification, le fait de cocher cette case accordera à tous les sous-groupes actuels et futurs les mêmes autorisations d'utilisateur et autorisations basées sur des balises que celles du groupe en cours de modification.

Notez que cette option n'est pas enregistrée dans la base de données et remplacera les autorisations existantes. Toute modification effectuée via cette option ne peut être annulée que manuellement.

Inventaire

Aperçu

Vous pouvez conserver l'inventaire des périphériques réseau dans Zabbix.

Il y a un menu spécial *Inventaire* dans l'interface Zabbix. Cependant, vous ne verrez aucune donnée initialement et ce n'est pas là que vous entrez les données. Construire les données d'inventaire est fait manuellement lors de la configuration d'un hôte ou automatiquement en utilisant certaines options de peuplement automatique.

Construction de l'inventaire

Mode manuel

Lors de la configuration d'un hôte, dans l'onglet *Inventaire*, vous pouvez saisir des informations telles que le type de périphérique, le numéro de série, l'emplacement, la personne responsable, les URL, etc. - ces données alimenteront les informations d'inventaire.

Si une URL est incluse dans les informations d'inventaire de l'hôte et qu'elle commence par « http » ou « https », elle apparaîtra sous forme de lien cliquable dans la section *Inventaire*.

Mode automatique

L'inventaire des hôtes peut également être renseigné automatiquement. Pour cela, lors de la configuration d'un hôte, le mode d'inventaire dans l'onglet *Inventaire* doit être défini sur *Automatique*.

Vous pouvez ensuite configurer des éléments d'hôte afin de renseigner n'importe quel champ d'inventaire de l'hôte avec leur valeur, en indiquant le champ de destination à l'aide de l'attribut correspondant (appelé *L'élément renseignera le champ d'inventaire de l'hôte*) dans la configuration de l'élément.

Éléments particulièrement utiles pour la collecte automatisée des données d'inventaire :

- system.hw.chassis[full|type|vendor|model|serial] - la valeur par défaut est [full], les permissions root sont nécessaires
- system.hw.cpu[all|cpunum,full|maxfreq|vendor|model|curfreq] - la valeur par défaut est [all,full]
- system.hw.devices[pci|usb] - la valeur par défaut est [pci]
- system.hw.macaddr[interface,short|full] - la valeur par défaut est [all,full], interface est une expression régulière
- system.sw.arch
- system.sw.os[name|short|full] - la valeur par défaut est [name]
- system.sw.packages[regexp,manager,short|full] - la valeur par défaut est [all,all,full].

Sélection du mode d'inventaire

Le mode d'inventaire peut être sélectionné dans le formulaire de configuration de l'hôte.

Par défaut, le mode d'inventaire des nouveaux hôtes est sélectionné en fonction du paramètre *Mode d'inventaire par défaut de l'hôte* dans *Administration* > *Général* > *Autre*.

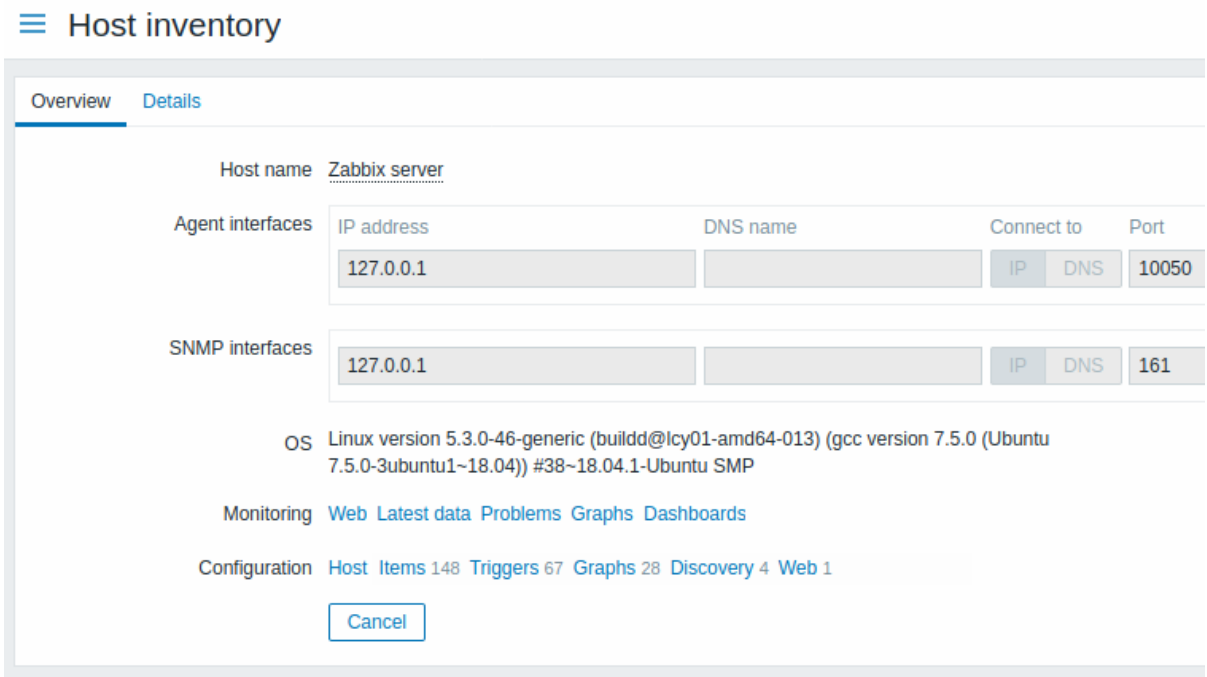
Pour les hôtes ajoutés par la découverte réseau ou par des actions d'enregistrement automatique, il est possible de définir une opération *Définir le mode d'inventaire de l'hôte* en sélectionnant le mode manuel ou automatique. Cette opération remplace le paramètre *Mode d'inventaire par défaut de l'hôte*.

Vue d'ensemble de l'inventaire

Les détails de toutes les données d'inventaire existantes sont disponibles dans le menu *Inventaire*.

Dans *Inventaire* > *Vue d'ensemble*, vous pouvez obtenir un nombre d'hôtes selon différents champs de l'inventaire.

Dans *Inventaire* > *Hôtes*, vous pouvez voir tous les hôtes pour lesquels des informations d'inventaire sont disponibles. En cliquant sur le nom de l'hôte, vous afficherez les détails de l'inventaire dans un formulaire.



L'onglet **Vue d'ensemble** affiche :

Parameter	Description
Host name	Nom de l'hôte. Cliquer sur le nom ouvre un menu avec les scripts définis pour l'hôte. Le nom de l'hôte est affiché avec une icône orange si l'hôte est en maintenance.
Visible name	Nom visible de l'hôte (s'il est défini).
Host (Agent, SNMP, JMX, IPMI) interfaces	Ce bloc fournit des détails sur les interfaces configurées pour l'hôte.
OS	Champ d'inventaire du système d'exploitation de l'hôte (s'il est défini).
Hardware	Champ d'inventaire du matériel de l'hôte (s'il est défini).
Software	Champ d'inventaire des logiciels de l'hôte (s'il est défini).
Description	Description de l'hôte.
Monitoring	Liens vers les sections de supervision contenant des données pour cet hôte : *Web*, *Dernières données*, *Problèmes*, *Graphiques*, *Tableaux de bord*.
Configuration	Liens vers les sections de configuration pour cet hôte : *Hôte*, *Éléments*, *Déclencheurs*, *Graphiques*, *Découverte*, *Web*. Le nombre d'entités configurées est indiqué après chaque lien.

L'onglet **Détails** affiche tous les champs d'inventaire qui sont renseignés (non vides).

Macros d'inventaire

Des macros d'inventaire d'hôte `{INVENTORY.*}` sont disponibles pour être utilisées dans les notifications, par exemple :

"Le serveur situé à {INVENTORY.LOCATION1} a un problème, la personne responsable est {INVENTORY.CONTACT1}, le numéro de téléphone est {INVENTORY.POC.PRIMARY.PHONE.A1}."

Pour plus de détails, consultez la page Macros prises en charge.

Mise à jour en masse

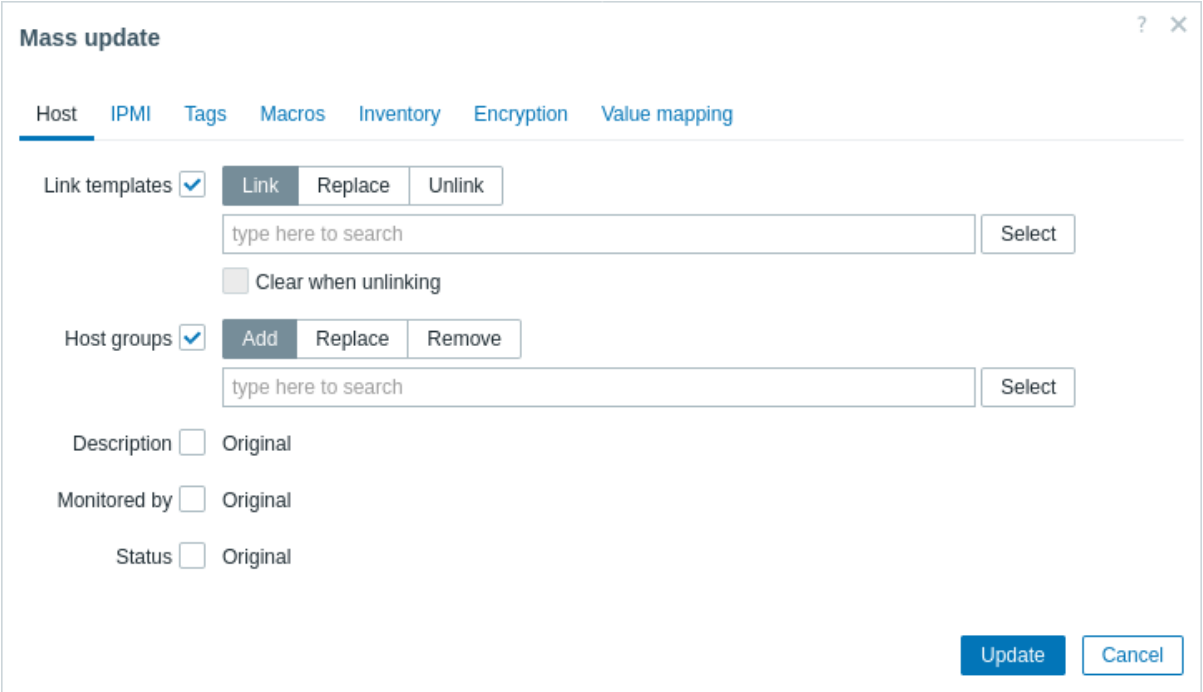
Aperçu

Parfois, vous voudrez peut-être changer certains attributs pour un certain nombre d'hôtes en une seule fois. Au lieu d'ouvrir chaque hôte individuellement pour l'édition, vous pouvez utiliser la fonction de modification collective pour cela.

Utilisation de la mise à jour en masse

Pour mettre à jour plusieurs hôtes en masse, procédez comme suit :

- Cochez les cases devant les hôtes que vous souhaitez mettre à jour dans la liste des hôtes
- Cliquez sur *Mise à jour en masse* sous la liste
- Accédez à l'onglet contenant les attributs requis (*Hôte*, *IPMI*, *Étiquettes*, *Macros*, *Inventaire*, *Chiffrement* ou *Mappage de valeurs*)
- Cochez les cases des attributs à mettre à jour et saisissez une nouvelle valeur pour ceux-ci



Les options suivantes sont disponibles lors de la sélection du bouton correspondant pour la mise à jour des liens de **modèle** :

- *Lier* - indiquez les modèles supplémentaires à lier
- *Remplacer* - indiquez les modèles à lier tout en dissociant tout modèle précédemment lié aux hôtes
- *Dissocier* - indiquez les modèles à dissocier

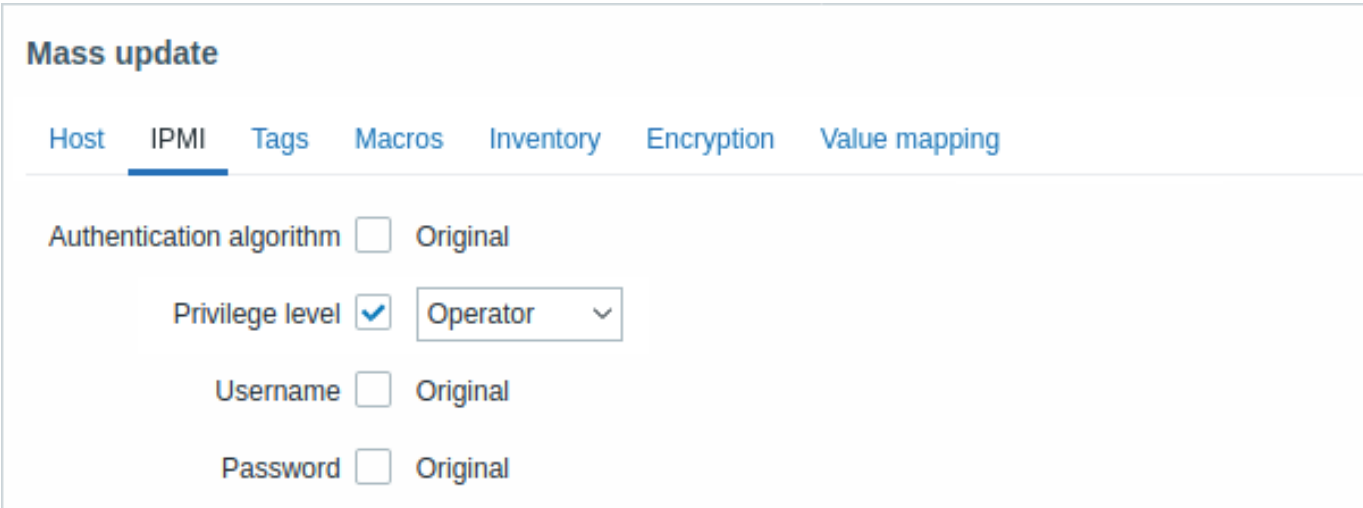
Pour indiquer les modèles à lier ou à dissocier, commencez à saisir le nom du modèle dans le champ de saisie automatique jusqu'à ce qu'une liste déroulante apparaisse avec les modèles correspondants. Faites défiler la liste pour sélectionner le modèle requis.

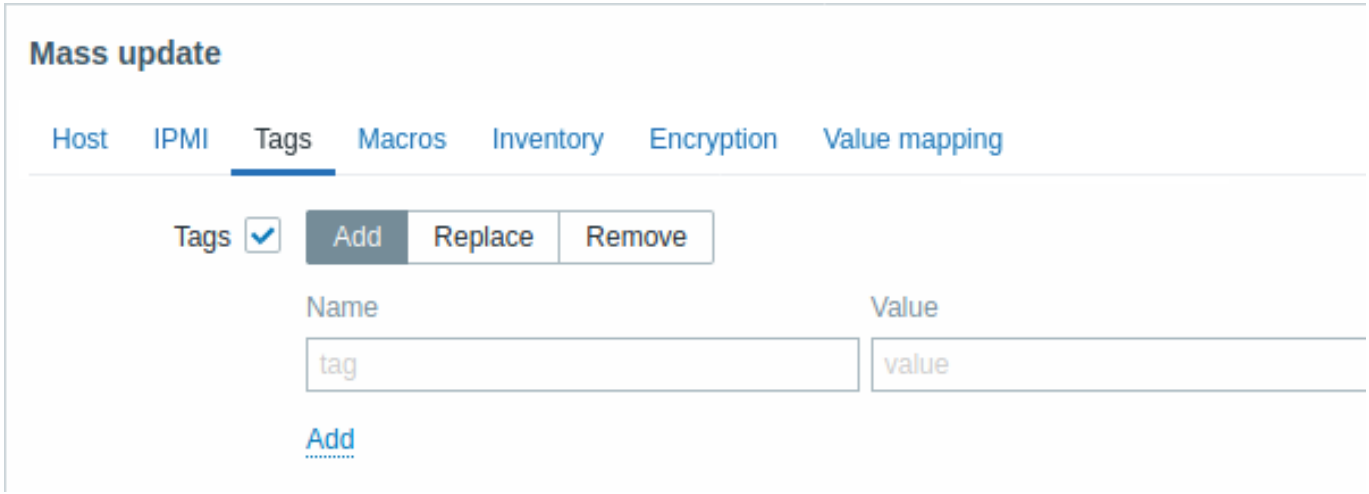
L'option *Effacer lors de la dissociation* permet non seulement de dissocier les modèles précédemment liés, mais également de supprimer toutes les entités qui en héritent (éléments, déclencheurs, graphiques, etc.).

Les options suivantes sont disponibles lors de la sélection du bouton correspondant pour la mise à jour des **groupes d'hôtes** :

- *Ajouter* - indiquez des groupes d'hôtes supplémentaires parmi ceux existants ou saisissez des groupes d'hôtes entièrement nouveaux pour les hôtes ;
- *Remplacer* - supprimez l'hôte de tous les groupes d'hôtes existants et remplacez-les par le ou les groupes indiqués dans ce champ (groupes d'hôtes existants ou nouveaux) ;
- *Supprimer* - supprimez des groupes d'hôtes spécifiques des hôtes.

Ces champs proposent une saisie automatique : commencez à y saisir du texte pour afficher une liste déroulante de groupes d'hôtes correspondants. Si le groupe d'hôtes est nouveau, il apparaît également dans la liste déroulante et est indiqué par *(nouveau)* après le texte. Faites défiler la liste pour le sélectionner.

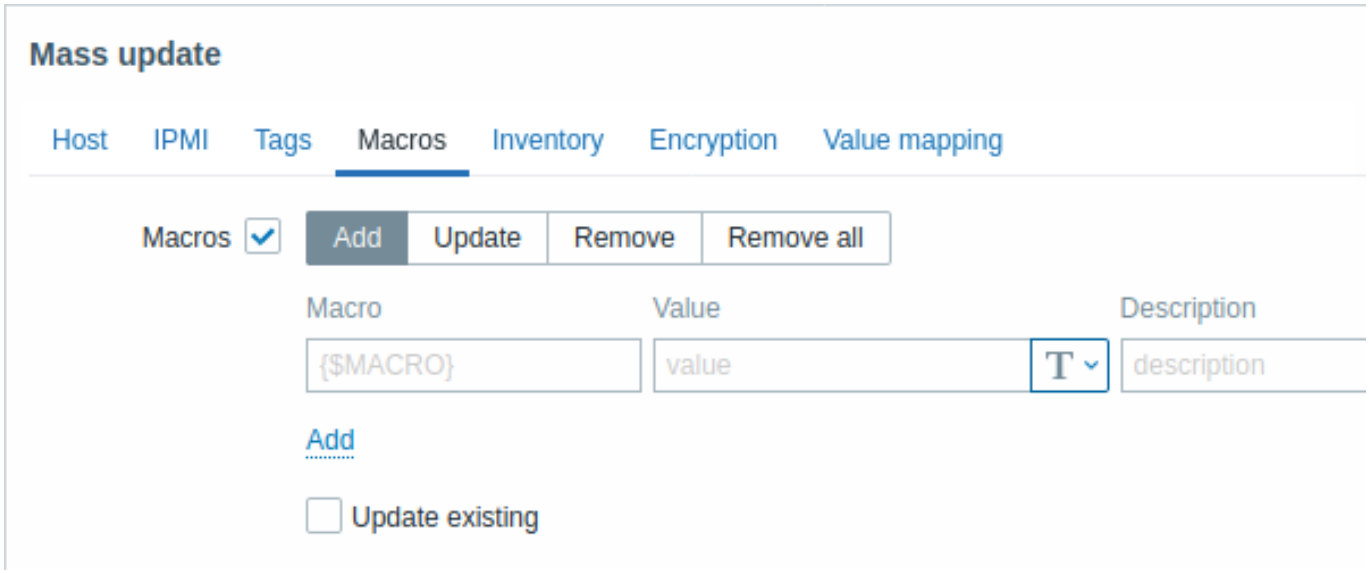




Les options suivantes sont disponibles lors de la sélection du bouton correspondant pour la mise à jour des étiquettes :

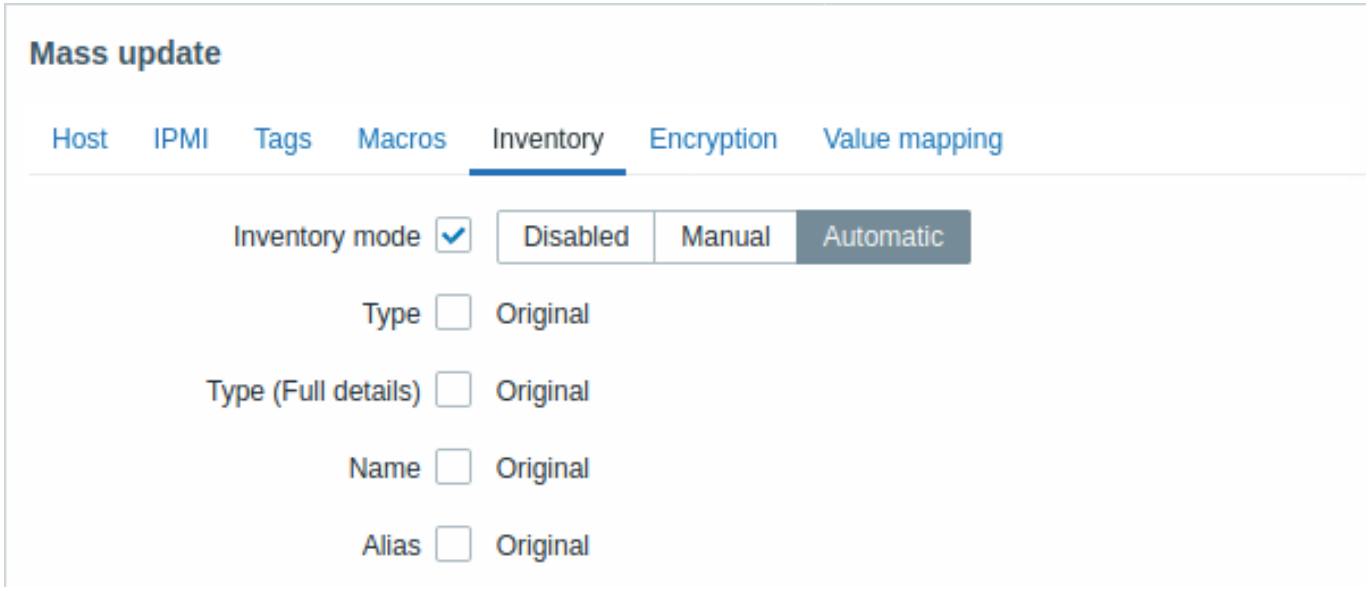
- *Ajouter* - ajoutez de nouvelles étiquettes aux hôtes ;
- *Remplacer* - supprimez toutes les étiquettes existantes de l'hôte et remplacez-les par celle(s) indiquée(s) ci-dessous ;
- *Supprimer* - supprimez les étiquettes indiquées des hôtes.

Les macros utilisateur, les macros `{INVENTORY.*}`, `{HOST.HOST}`, `{HOST.NAME}`, `{HOST.CONN}`, `{HOST.DNS}`, `{HOST.IP}`, `{HOST.PORT}` et `{HOST.ID}` sont prises en charge dans les étiquettes. Notez que les étiquettes portant le même nom mais ayant des valeurs différentes ne sont pas considérées comme des « doublons » et peuvent être ajoutées au même hôte.

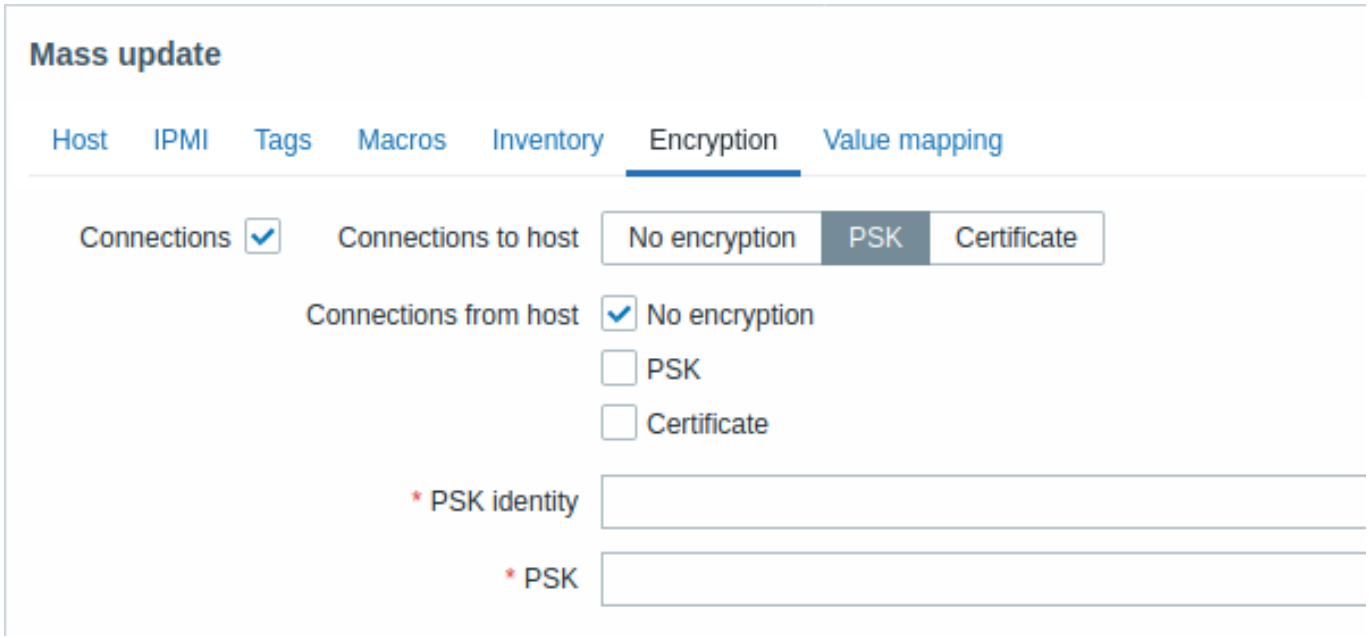


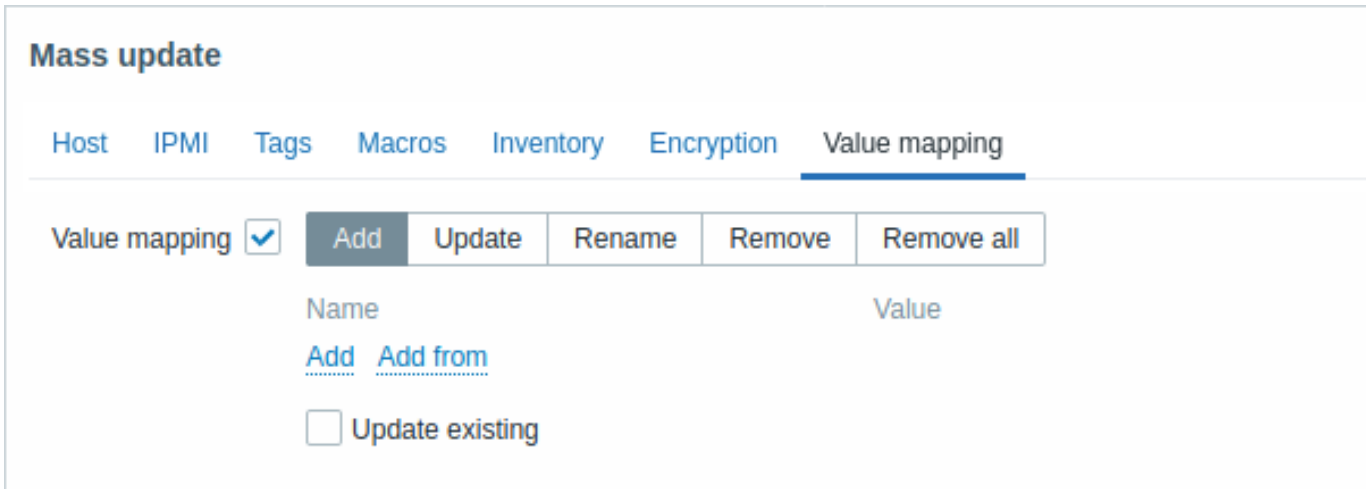
Les options suivantes sont disponibles lors de la sélection du bouton correspondant pour la mise à jour des macros :

- *Ajouter* - indiquez des macros utilisateur supplémentaires pour les hôtes. Si la case *Mettre à jour les éléments existants* est cochée, la valeur, le type et la description correspondant au nom de macro indiqué seront mis à jour. Si elle n'est pas cochée et qu'une macro portant ce nom existe déjà sur le ou les hôtes, elle ne sera pas mise à jour.
- *Mettre à jour* - remplacez les valeurs, les types et les descriptions des macros indiquées dans cette liste. Si la case *Ajouter les éléments manquants* est cochée, toute macro qui n'existait pas auparavant sur un hôte sera ajoutée comme nouvelle macro. Si elle n'est pas cochée, seules les macros qui existent déjà sur un hôte seront mises à jour.
- *Supprimer* - supprimez les macros indiquées des hôtes. Si la case *Sauf celles sélectionnées* est cochée, toutes les macros sauf celles indiquées dans la liste seront supprimées. Si elle n'est pas cochée, seules les macros indiquées dans la liste seront supprimées.
- *Tout supprimer* - supprimez toutes les macros utilisateur des hôtes. Si la case *Je confirme la suppression de toutes les macros* n'est pas cochée, une nouvelle fenêtre contextuelle s'ouvrira pour demander la confirmation de la suppression de toutes les macros.



Pour pouvoir mettre à jour les champs d'inventaire en masse, le *Mode d'inventaire* doit être défini sur « Manuel » ou « Automatique ».





Les boutons proposant les options suivantes sont disponibles pour la mise à jour des mappages de valeurs :

- *Ajouter* - ajoutez des mappages de valeurs aux hôtes. Si vous cochez *Mettre à jour les éléments existants*, toutes les propriétés du mappage de valeurs portant ce nom seront mises à jour. Sinon, si un mappage de valeurs portant ce nom existe déjà, il ne sera pas mis à jour.
- *Mettre à jour* - mettez à jour les mappages de valeurs existants. Si vous cochez *Ajouter les éléments manquants*, un

mappage de valeurs qui n'existait pas auparavant sur un hôte sera ajouté comme nouveau mappage de valeurs. Sinon, seuls les mappages de valeurs qui existent déjà sur un hôte seront mis à jour.

- *Renommer* - donnez un nouveau nom à un mappage de valeurs existant
- *Supprimer* - supprimez les mappages de valeurs indiqués des hôtes. Si vous cochez *Sauf celles sélectionnées*, tous les mappages de valeurs seront supprimés **à l'exception** de ceux qui sont indiqués.
- *Tout supprimer* - supprimez tous les mappages de valeurs des hôtes. Si la case *Je confirme la suppression de tous les mappages de valeurs* n'est pas cochée, une nouvelle fenêtre contextuelle s'ouvrira pour demander la confirmation de la suppression.

Une fois toutes les modifications requises effectuées, cliquez sur *Mettre à jour*. Les attributs seront mis à jour en conséquence pour tous les hôtes sélectionnés.

Éléments

Aperçu

Un élément est une métrique individuelle.

Les éléments sont utilisés pour collecter des données. Une fois qu'un hôte est configuré, vous devez ajouter des éléments pour obtenir des données réelles. Un moyen d'ajouter rapidement de nombreux éléments consiste à lier à un hôte l'un des modèles prédéfinis. Cependant, pour optimiser les performances du système, il peut être nécessaire d'ajuster finement les modèles afin d'avoir autant d'éléments et une fréquence de surveillance adaptée aux besoins.

Pour indiquer quel type de données collecter à partir d'un hôte, utilisez la clé d'élément. Par exemple, un élément avec le nom de clé **system.cpu.load** collectera des données sur la charge du processeur, tandis qu'un élément avec le nom de clé **net.if.in** collectera des informations sur le trafic entrant.

Des paramètres supplémentaires peuvent être spécifiés entre crochets après le nom de clé. Par exemple, system.cpu.load**[avg5]** renverra la charge moyenne du processeur sur les 5 dernières minutes, tandis que net.if.in**[eth0]** affichera le trafic entrant sur l'interface « eth0 ».

> **Note:**
> Consultez les sections individuelles des types d'élément pour connaître tous les types d'élément et toutes les clés d'élément pris en charge.

Passez à la création et la configuration d'un élément.

Création d'un élément

Vue d'ensemble

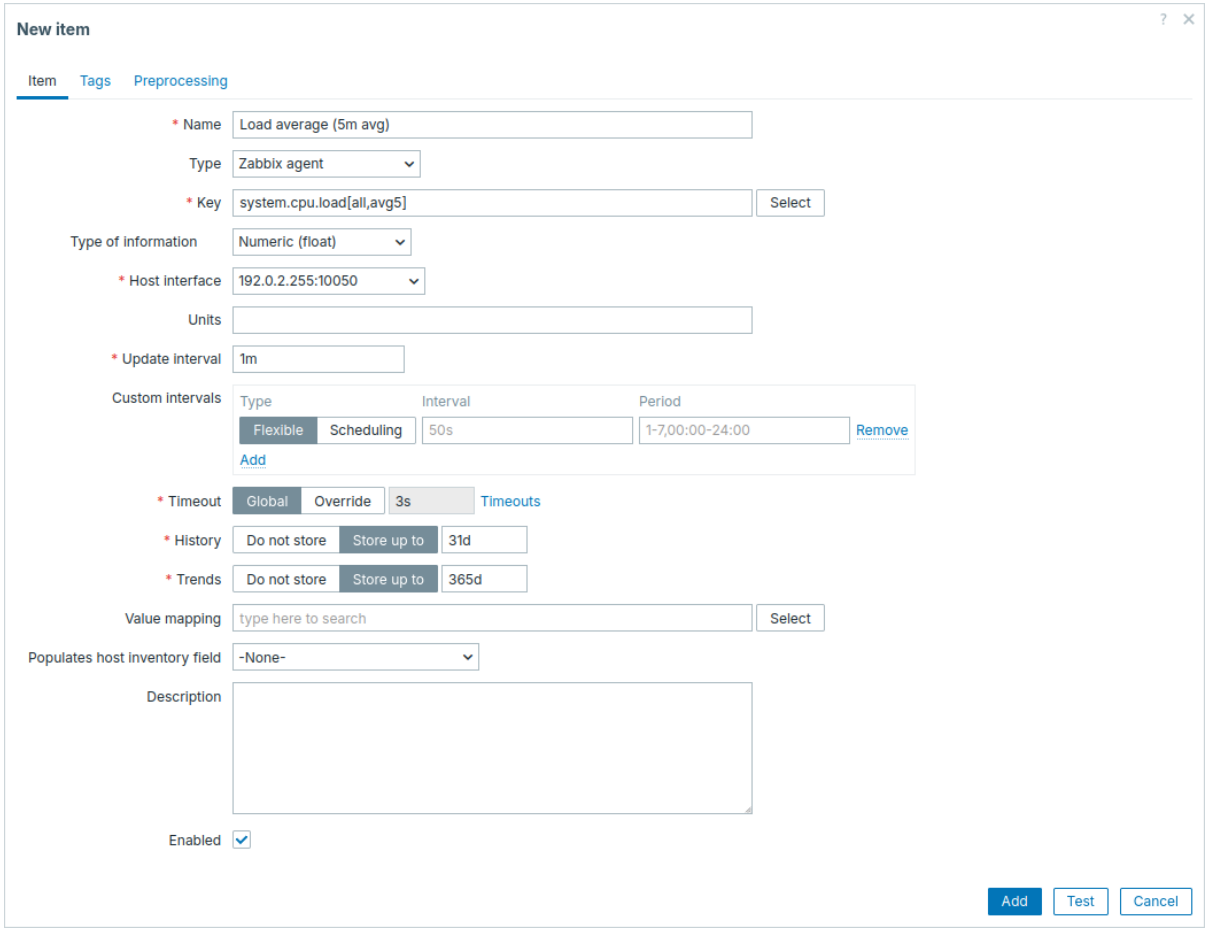
Pour créer un élément dans l'interface Zabbix, procédez comme suit :

1. Accédez à *Collecte de données* > *Hôtes*.
2. Cliquez sur *Éléments* dans la ligne du hôte.
3. Cliquez sur *Créer un élément* dans le coin supérieur droit de l'écran.
4. Saisissez les paramètres de l'élément dans le formulaire.

Vous pouvez également créer un élément en ouvrant un élément existant, en cliquant sur le bouton *Cloner*, puis en l'enregistrant sous un autre nom.

Configuration

L'onglet **Élément** contient les attributs généraux de l'élément.



Tous les champs obligatoires sont signalés par un astérisque rouge.

Paramètre	Description
Nom	Nom de l'élément. Les macros utilisateur sont prises en charge.
Type	Type d'élément. Consultez les sections consacrées à chaque type d'élément.
Clé	Clé de l'élément (jusqu'à 2048 caractères). Les clés d'élément prises en charge sont indiquées dans les sections consacrées à chaque type d'élément. La clé doit être unique au sein d'un même hôte. Si le type de clé est « Zabbix agent », « Zabbix agent (active) » ou « Simple check », la valeur de la clé doit être prise en charge par Zabbix agent ou le serveur Zabbix. Voir également le format de clé correct.
Type d'information	Type de données utilisé pour valider la valeur de l'élément et la stocker dans la base de données, après toute conversion : **Numérique (non signé)** - entier non signé de 64 bits ; **Numérique (flottant)** - nombre à virgule flottante de 64 bits ; **Caractère** - texte court ; **Journal** - texte long avec ou sans propriétés de journal (horodatage, source, gravité, ID de l'événement du journal) ; **Texte** - texte long ; **Binaire** - nombre binaire (pris en charge uniquement pour les éléments dépendants) ; **JSON** - données JSON structurées, stockées nativement dans la base de données (non pris en charge pour les éléments calculés). Les éléments renvoyant des valeurs de type binaire ou JSON ne sont pas pris en charge dans les formules d'éléments calculés ni dans les expressions de déclencheurs. Consultez les limites des données des éléments pour connaître les limites de stockage par type de données et moteur de base de données. Pour les éléments qui renvoient des données dans un format spécifique unique, un type de données correspondant est sélectionné automatiquement. Par exemple, `system.cpu.load` et `system.cpu.util` renvoient tous deux un nombre à virgule flottante de 64 bits ; *Numérique (flottant)* est donc sélectionné automatiquement.

Paramètre	Description
Interface de l'hôte	Sélectionnez l'interface de l'hôte. Ce champ est disponible lors de la modification d'un élément au niveau de l'hôte.
Unités	Si un symbole d'unité est défini, Zabbix applique un post-traitement à la valeur reçue de l'élément et l'affiche avec le suffixe d'unité indiqué. Symboles d'unité pris en charge avec un formatage spécial (et exemples de valeur reçue → valeur affichée) : **B** - octets (1024 → 1 KB) **Bps** - octets par seconde (1024 → 1 KBps) **s** - secondes, affichées avec jusqu'à trois unités de temps non nulles les plus grandes (881764 → 10d 4h 56m) **uptime** - durée écoulée au format hh:mm:ss ou N jours, hh:mm:ss (881764 → 10 days, 04:56:04) **unixtime** - horodatage Unix, formaté comme yyyy.mm.dd hh:mm:ss (881764 → 1970-01-11 04:56:04 AM) ; pour un formatage correct, la valeur reçue de l'élément doit être de type *Numérique (non signé)*. Pour les autres unités (telles que Hz, W, etc.), si la valeur reçue dépasse 1000, elle est divisée par 1000 et affichée avec le préfixe correspondant (5000 → 5 KHz, 881764 → 881.76 KW). Si le symbole d'unité est précédé de **!**, la conversion d'unité et l'ajout de préfixe sont désactivés (1024 !B → 1024 B, 61 !s → 61 s). Pour plus d'exemples et de détails sur les symboles d'unité et la conversion d'unité, consultez Suffixes des valeurs des éléments.
Intervalle de mise à jour	Récupérer une nouvelle valeur pour cet élément toutes les N secondes. L'intervalle de mise à jour maximal autorisé est de 86400 secondes (1 jour). Les suffixes de temps sont pris en charge, par exemple 30s, 1m, 2h, 1d. Les macros utilisateur sont prises en charge. Une seule macro doit remplir l'intégralité du champ. Plusieurs macros dans un champ ou des macros mélangées à du texte ne sont pas pris en charge. *Remarque* : l'intervalle de mise à jour ne peut être défini sur « 0 » que si des intervalles personnalisés existent avec une valeur différente de zéro. S'il est défini sur « 0 » et qu'un intervalle personnalisé (flexible ou planifié) existe avec une valeur différente de zéro, l'élément sera interrogé pendant la durée de l'intervalle personnalisé. *Notez* que la première interrogation de l'élément après son activation ou après la modification de l'intervalle de mise à jour peut avoir lieu plus tôt que la valeur configurée. Les nouveaux éléments seront vérifiés dans les 60 secondes suivant leur création, sauf s'ils disposent d'un intervalle de mise à jour planifié ou flexible et que l'*Intervalle de mise à jour* est défini sur 0. Un élément passif existant peut être interrogé immédiatement en cliquant sur le bouton *Exécuter maintenant* button.
Intervalles personnalisés	Vous pouvez créer des règles personnalisées pour vérifier l'élément : **Flexible** - crée une exception à l'*Intervalle de mise à jour* (intervalle avec une fréquence différente). **Planification** - crée un calendrier d'interrogation personnalisé. Pour plus d'informations, consultez Intervalles personnalisés. Les suffixes de temps sont pris en charge dans le champ *Intervalle*, par exemple 30s, 1m, 2h, 1d. Les macros utilisateur sont prises en charge. Une seule macro doit remplir l'intégralité du champ. Plusieurs macros dans un champ ou des macros mélangées à du texte ne sont pas pris en charge.
Délai d'expiration	Définissez le délai d'expiration de la vérification de l'élément (disponible pour les types d'éléments pris en charge). Sélectionnez l'option de délai d'expiration : **Global** - le délai d'expiration du proxy ou le délai global est utilisé (affiché dans le champ *Délai d'expiration* grisé). **Remplacement** - un délai d'expiration personnalisé est utilisé (défini dans le champ *Délai d'expiration* ; plage autorisée : 1 - 600s). Les suffixes de temps, par exemple 30s et 1m, ainsi que les macros utilisateur sont pris en charge. Cliquer sur le lien *Délais d'expiration* permet de configurer les délais d'expiration du proxy ou les délais d'expiration globaux (si aucun proxy n'est utilisé). Notez que le lien *Délais d'expiration* est visible uniquement pour les utilisateurs de type *Super administrateur* disposant des autorisations nécessaires pour les sections de l'interface *Administration* > *Général* ou *Administration* > *Proxies*.

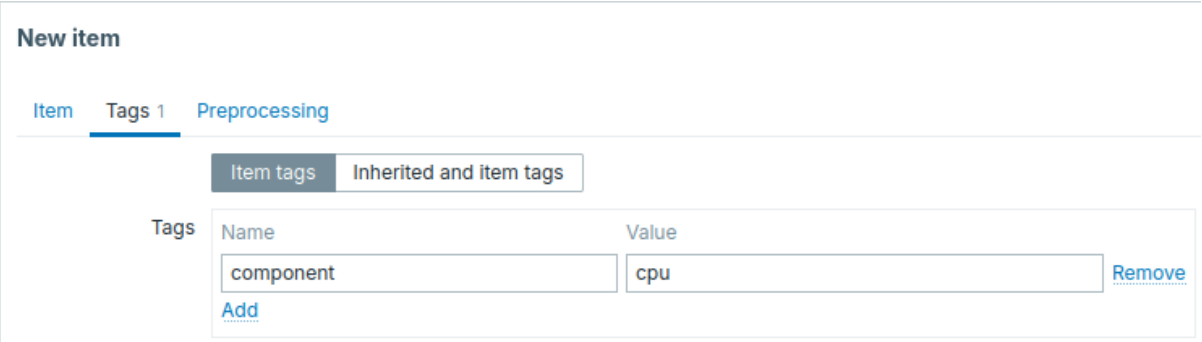
Paramètre	Description
Historique	Sélectionnez l'une des options suivantes : **Ne pas stocker** - l'historique de l'élément n'est pas stocké. Utile pour les éléments maîtres si seuls les éléments dépendants doivent conserver leur historique. Ce paramètre ne peut pas être remplacé par les paramètres globaux du housekeeper. **Stocker jusqu'à** - indiquez la durée de conservation de l'historique détaillé dans la base de données (de 1 heure à 25 ans). Les données plus anciennes seront supprimées par le housekeeper. La durée est stockée en secondes. Les suffixes de temps sont pris en charge, par exemple 2h, 1d. Les macros utilisateur sont prises en charge. La valeur *Stocker jusqu'à* peut être remplacée globalement dans *Administration* > *Housekeeper*. Si un paramètre global de remplacement existe, une icône d'information orange est affichée. Si vous placez le pointeur de la souris dessus, un message d'avertissement s'affiche, par exemple : *Remplacé par les paramètres globaux du housekeeper (1d)*. Il est recommandé de conserver les valeurs enregistrées pendant la durée la plus courte possible afin de réduire la taille de l'historique des valeurs dans la base de données. Au lieu de stocker un historique de valeurs long, vous pouvez conserver plus longtemps les données de tendance. Voir également Historique et tendances.
Tendances	Sélectionnez l'une des options suivantes : **Ne pas stocker** - les tendances ne sont pas stockées. Ce paramètre ne peut pas être remplacé par les paramètres globaux du housekeeper. **Stocker jusqu'à** - indiquez la durée de conservation de l'historique agrégé (minimum, maximum, moyenne et nombre horaires) dans la base de données (de 1 jour à 25 ans). Les données plus anciennes seront supprimées par le housekeeper. La durée est stockée en secondes. Les suffixes de temps sont pris en charge, par exemple 24h, 1d. Les macros utilisateur sont prises en charge. La valeur *Stocker jusqu'à* peut être remplacée globalement dans *Administration* > *Housekeeper*. Si un paramètre global de remplacement existe, une icône d'information orange est affichée. Si vous placez le pointeur de la souris dessus, un message d'avertissement s'affiche, par exemple : *Remplacé par les paramètres globaux du housekeeper (7d)*. *Remarque :* la conservation des tendances n'est pas disponible pour les données non numériques : caractère, journal et texte. Voir également Historique et tendances.
Mappage de valeurs	Appliquez un mappage de valeurs à cet élément. Le mappage de valeurs ne modifie pas les valeurs reçues ; il sert uniquement à afficher les données. Il fonctionne avec les éléments *Numérique (non signé)*, *Numérique (flottant)* et *Caractère*. Par exemple, « États des services Windows ».
Format de l'heure du journal	Disponible uniquement pour les éléments de type **Journal**. Espaces réservés pris en charge : **y** : *année (1970-2038)*. **M** : *mois (01-12)*. **d** : *jour (01-31)*. **h** : *heure (00-23)*. **m** : *minute (00-59)*. **s** : *seconde (00-59)*. Si le champ est laissé vide, l'horodatage sera défini sur 0 en temps Unix, ce qui représente le 1er janvier 1970. Par exemple, considérons la ligne suivante du fichier journal de Zabbix agent : « 23480:20100328:154718.045 Zabbix agent started. Zabbix 1.8.2 (revision 11211). » Elle commence par six positions de caractères pour le PID, suivies de la date, de l'heure et du reste du message. Le format de l'heure du journal pour cette ligne serait « ppppp:yyyyMMdd:hhmmss ». Notez que les caractères « p » et « : » sont des espaces réservés et peuvent être remplacés par n'importe quels caractères, à l'exception de « yMdhms ».
Remplit le champ d'inventaire de l'hôte	Vous pouvez sélectionner un champ d'inventaire de l'hôte que la valeur de l'élément remplira. Cela fonctionne si le remplissage automatique de l'inventaire est activé pour l'hôte et avec les éléments *Numérique (non signé)*, *Numérique (flottant)*, *Caractère* et *Texte*.
Description	Saisissez une description de l'élément. Les macros utilisateur sont prises en charge.
Activé	Cochez la case pour activer l'élément afin qu'il soit traité. Lorsque vous désactivez l'élément, il est immédiatement supprimé du cache de l'historique (à l'exception de sa dernière valeur, qui est conservée pour les journaux).

Paramètre	Description
Dernières données	Cliquez sur le lien pour afficher les dernières données de l'élément. Ce lien est disponible uniquement lors de la modification d'un élément existant.

Note:
Les champs spécifiques au type d'élément sont décrits sur les pages correspondantes.

Note:
Lors de la modification au niveau d'un hôte d'un élément existant au niveau d'un modèle, plusieurs champs sont en lecture seule. Vous pouvez utiliser le lien dans l'en-tête du formulaire pour accéder au niveau du modèle et les modifier à cet endroit, en gardant à l'esprit que les modifications effectuées au niveau du modèle modifieront l'élément pour tous les hôtes auxquels le modèle est lié.

L'onglet **Étiquettes** permet de définir les étiquettes au niveau de l'élément.



Prétraitement de la valeur de l'élément

L'onglet **Prétraitement** vous permet de définir des règles de transformation pour les valeurs reçues.

Test d'élément

Attention:
Pour effectuer le test d'un élément, assurez-vous que l'heure système sur le serveur et le proxy est synchronisée. Si l'heure du serveur est en retard, le test d'élément peut renvoyer le message d'erreur "The task has been expired." En revanche, le fait d'avoir défini des fuseaux horaires différents sur le serveur et le proxy n'affectera pas le résultat du test.

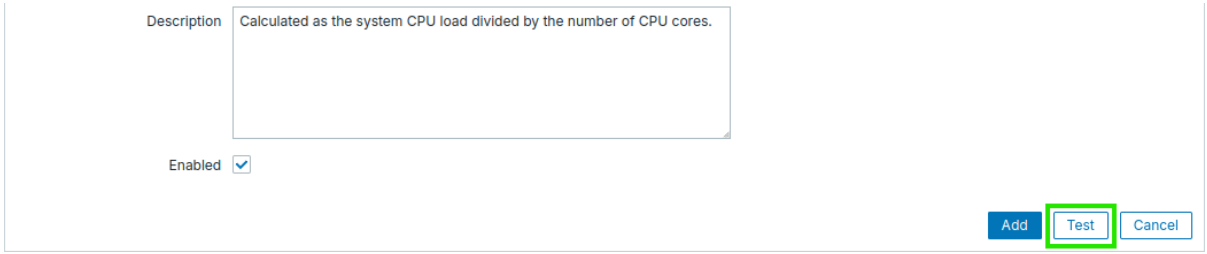
Il est possible de tester un élément et, s'il est correctement configuré, d'obtenir une valeur réelle en retour. Le test peut être effectué même avant l'enregistrement d'un élément.

Le test est disponible pour les éléments d'hôte et de modèle, les prototypes d'élément et les règles de découverte bas niveau. Le test n'est pas disponible pour les éléments actifs.

Le test d'élément est disponible pour les types d'élément passifs suivants :

- Zabbix agent
- SNMP agent (v1, v2, v3)
- IPMI agent
- SSH checks
- Telnet checks
- JMX agent
- Simple checks (except `icmpping*`, `vmware.*` items)
- Zabbix internal
- Calculated items
- External checks
- Database monitor
- HTTP agent
- Script
- Browser

Pour tester un élément, cliquez sur le bouton *Test* en bas du formulaire de configuration de l'élément. Notez que le bouton *Test* sera désactivé pour les éléments qui ne peuvent pas être testés (comme les vérifications actives, les vérifications simples exclues).

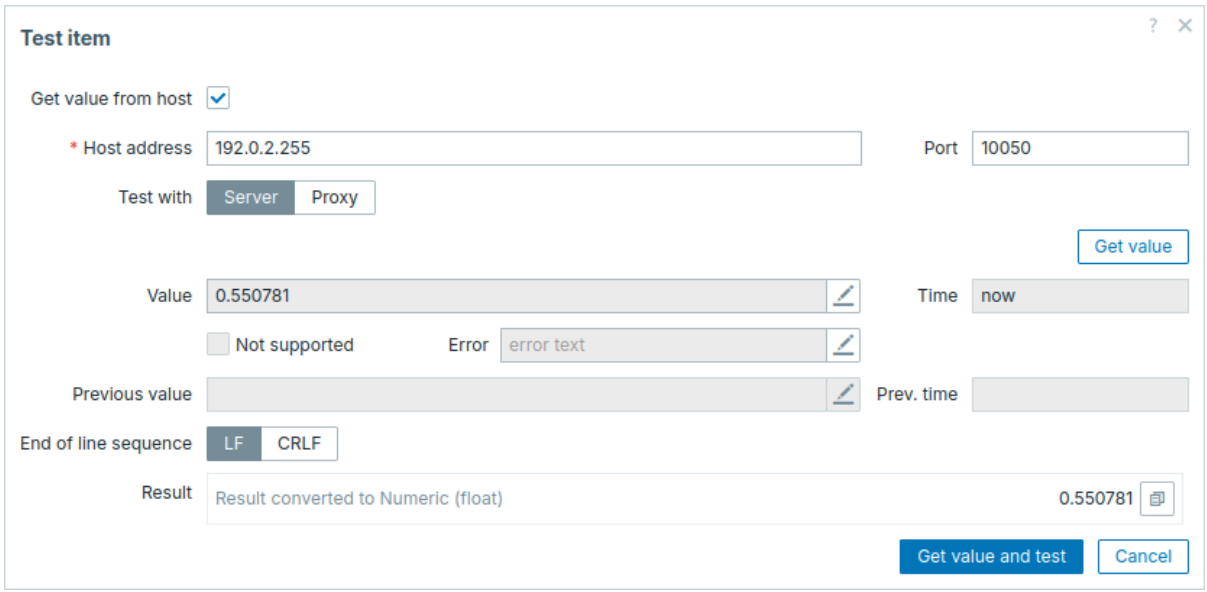


Le formulaire de test d'élément comporte des champs pour les paramètres d'hôte requis (adresse de l'hôte, port, test avec serveur/proxy (nom du proxy)) et des détails spécifiques à l'élément (tels que la communauté SNMPv2 ou les identifiants de sécurité SNMPv3). Ces champs sont contextuels :

- Les valeurs sont préremplies lorsque cela est possible, c'est-à-dire, pour les éléments nécessitant un agent, en prenant les informations de l'interface agent sélectionnée de l'hôte.
- Les valeurs doivent être saisies manuellement pour les éléments de modèle.
- Les valeurs des macros en texte brut sont résolues.
- Les champs dont la valeur (ou une partie de la valeur) est un secret ou une macro Vault sont vides et doivent être saisis manuellement. Si un paramètre d'élément contient une valeur de macro secrète, le message d'avertissement suivant s'affiche : "Item contains user-defined macros with secret values. Values of these macros should be entered manually."
- Les champs sont désactivés lorsqu'ils ne sont pas nécessaires dans le contexte du type d'élément (par exemple, le champ d'adresse de l'hôte et le champ du proxy sont désactivés pour les éléments calculés)

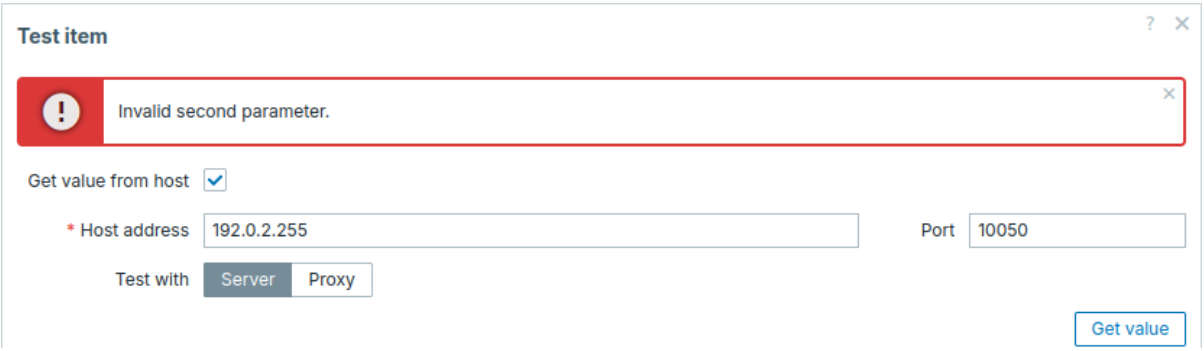
Pour tester l'élément, cliquez sur *Get value*. Si la valeur est récupérée avec succès, elle remplira le champ *Value*, en déplaçant la valeur actuelle (le cas échéant) vers le champ *Previous value* tout en calculant également le champ *Prev. time*, c'est-à-dire la différence de temps entre les deux valeurs (clics), et en essayant de détecter une séquence EOL et de passer à CRLF si "\n\r" est détecté dans la valeur récupérée.

Cliquez sur *Get value and test* pour tester le prétraitement.



Les valeurs récupérées depuis un hôte et les résultats de test sont tronqués à une taille maximale de 512 KB lorsqu'ils sont envoyés à l'interface. Si un résultat de test est tronqué, une icône d'avertissement s'affiche ; survolez-la pour plus de détails. Si une valeur est tronquée, elle peut échouer à la validation du type de données lors de tests consécutifs comparant de grandes valeurs (par exemple, JSON) ; dans ce cas, un message d'erreur s'affiche. Notez que les données supérieures à 512 KB sont toujours entièrement traitées par le serveur Zabbix.

Si la configuration est incorrecte, un message d'erreur décrivant la cause possible s'affiche.



Une valeur récupérée avec succès depuis l'hôte peut également être utilisée pour tester les étapes de prétraitement.

Boutons du formulaire

Les boutons situés en bas du formulaire vous permettent d'effectuer plusieurs opérations.

Add	Ajouter un élément. Ce bouton est disponible uniquement pour les nouveaux éléments.
Update	Mettre à jour les propriétés d'un élément.
Clone	Créer un autre élément à partir des propriétés de l'élément actuel.
Execute now	Exécuter immédiatement une vérification pour obtenir une nouvelle valeur d'élément. Pris en charge uniquement pour les vérifications **passives** (voir plus de détails). *Remarque* : lors de la vérification immédiate d'une valeur, le cache de configuration n'est pas mis à jour ; la valeur ne reflétera donc pas les modifications très récentes de la configuration de l'élément.
Test	Tester si la configuration de l'élément est correcte en obtenant une valeur.
Clear history and trends	Supprimer l'historique et les tendances de l'élément.
Delete	Supprimer l'élément.
Cancel	Annuler la modification des propriétés de l'élément.

Limites des données d'élément

Les limites des données d'élément dépendent du type de données de l'élément et du backend de base de données.

Les valeurs numériques (non signées) sont stockées telles quelles. Si une valeur à virgule flottante est reçue, seule la partie entière est stockée (par exemple, 1.23 → 1).

Les valeurs numériques (float) sont stockées telles quelles et prennent en charge environ 15 à 17 chiffres de précision, avec une plage d'environ -1.79E+308 à 1.79E+308. Les valeurs en notation scientifique sont également prises en charge (par exemple, 1.23E+7, 1e308, 1.1E-4).

Les valeurs textuelles sont tronquées avant d'être stockées afin de respecter la limite du type de valeur de la base de données :

Base de données	Type d'information		
	Caractère	**Journal**	**Texte**
MySQL	255 caractères	65536 octets	65536 octets
PostgreSQL	255 caractères	65536 caractères	65536 caractères
SQLite (uniquement proxy Zabbix)	255 caractères	65536 caractères	65536 caractères

Les valeurs binaires sont limitées à 16 MiB (16777216 octets). Les valeurs qui dépassent cette limite sont ignorées, et une erreur correspondante est affichée dans l'interface.

Les valeurs avec le type de données JSON sont limitées à 128 MiB (134217728 octets). Les valeurs qui dépassent cette limite sont ignorées, et une erreur correspondante est affichée dans l'interface. Pour MySQL/MariaDB, envisagez de configurer des partitions quotidiennes pour la table `history_json` (utilisée pour stocker les valeurs JSON), car elle peut grossir rapidement et verrouiller le housekeeper pendant une période prolongée. Si votre cas d'utilisation implique de grandes valeurs JSON (1 MiB ou plus), consultez la procédure de configuration de votre système pour la prise en charge des grandes valeurs JSON.

Lorsque Zabbix écrit des données d'élément dans la base de données, il utilise des requêtes INSERT et les journalise (si `DebugLevel` est défini sur 4 ou 5). Les grandes entrées de journal sont tronquées à 64 Ko.

Limite des scripts personnalisés

La longueur disponible pour les scripts personnalisés dépend de la base de données utilisée :

Base de données	Limite en caractères	Limite en octets
MySQL	65535	65535
PostgreSQL	65535	non limitée
SQLite (uniquement Zabbix proxy)	65535	non limitée

Délai d'expiration de l'élément

Le délai d'expiration de l'élément spécifie combien de temps Zabbix doit attendre avant d'interrompre la vérification en la considérant comme échouée.

Lorsque le délai d'expiration est atteint, la vérification est interrompue même si la récupération des données n'est pas terminée. Si les données sont reçues partiellement, l'élément devient non pris en charge et un message d'erreur est consigné (par exemple, si les données sont collectées avec succès pour un seul des plusieurs OID dans une vérification SNMP).

Pour de nombreux types d'éléments, vous pouvez définir des délais d'expiration **flexibles** pour les éléments:

- pour un seul élément
- par type d'élément (au niveau d'un proxy)
- par type d'élément (au niveau global interface)

Un délai d'expiration personnalisé *pour un seul élément* est utile si vous souhaitez spécifier un délai plus long pour un élément particulier, tout en conservant des délais courts pour les autres éléments.

Priorité des délais d'expiration

1. Le délai d'expiration d'un seul élément remplace tout autre délai.
2. Les délais d'expiration au niveau du proxy remplacent les délais d'expiration globaux.

Le délai d'expiration provenant de la configuration du serveur Zabbix ou du proxy ne joue **aucun** rôle dans les vérifications pour lesquelles des délais d'expiration flexibles sont définis.

Prise en charge des délais d'attente flexibles

Les délais d'attente flexibles des éléments sont pris en charge pour les types d'éléments suivants :

- agent Zabbix (vérifications passives et actives)
- agent SNMP (sauf les vérifications SNMP héritées[1])
- Vérification simple (sauf `icmpping*` et les éléments VMware[2])
- agent SSH
- agent Telnet
- Vérification externe
- Surveillance de base de données
- Script
- agent HTTP
- Navigateur

[1] Pour les vérifications SNMP héritées (interrogation d'un seul OID), les paramètres de délai d'attente du serveur ou du proxy s'appliquent.
 [2] Pour les éléments `icmpping*`, la valeur du délai d'attente est spécifiée directement dans la clé de l'élément ; une valeur maximale codée en dur de 600 secondes s'applique également. Les éléments de surveillance VMware utilisent leur propre paramètre VMwareTimeout issu de la configuration du serveur ou du proxy.

Éléments non pris en charge

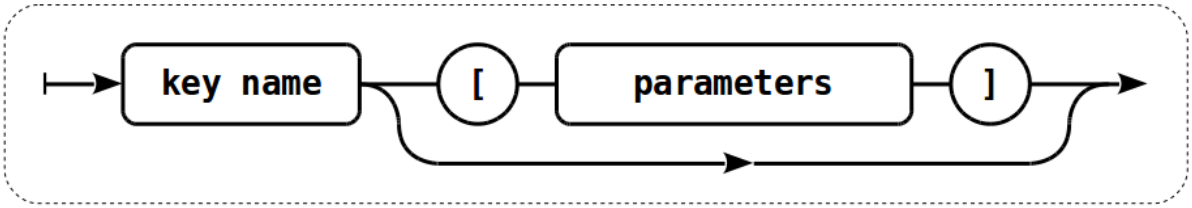
Les éléments non pris en charge sont signalés avec le statut *Non pris en charge* et sont toujours revérifiés selon leur *Intervalle de mise à jour* standard.

Un élément devient non pris en charge si sa valeur ne peut pas être récupérée pour une raison quelconque (par exemple, une erreur de connexion, aucun poller configuré pour traiter l'élément). De plus, les éléments qui ne reçoivent pas de données historiques restent dans le statut *Non pris en charge*. Il s'agit du comportement attendu : un élément passe au statut *Normal* uniquement lorsqu'il reçoit de nouvelles données historiques valides.

Comme les problèmes, les éléments non pris en charge ne sont réévalués que lorsque de nouvelles données sont reçues, même s'il n'y a plus de données historiques disponibles pour cet élément. En d'autres termes, les éléments et les déclencheurs changent d'état uniquement lors de la réception de nouvelles données. Cependant, si une expression de déclencheur inclut une fonction date and time et/ou nodata(), le déclencheur sera recalculé toutes les 30 secondes (voir triggers calculation time pour plus de détails).

Format de clé d’élément

Le format de clé d’élément, y compris les paramètres de clé, doit respecter des règles de syntaxe. Les illustrations suivantes présentent la syntaxe prise en charge. Les éléments et caractères autorisés à chaque point peuvent être déterminés en suivant les flèches - si un bloc peut être atteint par la ligne, il est autorisé, sinon - il ne l’est pas.



Pour construire une clé d’élément valide, on commence par spécifier le nom de la clé, puis on choisit soit d’ajouter des paramètres, soit de ne pas en ajouter - comme l’indiquent les deux lignes possibles à suivre.

Nom de la clé

Le nom de la clé elle-même a une gamme limitée de caractères autorisés, qui se suivent les uns les autres. Les caractères autorisés sont :

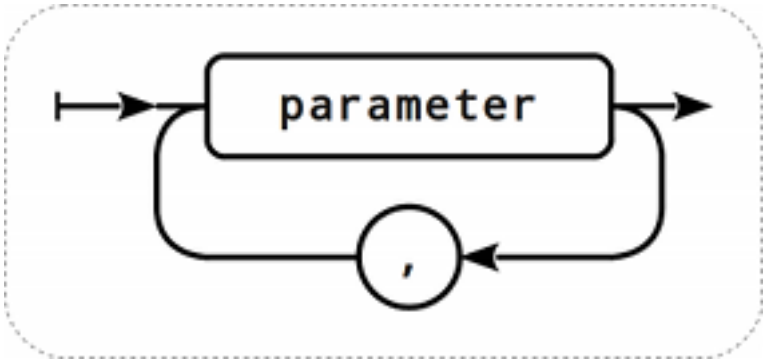
`0-9a-zA-Z_-.`

Ce qui signifie :

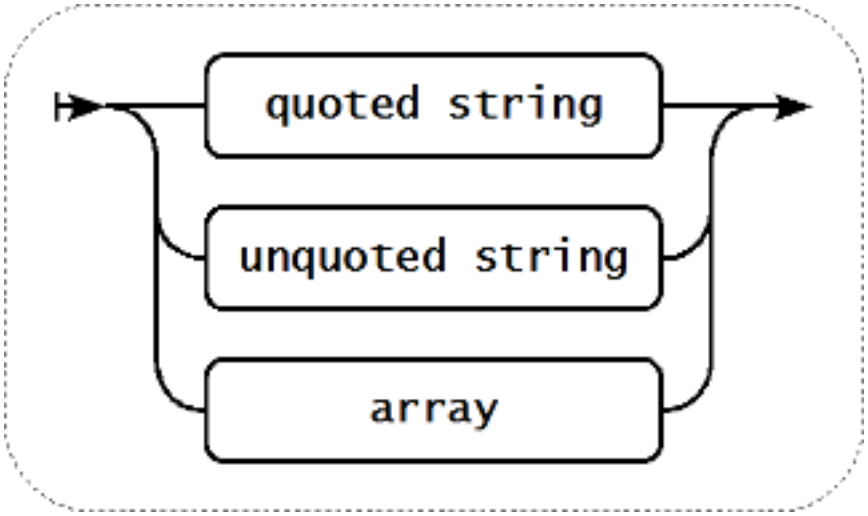
- tous les nombres ;
- toutes les lettres minuscules ;
- toutes les lettres majuscules ;
- underscore ;
- tiret ;
- point.

Paramètres de clé

Une clé d’élément peut avoir plusieurs paramètres séparés par des virgules.



Chaque paramètre de clé peut être soit une chaîne entre guillemets, une chaîne sans guillemets ou un tableau.

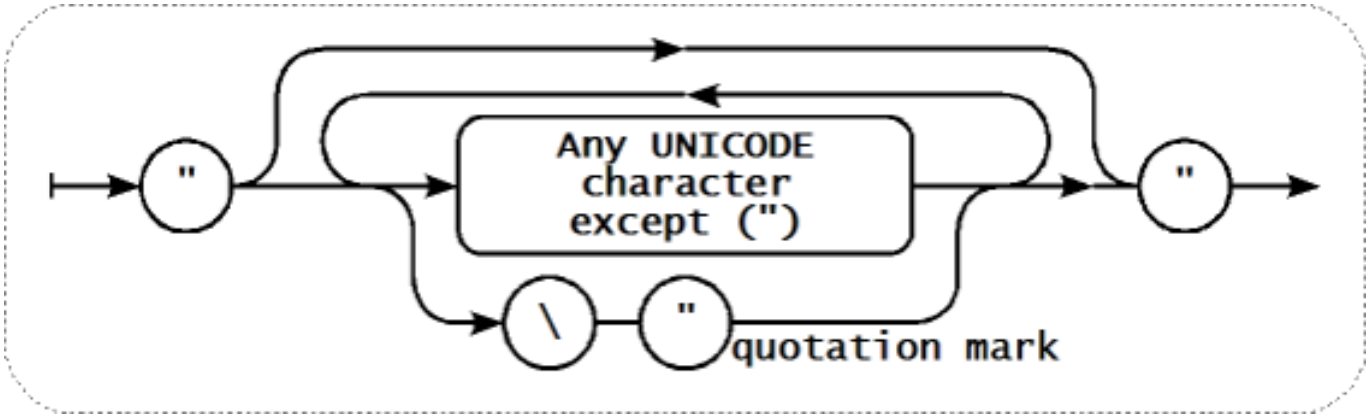


Le paramètre peut également être laissé vide, utilisant ainsi la valeur par défaut. Dans ce cas, le nombre approprié de virgules doit être ajouté si d'autres paramètres sont spécifiés. Par exemple, la clé d'élément **icmpping[„200„500]** indiquerait que l'intervalle entre les pings individuels est de 200 millisecondes, le délai d'expiration de 500 millisecondes, et tous les autres paramètres sont laissés à leurs valeurs par défaut.

Il est possible d'inclure des macros dans les paramètres. Il peut s'agir de macros utilisateur ou de certaines macros intégrées. Pour voir quelles macros intégrées particulières sont prises en charge dans les paramètres de clé d'élément, recherchez « item key parameters » sur la page Macros prises en charge.

Paramètre - chaîne entre guillemets

Si le paramètre de clé est une chaîne entre guillemets, tout caractère Unicode est autorisé. Si la chaîne du paramètre de clé contient un guillemet, ce paramètre doit être placé entre guillemets, et chaque guillemet faisant partie de la chaîne du paramètre doit être précédé d'un caractère barre oblique inverse (\). Si la chaîne du paramètre de clé contient une virgule, ce paramètre doit être placé entre guillemets.



Warning:
Pour mettre les paramètres de clé d'élément entre guillemets, utilisez uniquement des guillemets doubles. Les guillemets simples ne sont pas pris en charge.

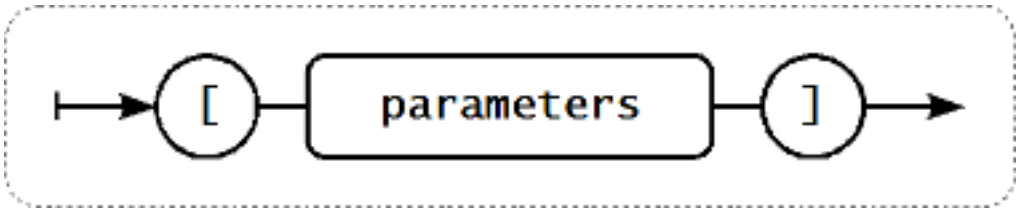
Attention:
Les tableaux de paramètres à plusieurs niveaux, par ex. `[a,[b,[c,d]],e]`, ne sont pas autorisés.

Paramètre - chaîne sans guillemet

Si le paramètre clé est une chaîne sans guillemet, tout caractère Unicode est autorisé sauf la virgule et le crochet de droite (]). Le paramètre sans guillemet ne peut pas commencer avec le crochet gauche ([).

Paramètre - array

Si le paramètre de clé est un array, il est à nouveau placé entre crochets, où les paramètres individuels sont conformes aux règles et à la syntaxe de spécification de plusieurs paramètres.



Intervalles personnalisés

Vue d'ensemble

Il est possible de créer des règles personnalisées concernant les moments où un élément est vérifié. Les deux méthodes disponibles sont les *intervalles flexibles*, qui permettent de redéfinir l'intervalle de mise à jour par défaut, et la *planification*, qui permet d'exécuter la vérification d'un élément à un moment précis ou selon une séquence de moments.

Intervalles flexibles

Les intervalles flexibles vous permettent de redéfinir l'intervalle de mise à jour par défaut pour des périodes spécifiques.

Un intervalle flexible est défini avec *Interval* et *Period* :

- *Interval* - l'intervalle de mise à jour pour la période spécifiée. Les suffixes de temps sont pris en charge, par exemple : 30s, 1m, 2h, 1d.
- *Period* - la période pendant laquelle l'intervalle flexible est actif (consultez les périodes pour obtenir une description détaillée du format de *Period*)

Si plusieurs intervalles flexibles se chevauchent, la plus petite valeur d'*Interval* est utilisée pour la période de chevauchement. Notez que si la plus petite valeur des intervalles flexibles qui se chevauchent est « 0 », aucune interrogation n'aura lieu. En dehors des intervalles flexibles, l'intervalle de mise à jour par défaut est utilisé.

Notez que si l'intervalle flexible est égal à la durée de la période, l'élément sera vérifié exactement une fois. Si l'intervalle flexible est supérieur à la période, l'élément peut être vérifié une fois ou ne pas être vérifié du tout (une telle configuration n'est donc pas recommandée). Si l'intervalle flexible est inférieur à la période, l'élément sera vérifié au moins une fois.

Si l'intervalle flexible est défini sur « 0 », l'élément n'est pas interrogé pendant la période de l'intervalle flexible et reprend les interrogations selon l'*intervalle de mise à jour* par défaut une fois la période terminée. Exemples :

Interval	Period	Description
10	1-5,09:00-18:00	L'élément sera vérifié toutes les 10 secondes pendant les heures de travail.
0	1-7,00:00-7:00	L'élément ne sera pas vérifié pendant la nuit.
0	7-7,00:00-24:00	L'élément ne sera pas vérifié le dimanche.
60	1-7,12:00-12:01	L'élément sera vérifié à 12:00 chaque jour. Notez que cette configuration était utilisée comme solution de contournement pour les vérifications planifiées ; il est recommandé d'utiliser des intervalles de planification pour ces vérifications.

Intervalles de planification

Les intervalles de planification sont utilisés pour vérifier les éléments à des moments précis. Alors que les intervalles flexibles sont conçus pour redéfinir l'intervalle de mise à jour par défaut d'un élément, les intervalles de planification servent à spécifier un calendrier de vérification indépendant, exécuté en parallèle.

Un intervalle de planification est défini comme suit : `md<filter>wd<filter>h<filter>m<filter>s<filter>` où :

- **md** - jours du mois
- **wd** - jours de la semaine
- **h** - heures
- **m** - minutes
- **s** - secondes

`<filter>` est utilisé pour spécifier des valeurs pour son préfixe (jours, heures, minutes, secondes) et est défini comme suit : `[<from>[-<to>]][/<step>][,<filter>]` où :

- `<from>` et `<to>` définissent la plage des valeurs correspondantes (incluse). Si `<to>` est omis, alors le filtre correspond à une plage `<from> - <from>`. Si `<from>` est également omis, alors le filtre correspond à toutes les valeurs possibles.
- `<step>` définit le pas des valeurs numériques dans la plage. Par défaut, `<step>` a la valeur 1, ce qui signifie que toutes les valeurs de la plage définie correspondent.

Bien que les définitions de filtre soient facultatives, au moins un filtre doit être utilisé. Un filtre doit avoir soit une plage, soit la valeur *<step>* définie.

Un filtre vide correspond soit à '0' si aucun filtre de niveau inférieur n'est défini, soit à toutes les valeurs possibles dans le cas contraire. Par exemple, si le filtre des heures est omis, alors seule l'heure '0' correspondra, à condition que les filtres des minutes et des secondes soient également omis ; sinon, un filtre des heures vide correspondra à toutes les valeurs d'heure.

Les valeurs valides de `<from>` et `<to>` pour le préfixe de filtre correspondant sont :

Prefix	Description	*<from>*	*<to>*
md	Jours du mois	1-31	1-31
wd	Jours de la semaine	1-7	1-7
h	Heures	0-23	0-23
m	Minutes	0-59	0-59
s	Secondes	0-59	0-59

La valeur `<from>` doit être inférieure ou égale à la valeur `<to>`. La valeur `<step>` doit être supérieure ou égale à 1 et inférieure ou égale à `<to> - <from>`.

Les valeurs à un seul chiffre pour les jours du mois, les heures, les minutes et les secondes peuvent être préfixées par 0. Par exemple, `md01-31` et `h/02` sont des intervalles valides, mais `md01-031` et `wd01-07` ne le sont pas.

Dans l'interface Zabbix, plusieurs intervalles de planification sont saisis sur des lignes distinctes. Dans l'API Zabbix, ils sont concaténés en une seule chaîne avec un point-virgule `;` comme séparateur.

Si une heure correspond à plusieurs intervalles, elle n'est exécutée qu'une seule fois. Par exemple, `wd1h9;h9` ne sera exécuté qu'une seule fois le lundi à 9h00.

Exemples :

Interval	Will be executed
m0-59	chaque minute
h9-17/2	toutes les 2 heures à partir de 9:00 (9:00, 11:00 ...)
m0,30 or m/30	toutes les heures à hh:00 et hh:30
m0,5,10,15,20,25,30,35,40,45,50,55 or m/5	toutes les cinq minutes
wd1-5h9	tous les lundis à vendredis à 9:00
wd1-5h9-18	tous les lundis à vendredis à 9:00,10:00,...,18:00
h9,10,11 or h9-11	tous les jours à 9:00, 10:00 et 11:00
md1h9m30	le 1er jour de chaque mois à 9:30
md1wd1h9m30	le 1er jour de chaque mois à 9:30 si c'est un lundi
h9m/30	tous les jours à 9:00, 9:30
h9m0-59/30	tous les jours à 9:00, 9:30
h9,10m/30	tous les jours à 9:00, 9:30, 10:00, 10:30
h9-10m30	tous les jours à 9:30, 10:30
h9m10-40/30	tous les jours à 9:10, 9:40
h9,10m10-40/30	tous les jours à 9:10, 9:40, 10:10, 10:40
h9-10m10-40/30	tous les jours à 9:10, 9:40, 10:10, 10:40
h9m10-40	tous les jours à 9:10, 9:11, 9:12, ... 9:40
h9m10-40/1	tous les jours à 9:10, 9:11, 9:12, ... 9:40
h9-12,15	tous les jours à 9:00, 10:00, 11:00, 12:00, 15:00
h9-12,15m0	tous les jours à 9:00, 10:00, 11:00, 12:00, 15:00
h9-12,15m0s30	tous les jours à 9:00:30, 10:00:30, 11:00:30, 12:00:30, 15:00:30
h9-12s30	tous les jours à 9:00:30, 9:01:30, 9:02:30 ... 12:58:30, 12:59:30
h9m/30;h10 (*syntaxe spécifique à l'API*)	tous les jours à 9:00, 9:30, 10:00
h9m/30 h10 (*ajoutez ceci comme une autre ligne dans l'interface*)	tous les jours à 9:00, 9:30, 10:00

Alignement des fuseaux horaires pour les proxies et l'agent

Notez que les proxies Zabbix et l'agent utilisent leurs fuseaux horaires locaux lors du traitement des intervalles de planification.

Pour cette raison, lorsque des intervalles de planification sont appliqués à des éléments supervisés par un proxy Zabbix ou à des éléments actifs de l'agent, il est recommandé de définir le fuseau horaire des proxies ou de l'agent concernés identique à celui du serveur Zabbix ; sinon, la queue peut signaler incorrectement des retards d'éléments.

Le fuseau horaire du proxy Zabbix ou de l'agent peut être défini à l'aide de la variable d'environnement TZ dans le fichier d'unité `systemd` :

```
[Service]
...
Environment="TZ=Europe/Amsterdam"
```

Prétraitement de la valeur de l'élément

Vue d'ensemble

Le prétraitement vous permet d'appliquer des transformations aux valeurs d'élément reçues avant de les enregistrer dans la base de données. Ces transformations/étapes de prétraitement sont effectuées par le serveur Zabbix ou le proxy (si les éléments sont supervisés par un proxy).

Cette fonctionnalité prend en charge divers cas d'utilisation, tels que :

- convertir des octets en bits (par exemple, multiplier les valeurs de trafic réseau par « 8 ») ;

- calculer des statistiques par seconde pour des valeurs augmentant de manière incrémentielle ;
- appliquer des expressions régulières pour extraire ou modifier des valeurs ;
- exécuter des scripts personnalisés sur des valeurs ;
- ignorer les valeurs inchangées afin d'optimiser le stockage dans la base de données.

Une ou plusieurs étapes de prétraitement peuvent être configurées pour un élément. Ces étapes sont exécutées dans l'ordre dans lequel elles sont configurées.

Attention:
Si une étape de prétraitement échoue, un élément devient non pris en charge. Cela peut être évité grâce à la gestion des erreurs *Personnalisé en cas d'échec* (disponible pour la plupart des transformations), qui vous permet d'ignorer des valeurs ou de définir des valeurs personnalisées.

 Pour les éléments de journal, les métadonnées du journal (sans valeur) réinitialisent toujours l'état non pris en charge de l'élément, ce qui le rend à nouveau pris en charge. Cela se produit même si l'erreur initiale s'est produite après la réception d'une valeur de journal provenant de l'agent.

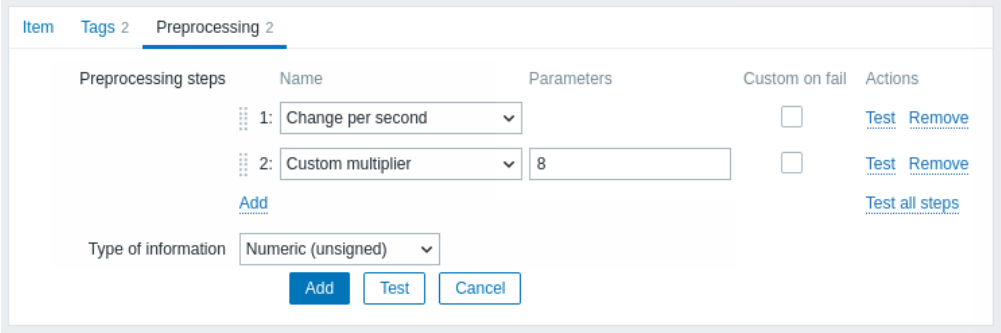
Toutes les valeurs transmises au prétraitement sont initialement traitées comme des chaînes de caractères. La conversion vers le type de valeur souhaité (tel que défini dans la configuration de l'élément) est effectuée à la fin du pipeline de prétraitement. Cependant, certaines étapes de prétraitement peuvent déclencher des conversions plus tôt, si nécessaire. Pour des informations techniques détaillées, voir Détails du prétraitement.

Pour vous assurer que votre configuration de prétraitement fonctionne comme prévu, vous pouvez la tester.

Voir aussi : Exemples de prétraitement

Configuration

Les étapes de prétraitement sont définies dans l'onglet **Prétraitement** du formulaire de configuration de l'élément.



Cliquez sur *Ajouter* pour sélectionner une transformation prise en charge.

Le champ *Type d'information* s'affiche en bas de l'onglet lorsqu'au moins une étape de prétraitement est définie. Si nécessaire, il est possible de modifier le type d'information sans quitter l'onglet *Prétraitement*. Consultez Créer un élément pour obtenir une description détaillée des paramètres.

Transformations prises en charge

Toutes les transformations prises en charge sont répertoriées ci-dessous. Cliquez sur le nom d'une transformation pour afficher tous ses détails.

Name	Description	Type
Expression régulière	Faire correspondre la valeur à l'expression régulière et la remplacer par la sortie requise.	Text
Remplacement	Rechercher la chaîne indiquée et la remplacer par une autre chaîne (ou par rien).	
Suppression des espaces	Supprimer les caractères indiqués au début et à la fin de la valeur.	
Suppression des espaces à droite	Supprimer les caractères indiqués à la fin de la valeur.	
Suppression des espaces à gauche	Supprimer les caractères indiqués au début de la valeur.	
XML XPath	Extraire une valeur ou un fragment de données XML à l'aide de la fonctionnalité XPath.	Données structurées
JSON Path	Extraire une valeur ou un fragment de données JSON à l'aide de la fonctionnalité JSONPath.	
CSV vers JSON	Convertir les données d'un fichier CSV au format JSON.	
XML vers JSON	Convertir des données au format XML en JSON.	

Name	Description	Type
Valeur SNMP walk	Extraire une valeur à l'aide du nom OID/MIB indiqué et appliquer les options de formatage.	SNMP
SNMP walk vers JSON	Convertir les valeurs SNMP en JSON.	
Valeur SNMP get	Appliquer les options de formatage à la valeur SNMP get.	
Multiplicateur personnalisé	Multiplier la valeur par la valeur entière ou à virgule flottante indiquée.	Arithmétique
Changement simple	Calculer la différence entre la valeur actuelle et la valeur précédente.	Changement
Changement par seconde	Calculer la vitesse de variation de la valeur (différence entre la valeur actuelle et la valeur précédente) par seconde.	
Booléen vers décimal	Convertir la valeur du format booléen au format décimal.	Systèmes numériques
Octal vers décimal	Convertir la valeur du format octal au format décimal.	
Hexadécimal vers décimal	Convertir la valeur du format hexadécimal au format décimal.	
JavaScript	Saisir du code JavaScript.	Scripts personnalisés
Dans l'intervalle	Définir l'intervalle dans lequel une valeur doit se trouver.	Validation
Correspond à une expression régulière	Indiquer une expression régulière à laquelle une valeur doit correspondre.	
Ne correspond pas à une expression régulière	Indiquer une expression régulière à laquelle une valeur ne doit pas correspondre.	
Vérifier les erreurs dans JSON	Vérifier la présence d'un message d'erreur au niveau de l'application situé à l'emplacement JSONPath.	
Vérifier les erreurs dans XML	Vérifier la présence d'un message d'erreur au niveau de l'application situé à l'emplacement XPath.	
Vérifier les erreurs à l'aide d'une expression régulière	Vérifier la présence d'un message d'erreur au niveau de l'application à l'aide d'une expression régulière.	
Vérifier l'absence de valeur non prise en charge	Vérifier si aucune valeur d'élément n'a pu être récupérée.	
Ignorer les valeurs inchangées	Ignorer une valeur si elle n'a pas changé.	Limitation
Ignorer les valeurs inchangées avec pulsation	Ignorer une valeur si elle n'a pas changé pendant la période définie.	
Modèle Prometheus	Utiliser la requête suivante pour extraire les données requises des métriques Prometheus.	Prometheus
Prometheus vers JSON	Convertir les métriques Prometheus requises en JSON.	

Si la case *Personnaliser en cas d'échec* est activée pour la transformation sélectionnée, vous pouvez la cocher et indiquer des options personnalisées de gestion des erreurs : *Ignorer la valeur* (ignorer la valeur), *Définir la valeur sur* (définir une valeur indiquée) ou *Définir l'erreur sur* (définir un message d'erreur indiqué). Si une étape de prétraitement échoue, l'élément ne deviendra pas non pris en charge lorsque l'option *Ignorer la valeur* ou *Définir la valeur sur* est sélectionnée.

Notez que pour les étapes de prétraitement *Changement* et *Limitation*, Zabbix doit mémoriser la dernière valeur afin de calculer ou de comparer la nouvelle valeur comme requis. Ces valeurs précédentes sont gérées par le gestionnaire de prétraitement. Si le serveur Zabbix ou le proxy est redémarré, ou si une modification est apportée aux étapes de prétraitement, la dernière valeur de l'élément correspondant est réinitialisée, ce qui entraîne les conséquences suivantes :

- pour les étapes *Changement simple* et *Changement par seconde* : la valeur suivante sera ignorée, car aucune valeur précédente ne permet de calculer le changement ;
- pour les étapes *Ignorer les valeurs inchangées* et *Ignorer les valeurs inchangées avec pulsation* : la valeur suivante ne sera jamais ignorée, même si elle aurait dû l'être selon les règles d'ignorance.

Expression régulière

Faites correspondre la valeur à l'expression régulière et remplacez-la par la sortie requise.

Paramètres :

- **pattern** - l'expression régulière ;

- **output** - le modèle de formatage de la sortie. Une séquence d'échappement \N (où N=1...9) est remplacée par le N^{e} groupe correspondant. Une séquence d'échappement \0 est remplacée par le texte correspondant.

Commentaires :

- Si la valeur d'entrée ne correspond pas, l'élément devient non pris en charge ;
- L'expression régulière prend en charge l'extraction de 10 groupes capturés au maximum avec la séquence \N ;
- Consultez la section expressions régulières pour voir quelques exemples existants.

Remplacer

Rechercher la chaîne et la remplacer par une autre chaîne (ou par rien).

Paramètres :

- **chaîne de recherche** - chaîne à rechercher et à remplacer, sensible à la casse (obligatoire) ;
- **remplacement** - chaîne par laquelle remplacer la chaîne de recherche. La chaîne de remplacement peut également être vide, ce qui permet de supprimer la chaîne de recherche lorsqu'elle est trouvée.

Commentaires :

- Toutes les occurrences de la chaîne de recherche seront remplacées ;
- Il est possible d'utiliser des séquences d'échappement pour rechercher ou remplacer des sauts de ligne, des retours chariot, des tabulations et des espaces "\n \r \t \s" ; la barre oblique inverse peut être échappée sous la forme "\\" et les séquences d'échappement peuvent être échappées sous la forme "\\n" ;
- L'échappement des sauts de ligne, des retours chariot et des tabulations est effectué automatiquement lors de la découverte de bas niveau.

Supprimer

Supprime les caractères spécifiés au début et à la fin de la valeur.

Suppression à droite

Supprime les caractères spécifiés à la fin de la valeur.

Suppression à gauche

Supprime les caractères spécifiés du début de la valeur.

XML XPath

Extraire une valeur ou un fragment de données XML à l'aide de la fonctionnalité XPath.

Commentaires :

- Pour que cette option fonctionne, le serveur Zabbix (ou le proxy Zabbix) doit être compilé avec la prise en charge de libxml ;
- Les espaces de noms ne sont pas pris en charge.

Exemples :

```
number(/document/item/value) #extraiera '10' de <document><item><value>10</value></item></document>
number(/document/item/@attribute) #extraiera '10' de <document><item attribute="10"></item></document>
/document/item #extraiera '<item><value>10</value></item>' de <document><item><value>10</value></item></do
```

JSONPath

Extraire une valeur ou un fragment de données JSON à l'aide de la fonctionnalité JSONPath.

CSV vers JSON

Convertissez les données d'un fichier CSV au format JSON.

Pour plus d'informations, voir : prétraitement CSV vers JSON.

XML vers JSON

Convertit les données au format XML en JSON.

Pour plus d'informations, consultez : Règles de sérialisation.

Valeur de parcours SNMP

Extraire la valeur à l'aide de l'OID/nom MIB spécifié et appliquer les options de formatage :

- **Inchangée** - renvoyer la chaîne hexadécimale sous forme de chaîne hexadécimale non échappée (*notez* que les indications d'affichage sont toujours appliquées) ;

- **UTF-8 depuis une chaîne HEX** - convertir la chaîne hexadécimale en chaîne UTF-8 ;
- **MAC depuis une chaîne HEX** - valider la chaîne hexadécimale comme adresse MAC et renvoyer une chaîne d'adresse MAC correcte (où les ' ' sont remplacés par ':') ;
- **Entier depuis BITS** - convertir les 8 premiers octets d'une chaîne de bits exprimée sous forme de séquence de caractères hexadécimaux (par exemple, « 1A 2B 3C 4D ») en entier non signé de 64 bits. Dans les chaînes de bits de plus de 8 octets, les octets suivants seront ignorés.

SNMP walk vers JSON

Convertit les valeurs SNMP en JSON.

Spécifiez un nom de champ dans le JSON et le chemin OID SNMP correspondant. Les valeurs des champs seront renseignées avec les valeurs du chemin OID SNMP spécifié.

Commentaires :

- Des options de formatage des valeurs similaires à celles de l'étape *Valeur SNMP walk* sont disponibles ;
- Vous pouvez utiliser cette étape de prétraitement pour la découverte des OID SNMP.

Valeur obtenue via SNMP

Appliquer les options de formatage à la valeur obtenue via SNMP :

- **UTF-8 depuis Hex-STRING** - convertir la chaîne hexadécimale en chaîne UTF-8 ;
- **MAC depuis Hex-STRING** - valider la chaîne hexadécimale comme adresse MAC et renvoyer une chaîne d'adresse MAC correcte (où les ' ' sont remplacés par des ':') ;
- **Entier depuis BITS** - convertir les 8 premiers octets d'une chaîne de bits exprimée sous forme de séquence de caractères hexadécimaux (par exemple, "1A 2B 3C 4D") en un entier non signé de 64 bits. Dans les chaînes de bits de plus de 8 octets, les octets suivants seront ignorés.

Multiplicateur personnalisé

Multiplie la valeur par la valeur entière ou à virgule flottante spécifiée.

Commentaires :

- Utilisez cette option pour convertir les valeurs reçues en Ko, Mo/s, etc., en o, o/s. Sinon, Zabbix ne peut pas définir correctement les préfixes (K, M, G, etc.).
- Notez que si le type d'information de l'élément est *Numérique (non signé)*, les valeurs entrantes comportant une partie fractionnaire seront tronquées (c'est-à-dire que « 0.9 » deviendra « 0 ») avant l'application du multiplicateur personnalisé ;
- Si vous utilisez un multiplicateur personnalisé ou stockez la valeur en tant que *Variation par seconde* pour des éléments dont le type d'information est défini sur *Numérique (non signé)* et que la valeur calculée résultante est en réalité un nombre à virgule flottante, la valeur calculée est tout de même acceptée comme correcte : sa partie décimale est supprimée et la valeur est stockée sous forme d'entier ;
- Formats pris en charge : notation scientifique, par exemple `1e+70` ; macros utilisateur et macros LLD ; chaînes contenant des macros, par exemple `{#MACRO}e+10`, `{$MACRO1}e+{$MACRO2}`. Les macros doivent être résolues en un nombre entier ou à virgule flottante.

Modification simple

Calcule la différence entre la valeur actuelle et la valeur précédente.

Commentaires :

- Cette étape peut être utile pour mesurer une valeur qui augmente constamment ;
- Évaluée comme **value**-**prev_value**, où *value* correspond à la valeur actuelle et *prev_value* à la valeur reçue précédemment ;
- Une seule opération de modification par élément (« Modification simple » ou « Modification par seconde ») est autorisée ;
- Si la valeur actuelle est inférieure à la valeur précédente, Zabbix ignore cette différence (ne stocke rien) et attend une autre valeur.

Variation par seconde

Calcule la vitesse de variation de la valeur par seconde (différence entre la valeur actuelle et la valeur précédente).

Commentaires :

- Cette étape est utile pour calculer la vitesse par seconde d'une valeur qui augmente constamment ;
- Comme ce calcul peut produire des nombres à virgule flottante, il est recommandé de définir le « Type d'information » sur *Numérique (flottant)*, même si les valeurs brutes reçues sont des entiers. Cela est particulièrement pertinent pour les petits nombres dont la partie décimale est importante. Si les valeurs à virgule flottante sont grandes et peuvent dépasser la

longueur du champ « float », auquel cas la valeur entière risque d'être perdue, il est en fait recommandé d'utiliser *Numérique (non signé)* et de supprimer ainsi uniquement la partie décimale.
- Évalué comme (**value**-**prev_value**)/(**time**-**prev_time**), où *value* correspond à la valeur actuelle ; *prev_value* à la valeur reçue précédemment ; *time* à l'horodatage actuel ; *prev_time* à l'horodatage de la valeur précédente ;
- Une seule opération de variation par élément (« Variation simple » ou « Variation par seconde ») est autorisée ;
- Si la valeur actuelle est inférieure à la valeur précédente, Zabbix ignore cette différence (ne stocke rien) et attend une autre valeur. Cela permet notamment de fonctionner correctement avec le bouclage (dépassement) des compteurs SNMP 32 bits.

Booléen en décimal

Convertit la valeur du format booléen au format décimal.

Commentaires :

- La représentation textuelle est convertie en 0 ou 1. Ainsi, « TRUE » est stocké sous la forme 1 et « FALSE » sous la forme 0. Toutes les valeurs sont comparées sans tenir compte de la casse. Les valeurs actuellement reconnues sont, pour *TRUE* : true, t, yes, y, on, up, running, enabled, available, ok, master ; pour *FALSE* : false, f, no, n, off, down, unused, disabled, unavailable, err, slave. De plus, toute valeur numérique non nulle est considérée comme TRUE et zéro comme FALSE.

Octal en décimal

Convertit la valeur du format octal au format décimal.

Hexadécimal en décimal

Convertit la valeur du format hexadécimal au format décimal.

JavaScript

Saisissez le code JavaScript dans l'éditeur modal qui s'ouvre lorsque vous cliquez dans le champ du paramètre ou sur l'icône en forme de crayon à côté.

> **Attention:**
> N'utilisez pas d'affectations non déclarées dans le prétraitement JavaScript. Utilisez `var` pour déclarer les variables locales.

Commentaires :

- La longueur disponible pour le code JavaScript dépend de la base de données utilisée ;
- Pour plus d'informations, voir : Prétraitement JavaScript.

Dans la plage

Définissez une plage dans laquelle une valeur doit se trouver en spécifiant les valeurs minimale et maximale (incluses).

Commentaires :

- Les valeurs numériques sont acceptées (y compris un nombre quelconque de chiffres, une partie décimale facultative et une partie exponentielle facultative, ainsi que les valeurs négatives) ;
- La valeur minimale doit être inférieure à la valeur maximale ;
- Au moins une valeur doit être présente ;
- Les macros utilisateur et les macros de découverte de bas niveau peuvent être utilisées.

Correspond à l'expression régulière

Spécifiez une expression régulière à laquelle une valeur doit correspondre.

Ne correspond pas à l'expression régulière

Spécifiez une expression régulière à laquelle une valeur ne doit pas correspondre.

Vérifier la présence d'une erreur dans le JSON

Vérifier la présence d'un message d'erreur au niveau de l'application situé à l'emplacement JSONPath. Arrêter le traitement si l'opération réussit et que le message n'est pas vide ; sinon, continuer le traitement avec la valeur qui précédait cette étape de prétraitement.

Commentaires :

- Ces erreurs de service externe sont signalées à l'utilisateur telles quelles, sans ajouter d'informations sur l'étape de prétraitement ;
- Aucune erreur ne sera signalée en cas d'échec de l'analyse d'un JSON non valide.

Vérifier la présence d'une erreur dans le XML

Vérifie la présence d'un message d'erreur au niveau de l'application situé à l'emplacement XPath. Arrête le traitement si l'opération réussit et que le message n'est pas vide ; sinon, poursuit le traitement avec la valeur qui précédait cette étape de prétraitement.

Commentaires :

- Ces erreurs de service externe sont signalées à l'utilisateur telles quelles, sans ajouter d'informations sur l'étape de prétraitement ;
- Aucune erreur ne sera signalée en cas d'échec de l'analyse d'un XML non valide.

Vérifier une erreur à l'aide d'une expression régulière

Vérifier la présence d'un message d'erreur au niveau de l'application à l'aide d'une expression régulière. Arrêter le traitement si la vérification réussit et que le message n'est pas vide ; sinon, poursuivre le traitement avec la valeur qui précédait cette étape de prétraitement.

Paramètres :

- **pattern** - l'expression régulière ;
- **output** - le modèle de formatage de la sortie. Une séquence d'échappement \N (où N=1…9) est remplacée par le N^{e} groupe correspondant. Une séquence d'échappement \0 est remplacée par le texte correspondant.

Commentaires :

- Ces erreurs de service externe sont signalées à l'utilisateur telles quelles, sans ajouter d'informations sur l'étape de prétraitement.

Vérifier la valeur non prise en charge

Vérifiez si aucune valeur d'élément n'a pu être récupérée. Indiquez comment l'échec doit être traité, en fonction de l'inspection du message d'erreur renvoyé.

Paramètres :

- **scope** - sélectionnez la portée du traitement de l'erreur :
any error - toute erreur ;
error matches - uniquement l'erreur qui correspond à l'expression régulière spécifiée dans *pattern* ;
error does not match - uniquement l'erreur qui ne correspond pas à l'expression régulière spécifiée dans *pattern*
- **pattern** - l'expression régulière à utiliser pour rechercher une correspondance avec l'erreur. Si *any error* est sélectionné dans le paramètre de portée, ce champ n'est pas affiché. Lorsqu'il est affiché, ce champ est obligatoire.

Commentaires :

- Normalement, l'absence d'une valeur ou l'échec de sa récupération entraîne le passage de l'élément à l'état non pris en charge. Cette étape de prétraitement vous permet de modifier ce comportement.
- Cette étape de prétraitement vérifie uniquement si aucune valeur d'élément n'a pu être récupérée. Elle ne vérifie pas, par exemple, si le type de la valeur récupérée (par exemple, une chaîne) correspond au type d'information de l'élément (par exemple, numérique) ; pour plus de détails, consultez Exemples de prétraitement. En cas de non-correspondance des types, l'élément peut tout de même passer à l'état non pris en charge après l'exécution de toutes les étapes de prétraitement. Pour vérifier une non-correspondance des types, vous pouvez utiliser, par exemple, l'étape de prétraitement *Custom multiplier* ; consultez Exemples de prétraitement.
- La capture des groupes d'expressions régulières est prise en charge dans les champs *Set value to* ou *Set error to*. Utilisez \N (où N=1…9) pour récupérer le Nième groupe correspondant ; utilisez \0 pour récupérer le texte correspondant ;
- Ces étapes sont toujours exécutées en premier parmi les étapes de prétraitement et sont placées au-dessus de toutes les autres après l'enregistrement des modifications apportées à l'élément ;
- Plusieurs étapes *Check for not supported value* sont prises en charge, dans l'ordre spécifié. Une étape *any error* est automatiquement placée en dernière position dans ce groupe.

Ignorer les valeurs inchangées

Ignorer une valeur si elle n'a pas changé.

Commentaires :

- Si une valeur est ignorée, elle n'est pas enregistrée dans la base de données et le serveur Zabbix n'a aucune connaissance de la réception de cette valeur. Aucune expression de déclencheur ne sera évaluée ; par conséquent, aucun problème pour les déclencheurs associés ne sera créé/résolu. Les fonctions ne fonctionneront qu'en se basant sur les données réellement enregistrées dans la base de données. Comme les tendances sont construites à partir des données de la base de données, si aucune valeur n'est enregistrée pendant une heure, il n'y aura pas non plus de données de tendances pour cette heure.
- Une seule option de limitation peut être spécifiée par élément.

Ignorer les valeurs inchangées avec battement de cœur

Ignorer une valeur si elle n'a pas changé pendant la période définie (en secondes).

Commentaires :

- Les valeurs entières positives sont prises en charge pour spécifier le nombre de secondes (minimum : 1 seconde) ;
- Les suffixes de temps peuvent être utilisés (par exemple, 30s, 1m, 2h, 1d) ;
- Les macros utilisateur et les macros de découverte de bas niveau peuvent être utilisées ;
- Si une valeur est ignorée, elle n'est pas enregistrée dans la base de données et le serveur Zabbix n'a pas connaissance de la réception de cette valeur. Aucune expression de déclencheur ne sera évaluée ; par conséquent, aucun problème ne sera créé ou résolu pour les déclencheurs associés. Les fonctions fonctionneront uniquement à partir des données effectivement enregistrées dans la base de données. Comme les tendances sont générées à partir des données de la base de données, si aucune valeur n'est enregistrée pendant une heure, aucune donnée de tendance ne sera disponible pour cette heure.
- Une seule option de limitation peut être spécifiée par élément.

Modèle Prometheus

Utilisez la requête suivante pour extraire les données requises à partir des métriques Prometheus.

Consultez Vérifications Prometheus pour plus de détails.

Prometheus vers JSON

Convertissez les métriques Prometheus requises en JSON.

Voir Vérifications Prometheus pour plus de détails.

Prise en charge des macros

Les macros utilisateur et les macros utilisateur avec contexte sont prises en charge dans :

- les paramètres des étapes de prétraitement, y compris le code JavaScript ;
- les paramètres personnalisés de gestion des erreurs (champs *Définir la valeur sur* et *Définir l'erreur sur*).

Note:
Le contexte de la macro est ignoré lorsqu'une macro est remplacée par sa valeur. La valeur de la macro est insérée dans le code telle quelle ; il n'est pas possible d'ajouter un échappement supplémentaire avant de placer la valeur dans le code JavaScript. Veuillez noter que cela peut, dans certains cas, provoquer des erreurs JavaScript.

Test

Voir le test du prétraitement.

Test du prétraitement

Test

Le test des étapes de prétraitement est utile pour s'assurer que des pipelines de prétraitement complexes produisent les résultats attendus, sans attendre que la valeur de l'élément soit reçue et prétraitée.

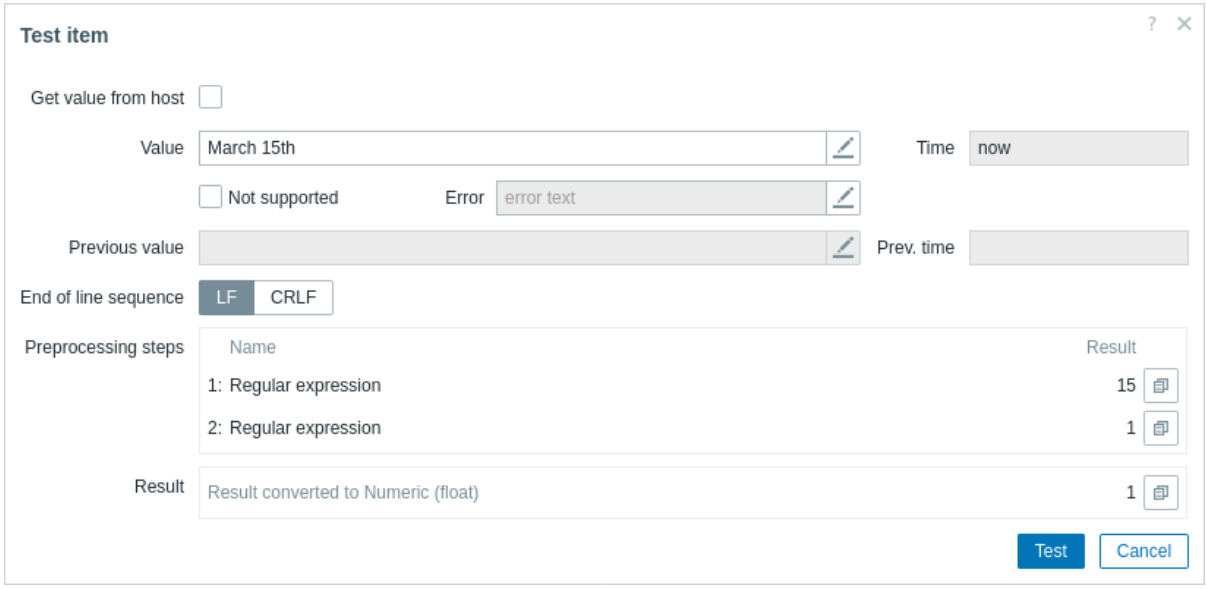


Il est possible de tester :

- avec une valeur hypothétique
- avec une valeur réelle provenant d'un hôte

Chaque étape de prétraitement peut être testée individuellement, et toutes les étapes peuvent également être testées ensemble. Lorsque vous cliquez sur le bouton *Test* ou *Tester toutes les étapes* respectivement dans le bloc Actions, une fenêtre de test s'ouvre.

Test de valeur hypothétique



Parameter	Description
Get value from host	Si vous souhaitez tester une valeur hypothétique, laissez cette case à cocher non sélectionnée. Voir aussi: Test de valeur réelle.
Value	Saisissez la valeur d'entrée à tester. Cliquer dans le champ du paramètre ou sur le bouton d'affichage/modification ouvrira une fenêtre de zone de texte pour saisir la valeur ou un bloc de code.
Not supported	Cochez cette case pour tester une valeur non prise en charge. Cette option est utile pour tester l'étape de prétraitement *Check for not supported value*.
Error	Saisissez le texte de l'erreur. Ce champ est activé lorsque *Get value from host* n'est pas coché, mais que *Not supported* est coché. Si *Get value from host* est coché, ce champ est rempli avec le message d'erreur réel (lecture seule) provenant de l'hôte.
Time	L'heure de la valeur d'entrée est affichée: `now` (lecture seule).
Previous value	Saisissez une valeur d'entrée précédente à comparer. Uniquement pour les étapes de prétraitement *Change* et *Throttling*.
Previous time	Saisissez l'heure de la valeur d'entrée précédente à comparer. Uniquement pour les étapes de prétraitement *Change* et *Throttling*. La valeur par défaut est basée sur la valeur du champ 'Update interval' de l'élément (si '1m', alors ce champ est rempli avec `now-1m`). Si rien n'est spécifié ou si l'utilisateur n'a pas accès à l'hôte, la valeur par défaut est `now-30s`.
Macros	Si des macros sont utilisées, elles sont listées avec leurs valeurs. Les valeurs sont modifiables à des fins de test, mais les modifications ne seront enregistrées que dans le contexte du test.
End of line sequence	Sélectionnez la séquence de fin de ligne pour les valeurs d'entrée multilignes: **LF** - séquence LF (line feed) **CRLF** - séquence CRLF (carriage-return line-feed).
Preprocessing steps	Les étapes de prétraitement sont listées; le résultat du test est affiché pour chaque étape après avoir cliqué sur le bouton *Test*. Les résultats de test sont tronqués à une taille maximale de 512KB lorsqu'ils sont envoyés à l'interface. Les résultats de test peuvent être copiés (sans dépasser les 512KB tronqués). Si un résultat est tronqué, une icône d'avertissement s'affiche. La description de l'avertissement s'affiche au survol de la souris. Notez que les données supérieures à 512KB sont toujours entièrement traitées par le serveur Zabbix. Si l'étape a échoué lors du test, une icône d'erreur s'affiche. La description de l'erreur s'affiche au survol de la souris. Dans le cas où "Custom on fail" est spécifié pour l'étape et que cette action est exécutée, une nouvelle ligne apparaît juste après la ligne de l'étape de test de prétraitement, indiquant quelle action a été effectuée et quel résultat elle a produit (erreur ou valeur).

Parameter	Description
Result	Le résultat final du test des étapes de prétraitement est affiché dans tous les cas lorsque toutes les étapes sont testées ensemble (lorsque vous cliquez sur le bouton *Test all steps*). Le type de conversion vers le type de valeur de l'élément est également affiché, par exemple `Result converted to Numeric (unsigned)`. Les résultats de test sont tronqués à une taille maximale de 512KB lorsqu'ils sont envoyés à l'interface. Le résultat final peut être copié (sans dépasser les 512KB tronqués). Si un résultat est tronqué, une icône d'avertissement s'affiche. La description de l'avertissement s'affiche au survol de la souris. Notez que les données supérieures à 512KB sont toujours entièrement traitées par le serveur Zabbix.

Cliquez sur *Test* pour voir le résultat après chaque étape de prétraitement.

Les valeurs de test sont stockées entre les sessions de test, que ce soit pour des étapes individuelles ou pour l'ensemble des étapes, ce qui permet à l'utilisateur de modifier les étapes de prétraitement ou la configuration de l'élément, puis de revenir à la fenêtre de test sans avoir à ressaisir les informations. Les valeurs sont toutefois perdues lors d'un rafraîchissement de la page.

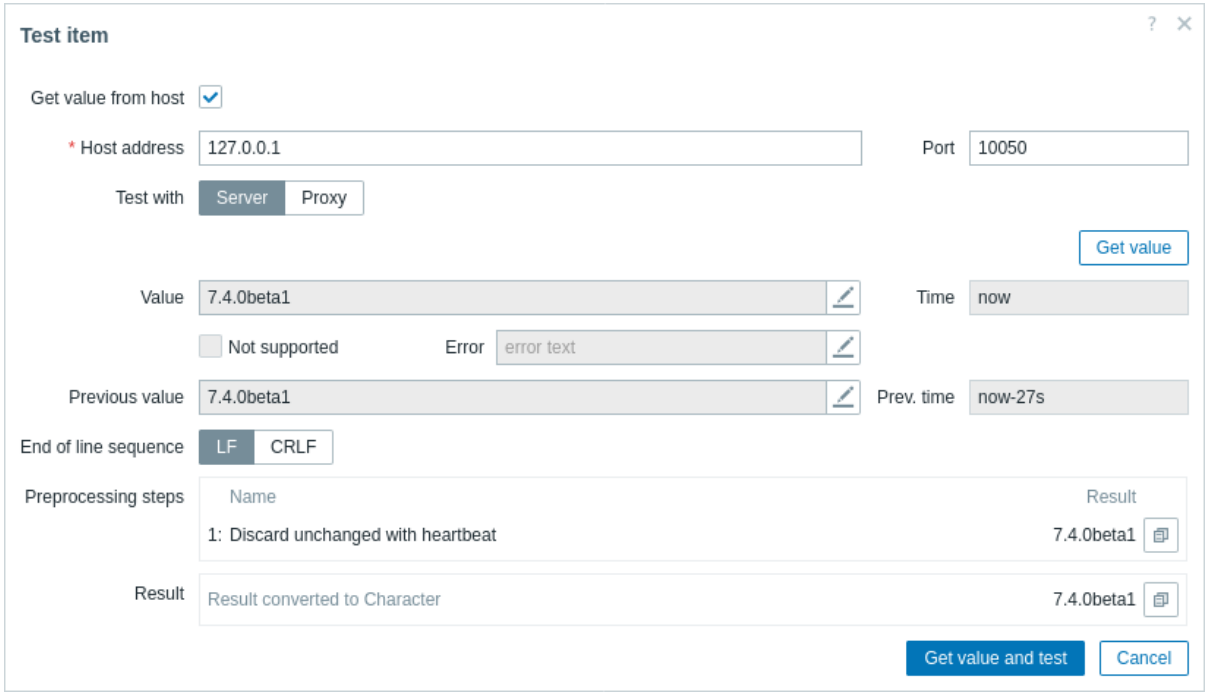
Le test est effectué par le serveur Zabbix. L'interface envoie une requête correspondante au serveur et attend le résultat. La requête contient la valeur d'entrée et les étapes de prétraitement (avec les macros utilisateur développées). Pour les étapes *Change* et *Throttling*, une valeur et une heure précédentes facultatives peuvent être spécifiées. Le serveur répond avec les résultats de chaque étape de prétraitement.

Toutes les erreurs techniques ou erreurs de validation des entrées sont affichées dans la zone d'erreur en haut de la fenêtre de test.

Tester une valeur réelle

Pour tester le prétraitement avec une valeur réelle :

- Cochez la case *Obtenir la valeur depuis l'hôte*
- Saisissez ou vérifiez les paramètres de l'hôte (adresse de l'hôte, port, nom du proxy ou absence de proxy) ainsi que les détails propres à l'élément (tels que la communauté SNMPv2 ou les identifiants de sécurité SNMPv3). Ces champs tiennent compte du contexte :
  - Les valeurs sont préremplies lorsque cela est possible, c'est-à-dire pour les éléments nécessitant un agent, en récupérant les informations depuis l'interface agent sélectionnée de l'hôte
  - Les valeurs doivent être saisies manuellement pour les éléments de modèle
  - Les valeurs des macros en texte brut sont résolues
  - Lorsque la valeur du champ (ou une partie de celle-ci) est un secret ou une macro Vault, le champ est vide et doit être rempli manuellement. Si un paramètre d'élément contient une valeur de macro secrète, le message d'avertissement suivant s'affiche : « L'élément contient des macros définies par l'utilisateur avec des valeurs secrètes. Les valeurs de ces macros doivent être saisies manuellement. »
  - Les champs sont désactivés lorsqu'ils ne sont pas nécessaires dans le contexte du type d'élément (par exemple, les champs d'adresse de l'hôte et de proxy sont désactivés pour les éléments calculés)
- Cliquez sur *Obtenir la valeur et tester* pour tester le prétraitement



Si vous avez spécifié une correspondance de valeurs dans le formulaire de configuration de l'élément (champ *Afficher la valeur*), la boîte de dialogue de test de l'élément affiche une ligne supplémentaire après le résultat final, nommée *Résultat avec la correspondance de valeurs appliquée*.

Paramètres spécifiques à l'obtention d'une valeur réelle depuis un hôte :

Paramètre	Description
Obtenir la valeur depuis l'hôte	Cochez cette case pour obtenir une valeur réelle depuis l'hôte.
Adresse de l'hôte	Saisissez l'adresse de l'hôte. Ce champ est automatiquement rempli avec l'adresse de l'interface de l'hôte de l'élément.
Port	Saisissez le port de l'hôte. Ce champ est automatiquement rempli avec le port de l'interface de l'hôte de l'élément.
Champs supplémentaires pour les interfaces SNMP (version SNMP, communauté SNMP, nom de contexte, etc.)	Consultez Configuration de la supervision SNMP pour plus d'informations sur la configuration d'une interface SNMP (v1, v2 et v3). Ces champs sont automatiquement remplis à partir de l'interface de l'hôte de l'élément.
Proxy	Indiquez le proxy si l'hôte est supervisé par un proxy. Ce champ est automatiquement rempli avec le proxy de l'hôte (le cas échéant).
Valeur	Valeur récupérée depuis l'hôte. Cliquer dans le champ du paramètre ou sur le bouton d'affichage/modification ouvre une fenêtre contenant une zone de texte avec la valeur ou le bloc de code. Les valeurs sont tronquées à une taille maximale de 512 Ko, uniquement dans l'interface. Si un résultat est tronqué, une icône d'avertissement s'affiche. La description de l'avertissement s'affiche au survol de la souris. Notez que les données de plus de 512 Ko sont tout de même entièrement traitées par le serveur Zabbix.

Pour les autres paramètres, consultez Tester une valeur hypothétique ci-dessus.

Détails du prétraitement

Présentation générale

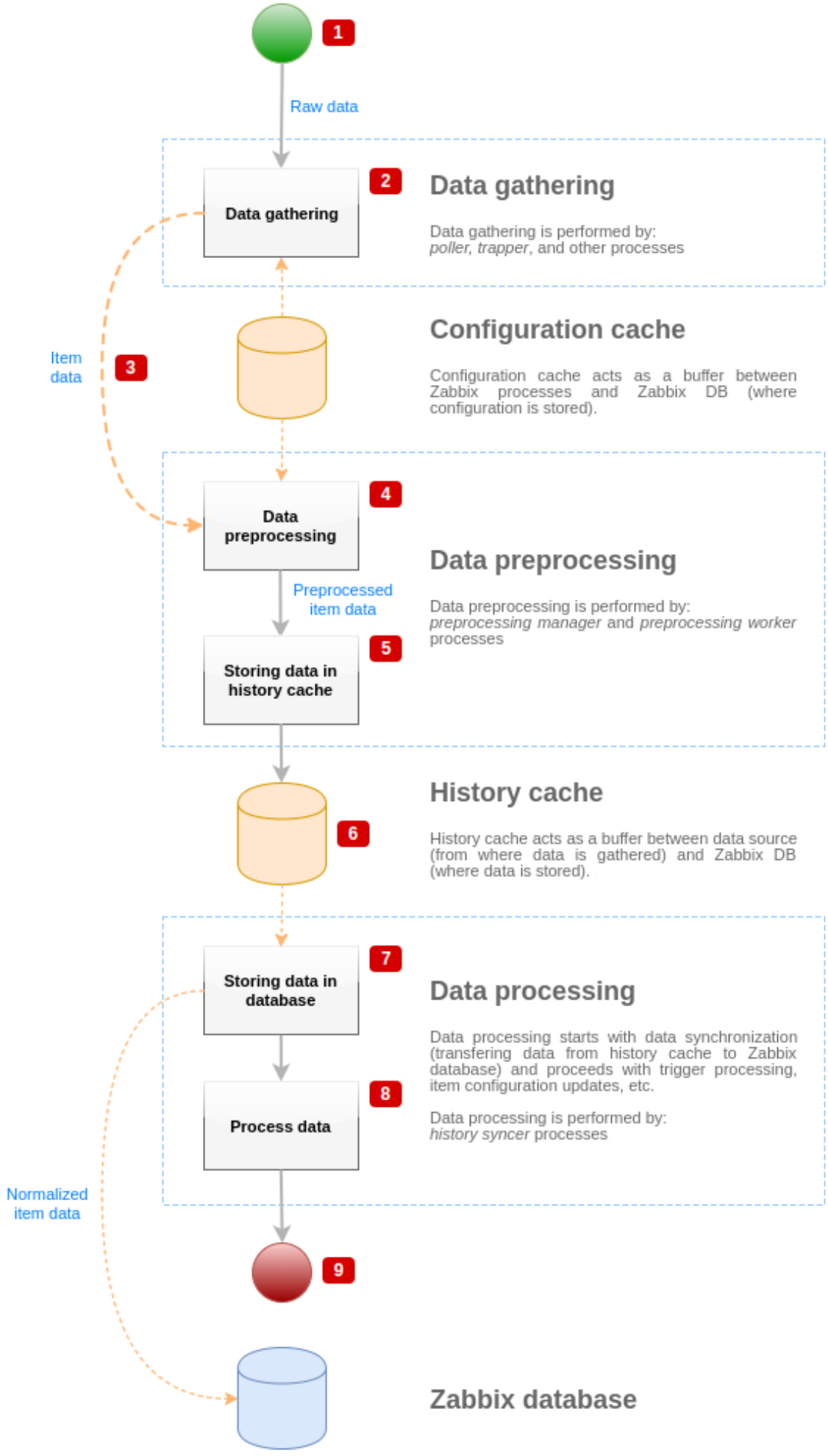
Cette section fournit des informations détaillées sur le prétraitement des valeurs des éléments. Le prétraitement des valeurs des éléments vous permet de définir et d'exécuter des règles de transformation pour les valeurs reçues des éléments.

Le prétraitement est géré par le processus de gestionnaire de prétraitement ainsi que par des workers de prétraitement qui exécutent les étapes de prétraitement. Toutes les valeurs soumises au prétraitement et reçues de différents collecteurs de données

passent par le gestionnaire de prétraitement avant d'être ajoutées au cache d'historique. Une communication IPC basée sur des sockets est utilisée entre les collecteurs de données (pollers, trappers, etc.) et le processus de prétraitement. Le serveur Zabbix ou le proxy Zabbix (pour les éléments surveillés par le proxy) exécute les étapes de prétraitement.

Traitement des valeurs d'élément

Pour visualiser le flux de données depuis la source de données jusqu'à la base de données Zabbix, nous pouvons utiliser le schéma simplifié suivant :


1
Raw data
Data gathering
2
Data gathering
Data gathering is performed by:
poller, trapper, and other processes
Item data
3
Configuration cache
Configuration cache acts as a buffer between Zabbix processes and Zabbix DB (where configuration is stored).
Data preprocessing
4
Data preprocessing
Data preprocessing is performed by:
preprocessing manager and preprocessing worker processes
Preprocessed item data
Storing data in history cache
5
History cache
6
History cache acts as a buffer between data source (from where data is gathered) and Zabbix DB (where data is stored).
Storing data in database
7
Data processing
Data processing starts with data synchronization (transfering data from history cache to Zabbix database) and proceeds with trigger processing, item configuration updates, etc.
Process data
8
Data processing is performed by:
history syncer processes
Normalized item data
9
Zabbix database

Le schéma ci-dessus montre uniquement les processus, objets et actions liés au traitement des valeurs d'élément sous une forme **simplifiée**. Le schéma ne montre pas les changements de direction conditionnels, la gestion des erreurs ni les boucles. Le cache local de données du gestionnaire de prétraitement n'est pas non plus affiché, car il n'affecte pas directement le flux de données. L'objectif de ce schéma est de montrer les processus impliqués dans le traitement des valeurs d'élément et la manière dont ils interagissent.

- La collecte des données commence par des données brutes provenant d'une source de données. À ce stade, les données ne contiennent que l'ID, l'horodatage et la valeur (il peut également s'agir de plusieurs valeurs).
- Quel que soit le type de collecteur de données utilisé, le principe reste le même pour les contrôles actifs ou passifs, pour les éléments trapper, etc., car cela ne modifie que le format des données et l'initiateur de la communication (soit le collecteur de données attend une connexion et des données, soit le collecteur de données initie la communication et demande les données). Les données brutes sont validées, la configuration de l'élément est récupérée depuis le cache de configuration (les données sont enrichies avec les données de configuration).
- Un mécanisme IPC basé sur des sockets est utilisé pour transmettre les données des collecteurs de données au gestionnaire de prétraitement. À ce stade, le collecteur de données continue à collecter des données sans attendre la réponse du gestionnaire de prétraitement.
- Le prétraitement des données est effectué. Cela inclut l'exécution des étapes de prétraitement et le traitement des éléments dépendants.

Note:
Un élément peut changer son état en NOT SUPPORTED pendant le prétraitement si l'une des étapes de prétraitement échoue.

- Les données d'historique du cache local de données du gestionnaire de prétraitement sont vidées dans le cache d'historique.
- À ce stade, le flux de données s'arrête jusqu'à la prochaine synchronisation du cache d'historique (lorsque le processus history syncer effectue la synchronisation des données).
- Le processus de synchronisation commence par la normalisation des données avant leur stockage dans la base de données Zabbix. La normalisation des données effectue des conversions vers le type d'élément souhaité (type défini dans la configuration de l'élément), y compris la troncature des données textuelles selon les tailles prédéfinies autorisées pour ces types (HISTORY_STR_VALUE_LEN pour string, HISTORY_TEXT_VALUE_LEN pour text et HISTORY_LOG_VALUE_LEN pour les valeurs log). Les données sont envoyées à la base de données Zabbix une fois la normalisation terminée.

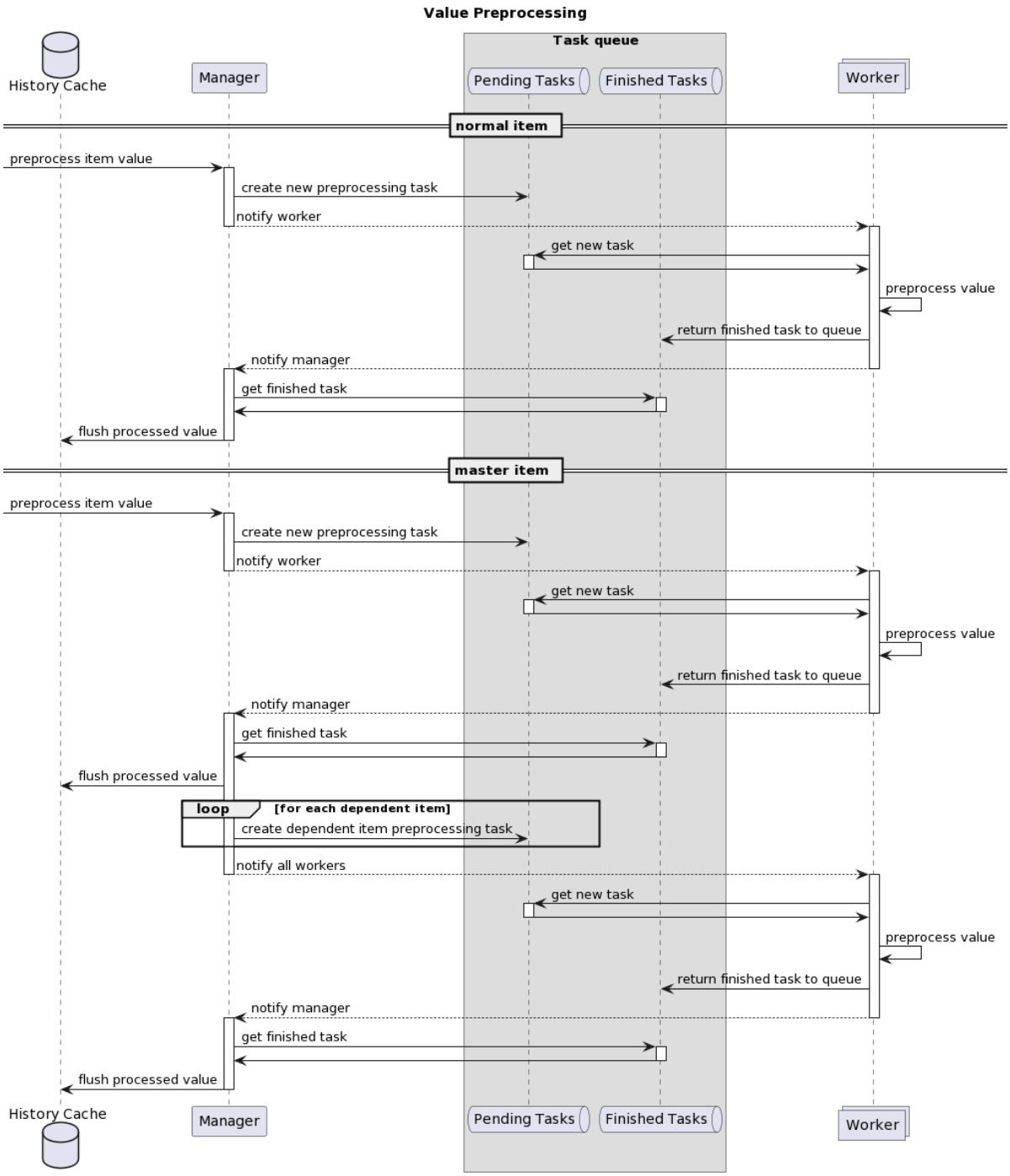
Note:
Un élément peut changer son état en NOT SUPPORTED si la normalisation des données échoue (par exemple, lorsqu'une valeur textuelle ne peut pas être convertie en nombre).

- Les données collectées sont traitées : les déclencheurs sont vérifiés, la configuration de l'élément est mise à jour si l'élément devient NOT SUPPORTED, etc.
- Ceci est considéré comme la fin du flux de données du point de vue du traitement des valeurs d'élément.

Prétraitement de la valeur de l'élément

Le prétraitement des données est effectué selon les étapes suivantes :

- Si l'élément n'a ni prétraitement ni éléments dépendants, sa valeur est soit ajoutée au cache d'historique, soit envoyée au gestionnaire LLD. Sinon, la valeur de l'élément est transmise au gestionnaire de prétraitement à l'aide d'un mécanisme IPC basé sur des sockets UNIX.
- Une tâche de prétraitement est créée, ajoutée à la file d'attente, et les workers de prétraitement sont informés de la nouvelle tâche.
- À ce stade, le flux de données s'arrête jusqu'à ce qu'il y ait au moins un worker de prétraitement inoccupé (c'est-à-dire n'exécutant aucune tâche).
- Lorsqu'un worker de prétraitement est disponible, il prend la tâche suivante dans la file d'attente.
- Une fois le prétraitement terminé (que l'exécution des étapes de prétraitement ait échoué ou réussi), la valeur prétraitée est ajoutée à la file d'attente des tâches terminées et le gestionnaire est informé qu'une nouvelle tâche est terminée.
- Le gestionnaire de prétraitement convertit le résultat au format souhaité (défini par le type de valeur de l'élément), puis l'ajoute soit au cache d'historique, soit l'envoie au gestionnaire LLD.
- S'il existe des éléments dépendants pour l'élément traité, ceux-ci sont alors ajoutés à la file d'attente de prétraitement avec la valeur prétraitée de l'élément maître. Les éléments dépendants sont mis en file d'attente en contournant les requêtes normales de prétraitement des valeurs, mais uniquement pour les éléments maîtres dont la valeur est définie et qui ne sont pas dans l'état NOT SUPPORTED.



Notez que, dans le diagramme, le prétraitement de l'élément maître est légèrement simplifié, car la mise en cache du prétraitement n'y est pas représentée.

File d'attente de prétraitement

La file d'attente de prétraitement est organisée comme suit :

- la liste des tâches en attente :
  - tâches créées directement à partir des demandes de prétraitement des valeurs, dans l'ordre de leur réception
- la liste des tâches immédiates (traitées avant les tâches en attente) :
  - tâches de test (créées en réponse aux demandes de test d'élément/de prétraitement par le frontend)
  - tâches des éléments dépendants
  - tâches de séquence (tâches devant être exécutées dans un ordre strict) :
    - comportant des étapes de prétraitement utilisant la dernière valeur :
      - changement
      - limitation du débit
      - JavaScript (mise en cache du bytecode)
    - mise en cache du prétraitement des éléments dépendants
- la liste des tâches terminées

Mise en cache du prétraitement

La mise en cache du prétraitement a été introduite pour améliorer les performances du prétraitement pour plusieurs éléments dépendants ayant des étapes de prétraitement similaires (ce qui est un résultat courant de la LLD).

La mise en cache consiste à prétraiter un élément dépendant et à réutiliser une partie des données internes de prétraitement pour le reste des éléments dépendants. Le cache de prétraitement n'est pris en charge que pour la première étape de prétraitement des types suivants :

- Motif Prometheus (indexe l'entrée par métriques)
- JSONPath (analyse les données en arbre d'objets et indexe la première expression `[?(@.path == "value")]`)

Workers de prétraitement

Le fichier de configuration du serveur Zabbix permet aux utilisateurs de définir le nombre de threads de workers de prétraitement. Le paramètre de configuration StartPreprocessors doit être utilisé pour définir le nombre d'instances de workers de prétraitement pré-démarrées, qui devrait au moins correspondre au nombre de cœurs CPU disponibles.

Si les tâches de prétraitement ne sont pas limitées par le CPU et impliquent des requêtes réseau fréquentes, il est recommandé de configurer des workers supplémentaires. Le nombre optimal de workers de prétraitement peut être déterminé par de nombreux facteurs, notamment le nombre d'éléments "prétraitables" (éléments qui nécessitent l'exécution d'étapes de prétraitement), le nombre de processus de collecte de données, le nombre moyen d'étapes de prétraitement par élément, etc. Un nombre insuffisant de workers peut entraîner une utilisation élevée de la mémoire. Pour le dépannage d'une utilisation excessive de la mémoire sur votre installation Zabbix, consultez Profiling excessive memory usage with tcmalloc.

Mais en supposant qu'il n'y ait pas d'opérations de prétraitement lourdes, comme l'analyse de gros blocs XML/JSON, le nombre de workers de prétraitement peut correspondre au nombre total de collecteurs de données. De cette façon, il y aura généralement (sauf dans les cas où les données du collecteur arrivent en masse) au moins un worker de prétraitement inoccupé pour les données collectées.

Warning:
Trop de processus de collecte de données (pollers, pollers injoignables, pollers ODBC, pollers HTTP, pollers Java, pingers, trappers, proxypollers) associés au gestionnaire IPMI, au trappeur SNMP et aux workers de prétraitement peuvent épuiser la limite de descripteurs de fichiers par processus pour le gestionnaire de prétraitement.

 L'épuisement de la limite de descripteurs de fichiers par processus entraînera l'arrêt du serveur Zabbix, généralement peu après le démarrage, mais parfois après un délai plus long. Pour éviter de tels problèmes, consultez le fichier de configuration du serveur Zabbix afin d'optimiser le nombre de vérifications et de processus simultanés. De plus, si nécessaire, assurez-vous que la limite de descripteurs de fichiers est suffisamment élevée en vérifiant et en ajustant les limites du système.

Pipeline de traitement des valeurs

Le traitement des valeurs des éléments est exécuté en plusieurs étapes (ou phases) par plusieurs processus. Cela peut entraîner :

- Un élément dépendant peut recevoir des valeurs, alors que la valeur maître, elle, ne le peut pas. Cela peut être obtenu en utilisant le cas d'usage suivant :
  - L'élément maître a pour type de valeur `UINT` (un élément trapper peut être utilisé), l'élément dépendant a pour type de valeur `TEXT`.
  - Aucune étape de prétraitement n'est requise ni pour l'élément maître ni pour l'élément dépendant.
  - Une valeur textuelle (par exemple, "abc") doit être transmise à l'élément maître.
  - Comme il n'y a aucune étape de prétraitement à exécuter, le gestionnaire de prétraitement vérifie que l'élément maître n'est pas dans l'état NOT SUPPORTED et qu'une valeur est définie (les deux conditions sont vraies), puis met en file d'attente l'élément dépendant avec la même valeur que l'élément maître (puisqu'il n'y a pas d'étapes de prétraitement).
  - Lorsque l'élément maître et l'élément dépendant atteignent tous deux la phase de synchronisation de l'historique, l'élément maître devient NOT SUPPORTED en raison d'une erreur de conversion de valeur (les données textuelles ne peuvent pas être converties en entier non signé).

Par conséquent, l'élément dépendant reçoit une valeur, tandis que l'élément maître change son état en NOT SUPPORTED.

- Un élément dépendant reçoit une valeur qui n'est pas présente dans l'historique de l'élément maître. Le cas d'usage est très similaire au précédent, à l'exception du type de l'élément maître. Par exemple, si le type `CHAR` est utilisé pour l'élément maître, alors la valeur de l'élément maître sera tronquée lors de la phase de synchronisation de l'historique, tandis que les éléments dépendants recevront leurs valeurs à partir de la valeur initiale (non tronquée) de l'élément maître.

Exemples de prétraitement

Aperçu

Cette section présente des exemples d'utilisation des étapes de prétraitement pour accomplir certaines tâches pratiques.

Filtrage des enregistrements du journal des événements VMware

Cet exemple utilise l'étape de prétraitement Correspond à l'expression régulière pour filtrer les événements inutiles du journal des événements VMware.

1. Sur un hôte VMware Hypervisor fonctionnel, vérifiez que l'élément vmware.eventlog est présent et fonctionne correctement. Notez que l'élément de journal des événements peut déjà être présent sur l'hyperviseur si un modèle VMware a été lié lors de la création de l'hôte.

2. Sur l'hôte VMware Hypervisor, créez un élément dépendant de type *Log* et définissez l'élément de journal des événements comme son élément maître.

3. Dans l'onglet *Prétraitement* de l'élément dépendant, cliquez sur *Ajouter* pour créer une étape de prétraitement et sélectionnez *Correspond à l'expression régulière* dans la liste déroulante. Ensuite, indiquez l'un des motifs suivants :

- Pour filtrer tous les événements du journal :

```
.* logged in .*
```

- Pour filtrer les lignes contenant des noms d'utilisateur après « User » :

```
\bUser\s+\K\S+
```

Attention:
Si l'expression régulière ne correspond pas, l'élément dépendant devient non pris en charge avec un message d'erreur correspondant. Pour éviter cela, cochez la case *Personnalisé en cas d'échec* et sélectionnez une option telle que l'abandon de la valeur ou la définition d'une valeur personnalisée. Veuillez noter que les valeurs ignorées ne sont pas stockées dans la base de données ; par conséquent, les déclencheurs ne sont pas évalués et les données de tendance ne sont pas générées.

Vous pouvez également utiliser l'étape de prétraitement Expression régulière pour extraire les groupes correspondants et contrôler la sortie :

- Pour extraire et afficher l'intégralité de l'événement du journal contenant « logged in », indiquez les paramètres suivants :

```
Pattern: .*logged in.*
Output: \0
```

- Pour extraire et afficher les noms d'utilisateur suivant « User » :

```
Pattern: User (.*?)(?=\ )
Output: \1
```

Vérification du type de la valeur récupérée

Cet exemple utilise l'étape de prétraitement Multiplicateur personnalisé pour vérifier si le type de la valeur de l'élément récupérée est numérique.

Dans l'onglet *Prétraitement* d'un élément, sélectionnez l'étape de prétraitement *Multiplicateur personnalisé* et indiquez le paramètre suivant (multiplie la valeur récupérée par 1) :

```
1
```

Attention:
Si le prétraitement échoue (par exemple, si l'entrée n'est pas numérique), l'élément devient non pris en charge avec un message d'erreur correspondant. Pour éviter cela, cochez la case *Personnalisé en cas d'échec* et sélectionnez une option telle que l'abandon de la valeur ou la définition d'une valeur personnalisée. Veuillez noter que les valeurs ignorées ne sont pas stockées dans la base de données ; par conséquent, les déclencheurs ne sont pas évalués et les données de tendance ne sont pas générées.

Vérification d'une valeur non prise en charge

Cet exemple utilise l'étape de prétraitement Check for not supported value pour vérifier si la valeur de l'élément n'a pas pu être récupérée.

Lorsqu'un processus poller du serveur/proxy Zabbix tente de collecter une valeur d'élément, il peut :

- Renvoyer un résultat valide.
- Renvoyer un résultat qui semble initialement valide, mais qui peut devenir non pris en charge plus tard (par exemple, en raison d'une incompatibilité de type de valeur après le prétraitement).

- Renvoyer une erreur lors de la collecte de la valeur, ce qui fait que l'élément devient non pris en charge. Les causes courantes incluent :
  - Clé d'élément inconnue (pour les éléments Zabbix agent, Simple check ou Zabbix internal)
  - OID inconnu (agent SNMP), capteur inconnu (agent IPMI) ou absence de métrique JMX (agent JMX)
  - Impossible de lire le fichier de trap (SNMP trap)
  - Script introuvable (External check)
  - URL inexistante (HTTP agent, Browser)
  - Échec de la connexion (SSH agent, TELNET agent)
  - Syntaxe de formule invalide (Calculated), erreur de syntaxe JavaScript (Script) ou SQL invalide (Database monitor)

Pour détecter et gérer les erreurs de collecte des valeurs d'élément, vous pouvez utiliser l'étape de prétraitement *Check for not supported value*. Notez que cette étape est toujours exécutée en premier et ne détecte que les erreurs qui se produisent avant le début du prétraitement.

Dans l'onglet *Preprocessing* d'un élément, sélectionnez l'étape de prétraitement *Check for not supported value* et spécifiez l'un des paramètres suivants :

- Pour toute erreur :

```
Parameter: any error
```

- Pour les erreurs contenant "cannot connect" :

```
Parameter: error matches
Pattern: (?i)cannot connect
```

Ensuite, utilisez l'option *Custom on fail* pour ignorer la valeur (dans ce cas, l'erreur), définir une valeur personnalisée ou renvoyer un message d'erreur personnalisé. Veuillez noter que les valeurs discarded ne sont pas stockées dans la base de données ; par conséquent, les déclencheurs ne sont pas évalués et les données de tendance ne sont pas générées.

Fonctionnalité JSONPath

Vue d'ensemble

Cette section présente les fonctionnalités JSONPath prises en charge dans les étapes de prétraitement des valeurs d'élément.

JSONPath est composé de segments séparés par des points. Un segment peut prendre la forme d'un mot simple, représentant un nom de valeur JSON, du caractère générique (*) ou d'une construction plus complexe placée entre crochets. Le point précédant un segment entre crochets est facultatif et peut être omis.

JSONPath example	Description
`$.object.name`	Renvoie le contenu de `object.name`.
`$.object['name']`	Renvoie le contenu de `object.name`.
`$.object.['name']`	Renvoie le contenu de `object.name`.
`$["object"]['name']`	Renvoie le contenu de `object.name`.
`$.['object'].["name"]`	Renvoie le contenu de `object.name`.
`$.object.history.length()`	Renvoie le nombre d'éléments du tableau `object.history`.
`$[?(@.name == 'Object')].price.first()`	Renvoie la valeur de la propriété `price` du premier objet nommé "Object".
`$[?(@.name == 'Object')].history.first().length()`	Renvoie le nombre d'éléments du tableau d'historique du premier objet nommé "Object".
`$[?(@.price > 10)].length()`	Renvoie le nombre d'objets dont le prix est supérieur à 10.

Voir aussi : Échappement des caractères spéciaux des valeurs de macro LLD dans JSONPath.

Segments pris en charge

Segment	Description
`<name>`	Correspond à une propriété d'objet par son nom.
`*`	Correspond à toutes les propriétés d'objet.
`['<name>']`	Correspond à une propriété d'objet par son nom.
`['<name>', '<name>', ...]`	Correspond à une propriété d'objet par l'un des noms listés.
`[<index>]`	Correspond à un élément de tableau par son index.

Segment	Description
`[<number>, <number>, ...]`	Correspond à un élément de tableau par l'un des index listés.
`[*]`	Correspond à toutes les propriétés d'objet ou à tous les éléments de tableau.
`[<start>:<end>]`	Correspond aux éléments de tableau selon la plage définie : **<start>** - le premier index à faire correspondre (inclus) ; s'il n'est pas spécifié, tous les éléments du tableau depuis le début correspondent ; s'il est négatif, il indique un décalage de départ à partir de la fin du tableau ; **<end>** - le dernier index à faire correspondre (exclu) ; s'il n'est pas spécifié, tous les éléments du tableau jusqu'à la fin correspondent ; s'il est négatif, il indique un décalage de départ à partir de la fin du tableau.
`[?(<expression>)]`	Correspond aux objets/éléments de tableau en appliquant une expression de filtre.

Pour trouver un segment correspondant sans tenir compte de son ascendance (segment détaché), il doit être préfixé par deux points (`..`). Par exemple, `$..name` ou `$..['name']` renvoient les valeurs de toutes les propriétés `name`.

Les noms des éléments correspondants peuvent être extraits en ajoutant un suffixe tilde (`~`) au JSONPath. Cela renvoie le nom de l'objet correspondant ou un index au format chaîne de l'élément de tableau correspondant. Le format de sortie suit les mêmes règles que les autres requêtes JSONPath : les résultats d'un chemin défini sont renvoyés « tels quels », et les résultats d'un chemin indéfini sont renvoyés dans un tableau. Cependant, il y a peu d'intérêt à extraire le nom d'un élément correspondant à un chemin défini, car il est déjà connu.

Expression de filtre

L'expression de filtre est une expression arithmétique en notation infixe.

Opérandes pris en charge :

Operand	Description

`"<text>"`
`'<text>'`
	Constante textuelle.
	
	Exemple :
	`'value: \'1\''`
	`"value: '1'"`

`<number>`	Constante numérique prenant en charge la notation scientifique. Exemple : 123
`<jsonpath starting with $>`	Valeur référencée par le JSONPath à partir du nœud racine du document d'entrée ; seuls les chemins déterministes sont pris en charge. Exemple : `$.object.name`
`<jsonpath starting with @>`	Valeur référencée par le JSONPath à partir de l'objet/élément courant ; seuls les chemins déterministes sont pris en charge. Exemple : `@.name`

Opérateurs pris en charge : |Operator|Type|Description|Result| |-|---|---|---| |−|Binaire|Soustraction|Nombre| |+|Binaire|Addition|Nombre| |/|Binaire|Division|Nombre| |*|Binaire|Multiplication|Nombre| |==|Binaire|Égalité|Booléen (1/0)| |!=|Binaire|Inégalité|Booléen (1/0)| |<|Binaire|Inférieur à|Booléen (1/0)| |<=|Binaire|Inférieur ou égal à|Booléen (1/0)| |>|Binaire|Supérieur à|Booléen (1/0)| |>=|Binaire|Supérieur ou égal à|Booléen (1/0)| |=~|Binaire|Correspond à une expression régulière|Booléen (1/0)| |!|Unaire|NON booléen|Booléen (1/0)| ||||Binaire|OU booléen|Booléen (1/0)| |&&|Binaire|ET booléen|Booléen (1/0)|

Fonctions

Les fonctions peuvent être utilisées à la fin d'un JSONPath. Plusieurs fonctions peuvent être chaînées si la fonction précédente renvoie une valeur acceptée par la fonction suivante.

Fonctions prises en charge :

Fonction	Description	Entrée	Sortie
`avg`	Valeur moyenne des nombres dans un tableau d'entrée	Tableau de nombres	Nombre

Fonction	Description	Entrée	Sortie
`min`	Valeur minimale des nombres dans un tableau d'entrée	Tableau de nombres	Nombre
`max`	Valeur maximale des nombres dans un tableau d'entrée	Tableau de nombres	Nombre
`sum`	Somme des nombres dans un tableau d'entrée	Tableau de nombres	Nombre
`length`	Nombre d'éléments dans un tableau d'entrée	Tableau	Nombre
`first`	Le premier élément d'un tableau	Tableau	Une construction JSON (objet, tableau, valeur) selon le contenu du tableau d'entrée

Les fonctions d'agrégation JSONPath acceptent des valeurs numériques entre guillemets. Ces valeurs sont automatiquement converties de chaînes en types numériques lorsqu'une agrégation est nécessaire. Une entrée incompatible entraînera la génération d'une erreur par la fonction.

Valeur de sortie

Les JSONPath peuvent être divisés en chemins définis et indéfinis. Un chemin défini ne peut renvoyer que null ou une seule correspondance. Un chemin indéfini peut renvoyer plusieurs correspondances : JSONPath avec des listes de noms/index détachées ou multiples, des tranches de tableau ou des segments d'expression. Cependant, lorsqu'une fonction est utilisée, le JSONPath devient défini, car les fonctions produisent toujours une seule valeur.

Un chemin défini renvoie l'objet/le tableau/la valeur auquel il fait référence. En revanche, un chemin indéfini renvoie un tableau des objets/tableaux/valeurs correspondants.

> **Attention:**
> L'ordre des propriétés dans les résultats de requête JSONPath peut ne pas correspondre à l'ordre d'origine des propriétés JSON en raison des méthodes d'optimisation internes. Par exemple, le JSONPath `$.books[1]["author", "title"]` peut renvoyer `["title", "author"]`. S'il est essentiel de préserver l'ordre d'origine des propriétés, il convient d'envisager d'autres méthodes de post-traitement après la requête.

Règles de formatage des chemins

Les espaces (espace, caractère de tabulation) peuvent être utilisés dans les segments et expressions en notation entre crochets, par exemple : `$[ 'a' ][ 0 ][ ?( $.b == 'c' ) ][ : -1 ].first( )`.

Les chaînes doivent être entourées de guillemets simples (`'`) ou doubles (`"`). À l'intérieur des chaînes, les guillemets simples ou doubles (selon ceux utilisés pour les entourer) ainsi que les barres obliques inverses (`\`) sont échappés avec le caractère barre oblique inverse (`\`).

Exemple

```
{
  "books": [
    {
      "category": "reference",
      "author": "Nigel Rees",
      "title": "Sayings of the Century",
      "price": 8.95,
      "id": 1
    },
    {
      "category": "fiction",
      "author": "Evelyn Waugh",
      "title": "Sword of Honour",
      "price": 12.99,
      "id": 2
    },
    {
      "category": "fiction",
      "author": "Herman Melville",
      "title": "Moby Dick",
      "isbn": "0-553-21311-3",
      "price": 8.99,
```

```
      "id": 3
    },
    {
      "category": "fiction",
      "author": "J. R. R. Tolkien",
      "title": "The Lord of the Rings",
      "isbn": "0-395-19395-8",
      "price": 22.99,
      "id": 4
    }
  ],
  "services": {
    "delivery": {
      "servicegroup": 1000,
      "description": "Livraison le lendemain dans la ville locale",
      "active": true,
      "price": 5
    },
    "bookbinding": {
      "servicegroup": 1001,
      "description": "Impression et assemblage du livre au format A5",
      "active": true,
      "price": 154.99
    },
    "restoration": {
      "servicegroup": 1002,
      "description": "Diverses méthodes de restauration",
      "active": false,
      "methods": [
        {
          "description": "Nettoyage chimique",
          "price": 46
        },
        {
          "description": "Pressage des pages endommagées par l'humidité",
          "price": 24.5
        },
        {
          "description": "Rebrochage d'un livre déchiré",
          "price": 99.49
        }
      ]
    }
  },
  "filters": {
    "price": 10,
    "category": "fiction",
    "no filters": "no \"filters\""
  },
  "closed message": "Le magasin est fermé",
  "tags": [
    "a",
    "b",
    "c",
    "d",
    "e"
  ]
}
```


JSONPath	Type	Result
`$.filters.price`	definite	10
`$.filters.category`	definite	fiction

JSONPath	Type	Result
`$.filters['no filters']`	definite	no ”filters”
`$.filters`	definite	{ ”price”: 10, ”category”: ”fiction”, ”no filters”: ”no \”filters\”” }
`$.books[1].title`	definite	Sword of Honour
`$.books[-1].author`	definite	J. R. R. Tolkien
`$.books.length()`	definite	4
`$.tags[:]`	indefinite	[”a”, ”b”, ”c”, ”d”, ”e”]
`$.tags[2:]`	indefinite	[”c”, ”d”, ”e”]
`$.tags[:3]`	indefinite	[”a”, ”b”, ”c”]
`$.tags[1:4]`	indefinite	[”b”, ”c”, ”d”]
`$.tags[-2:]`	indefinite	[”d”, ”e”]
`$.tags[:-3]`	indefinite	[”a”, ”b”]
`$.tags[:-3].length()`	definite	2
`$.books[0, 2].title`	indefinite	[”Moby Dick”, ”Sayings of the Century”]
`$.books[1]['author', "title"]`	indefinite	[”Sword of Honour”, ”Evelyn Waugh”]
`$..id`	indefinite	[1, 2, 3, 4]
`$.services..price`	indefinite	[154.99, 5, 46, 24.5, 99.49]
`$.books[?(@.id == 4 - 0.4 * 5)].title`	indefinite	[”Sword of Honour”] Note: Cette requête montre que des opérations arithmétiques peuvent être utilisées dans les requêtes ; elle peut être simplifiée en `$.books[?(@.id == 2)].title`
`$.books[?(@.id == 2 || @.id == 4)].title`	indefinite	[”Sword of Honour”, ”The Lord of the Rings”]
`$.books[?(!(@.id == 2))].title`	indefinite	[”Sayings of the Century”, ”Moby Dick”, ”The Lord of the Rings”]
`$.books[?(@.id != 2)].title`	indefinite	[”Sayings of the Century”, ”Moby Dick”, ”The Lord of the Rings”]
`$.books[?(@.title =~ " of ")].title`	indefinite	[”Sayings of the Century”, ”Sword of Honour”, ”The Lord of the Rings”]
`$.books[?(@.price > 12.99)].title`	indefinite	[”The Lord of the Rings”]
`$.books[?(@.author > "Herman Melville")].title`	indefinite	[”Sayings of the Century”, ”The Lord of the Rings”]
`$.books[?(@.price > $.filters.price)].title`	indefinite	[”Sword of Honour”, ”The Lord of the Rings”]
`$.books[?(@.category == $.filters.category)].title`	indefinite	[”Sword of Honour”,”Moby Dick”,”The Lord of the Rings”]
`$.books[?(@.category == "fiction" && @.price < 10)].title`	indefinite	[”Moby Dick”]

JSONPath	Type	Result

`$..[?(@.id)]`	indefinite	[
		{
		”price”: 8.95,
		”id”: 1,
		”category”: ”reference”,
		”author”: ”Nigel Rees”,
		”title”: ”Sayings of the Century”
		},
		{
		”price”: 12.99,
		”id”: 2,
		”category”: ”fiction”,
		”author”: ”Evelyn Waugh”,
		”title”: ”Sword of Honour”
		},
		{
		”price”: 8.99,
		”id”: 3,
		”category”: ”fiction”,
		”author”: ”Herman Melville”,
		”title”: ”Moby Dick”,
		”isbn”: ”0-553-21311-3”
		},
		{
		”price”: 22.99,
		”id”: 4,
		”category”: ”fiction”,
		”author”: ”J. R. R. Tolkien”,
		”title”: ”The Lord of the Rings”,
		”isbn”: ”0-395-19395-8”
		}
		]

`$.services..[?(@.price > 50)].description`	indefinite	[”Impression et assemblage du livre au format A5”, ”Rebrochage d’un livre déchiré”]
`$..id.length()`	definite	4
`$.books[?(@.id == 2)].title.first()`	definite	Sword of Honour
`$..tags.first().length()`	definite	5 Note: `$..tags` est un chemin indéfini, il renvoie donc un tableau des éléments correspondants, c’est-à-dire `[["a", "b", "c", "d", "e" ]]` ; `first()` renvoie le premier élément, c’est-à-dire `["a", "b", "c", "d", "e"]` ; `length()` calcule la longueur de l’élément, c’est-à-dire 5.
`$.books[*].price.min()`	definite	8.95
`$..price.max()`	definite	154.99
`$.books[?(@.category == "fiction")].price.avg()`	definite	14.99
`$.books[?(@.category == $.filters.xyz)].title`	indefinite	Note: Une requête sans correspondance renvoie NULL pour les chemins définis et indéfinis.
`$.services[?(@.active=="true")].servicegroup`	indefinite	[1001,1000] Note: Les constantes textuelles doivent être utilisées dans les comparaisons de valeurs booléennes.
`$.services[?(@.active=="false")].servicegroup`	indefinite	[1002] Note: Les constantes textuelles doivent être utilisées dans les comparaisons de valeurs booléennes.
`$.services[?(@.servicegroup==1002)].description`	definite	test

Échappement des caractères spéciaux des valeurs de macro LLD dans JSONPath

Lorsque des macros de découverte de bas niveau sont utilisées dans le prétraitement JSONPath et que leurs valeurs sont résolues, les règles d'échappement des caractères spéciaux suivantes s'appliquent :

- seul le caractère antislash (\) et le caractère guillemet double (") sont pris en compte pour l'échappement ;
- si la valeur de macro résolue contient ces caractères, chacun d'eux est échappé avec un antislash ;
- s'ils sont déjà échappés avec un antislash, cela n'est pas considéré comme un échappement et l'antislash ainsi que les caractères spéciaux suivants sont échappés à nouveau.

Par exemple :

JSONPath	Valeur de macro LLD	Après substitution
$.[?(@.value == "{#MACRO}")]	special "value"	$.[?(@.value == "special \"value\"")]
	c:\temp	$.[?(@.value == "c:\\temp")]
	a\\b	$.[?(@.value == "a\\\\b")]

Lorsqu'elle est utilisée dans l'expression, la macro qui peut contenir des caractères spéciaux doit être placée entre guillemets doubles :

JSONPath	Valeur de macro LLD	Après substitution	Résultat
$.[?(@.value == "{#MACRO}")]	special "value"	$.[?(@.value == "special \"value\"")]	OK
$.[?(@.value == {#MACRO})]		$.[?(@.value == special \"value\")]	**Expression JSONPath incorrecte**

Lorsqu'elle est utilisée dans le chemin, la macro qui peut contenir des caractères spéciaux doit être placée entre crochets **et** entre guillemets doubles :

JSONPath	Valeur de macro LLD	Après substitution	Résultat
$.["{#MACRO}"].value	c:\temp	$.["c:\\temp"].value	OK
$.{#MACRO}.value		$.c:\\temp.value	**Expression JSONPath incorrecte**

Prétraitement JavaScript

Vue d'ensemble

Cette section fournit des détails sur le prétraitement par JavaScript.

> **Attention:**
> N'utilisez pas d'affectations non déclarées dans le JavaScript de prétraitement. Utilisez `var` pour déclarer les variables locales.

Prétraitement JavaScript

Le prétraitement JavaScript est effectué en appelant une fonction JavaScript avec un seul paramètre 'value' et un corps de fonction fourni par l'utilisateur. Le résultat de l'étape de prétraitement est la valeur renvoyée par cette fonction. Par exemple, pour effectuer une conversion de Fahrenheit en Celsius, saisissez :

```
return (value - 32)  * 5 / 9
```

dans les paramètres de prétraitement JavaScript, qui seront encapsulés dans une fonction JavaScript par le serveur :

```
function (value)
{
   return (value - 32) * 5 / 9
}
```

Le paramètre d'entrée 'value' est toujours transmis sous forme de chaîne. La valeur de retour est automatiquement convertie en chaîne via la méthode toString() (si cela échoue, l'erreur est renvoyée sous forme de valeur de chaîne), avec quelques exceptions :

- le renvoi de la valeur undefined entraîne une erreur ;
- le renvoi de la valeur null provoque l'abandon de la valeur d'entrée, un peu comme le prétraitement 'Discard value' dans l'action 'Custom on fail'.

Les erreurs peuvent être renvoyées en lançant des valeurs/objets (généralement des chaînes ou des objets Error).

Par exemple :

```
if (value == 0)
    throw "Zero input value"
return 1/value
```

Chaque script dispose d'un délai d'exécution de 10 secondes (selon le script, le déclenchement du délai peut prendre plus de temps) ; tout dépassement renverra une erreur. Une limite de tas de 512 mégaoctets est appliquée.

Le bytecode de l'étape de prétraitement JavaScript est mis en cache et réutilisé lorsque l'étape est appliquée à nouveau. Toute modification des étapes de prétraitement de l'élément entraînera la réinitialisation du script mis en cache, qui sera recompilé ultérieurement.

Des échecs d'exécution consécutifs (3 d'affilée) provoqueront la réinitialisation du moteur afin de réduire le risque qu'un script perturbe l'environnement d'exécution des scripts suivants (cette action est consignée avec DebugLevel 4 et supérieur).

Le prétraitement JavaScript est implémenté avec le moteur JavaScript Duktape.

> **Note:**
> Les définitions des méthodes d'objets JavaScript intégrées ne peuvent pas être redéfinies dans les scripts de prétraitement d'éléments.

Voir aussi : Additional JavaScript objects and global functions

Utilisation des macros dans les scripts

Il est possible d'utiliser des macros utilisateur (ainsi que des macros LLD dans le contexte de la découverte de bas niveau) dans le code JavaScript. Si un script contient des macros utilisateur, ces macros sont résolues par le serveur/proxy avant l'exécution des étapes de prétraitement spécifiques. Notez que lors du test des étapes de prétraitement dans le frontend, les valeurs des macros ne seront pas récupérées et devront être saisies manuellement.

> **Note:**
> Le contexte est ignoré lorsqu'une macro est remplacée par sa valeur. La valeur de la macro est insérée telle quelle dans le code, il n'est pas possible d'ajouter un échappement supplémentaire avant de placer la valeur dans le code JavaScript. Veuillez noter que cela peut provoquer des erreurs JavaScript dans certains cas.

Dans l'exemple ci-dessous, si la valeur reçue dépasse la valeur de la macro {$THRESHOLD}, la valeur du seuil (si elle est présente) sera renvoyée à la place :

```
var threshold = '{$THRESHOLD}';
return (!isNaN(threshold) && value > threshold) ? threshold : value;
```

Exemples

Les exemples suivants illustrent comment vous pouvez utiliser le prétraitement JavaScript.

Chaque exemple contient une brève description, un corps de fonction pour les paramètres de prétraitement JavaScript, ainsi que le résultat de l'étape de prétraitement : la valeur renvoyée par la fonction.

Exemple 1 : Convertir un nombre (notation scientifique en entier)

Convertissez le nombre « 2.62128e+07 » de la notation scientifique en entier.

```
return (Number(value))
```

Valeur renvoyée par la fonction : 26212800.

Exemple 2 : Convertir un nombre (binaire en décimal)

Convertissez le nombre binaire « 11010010 » en nombre décimal.

```
return(parseInt(value,2))
```

Valeur renvoyée par la fonction : 210.

Exemple 3 : Arrondir un nombre

Arrondissez le nombre « 18.2345 » à 2 chiffres.

```
return(Math.round(value* 100) / 100)
```

Valeur renvoyée par la fonction : 18.23.

Exemple 4 : Compter les lettres dans une chaîne

Comptez le nombre de lettres dans la chaîne « Zabbix ».

```
return (value.length)
```

Valeur renvoyée par la fonction : 6.

Exemple 5 : Obtenir le temps restant

Obtenez le temps restant (en secondes) jusqu'à la date d'expiration d'un certificat (12 févr. 12:33:56 2022 GMT).

```
var split = value.split(' '),
    MONTHS_LIST = ['Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun', 'Jul', 'Aug', 'Sep', 'Oct', 'Nov', 'Dec'],
    month_index = ('0' + (MONTHS_LIST.indexOf(split[0]) + 1)).slice(-2),
    ISOdate = split[3] + '-' + month_index + '-' + split[1] + 'T' + split[2],
    now = Date.now();

return parseInt((Date.parse(ISOdate) - now) / 1000);
```

Valeur renvoyée par la fonction : 44380233.

Exemple 6 : Supprimer des propriétés JSON

Modifiez la structure des données JSON en supprimant toutes les propriétés dont la clé est "data_size" ou "index_size".

```
var obj=JSON.parse(value);

for (i = 0; i < Object.keys(obj).length; i++) {
    delete obj[i]["data_size"];
    delete obj[i]["index_size"];
}

return JSON.stringify(obj)
```

Valeur acceptée par la fonction :

```
[
    {
        "table_name":"history",
        "data_size":"326.05",
        "index_size":"174.34"
    },
    {
        "table_name":"history_log",
        "data_size":"6.02",
        "index_size":"3.45"
    }
]
```

Valeur renvoyée par la fonction :

```
[
    {
        "table_name":"history"
    },
    {
        "table_name":"history_log"
    }
]
```

Exemple 7 : Convertir le statut Apache en JSON

Convertissez la valeur reçue d'un élément agent Zabbix web.page.get (par exemple, web.page.get[http://127.0.0.1:80/server-status?auto]) en objet JSON.

```
// Convertir le statut Apache en JSON

// Diviser la valeur en sous-chaînes et placer ces sous-chaînes dans un tableau
var lines = value.split('\n');

// Créer un objet vide "output"
var output = {};

// Créer un objet "workers" avec des propriétés prédéfinies
var workers = {
    '_': 0, 'S': 0, 'R': 0, 'W': 0,
    'K': 0, 'D': 0, 'C': 0, 'L': 0,
    'G': 0, 'I': 0, '.': 0
};

// Ajouter les sous-chaînes du tableau "lines" à l'objet "output" en tant que propriétés (paires clé-valeu
for (var i = 0; i < lines.length; i++) {
    var line = lines[i].match(/([A-z0-9 ]+): (.*)/);

    if (line !== null) {
        output[line[1]] = isNaN(line[2]) ? line[2] : Number(line[2]);
    }
}

// Métriques multiversion
output.ServerUptimeSeconds = output.ServerUptimeSeconds || output.Uptime;
output.ServerVersion = output.ServerVersion || output.Server;

// Analyser la propriété "Scoreboard" pour obtenir le nombre de workers
if (typeof output.Scoreboard === 'string') {
    for (var i = 0; i < output.Scoreboard.length; i++) {
        var char = output.Scoreboard[i];

        workers[char]++;
    }
}

// Ajouter les données des workers à l'objet "output"
output.Workers = {
    waiting: workers['_'], starting: workers['S'], reading: workers['R'],
    sending: workers['W'], keepalive: workers['K'], dnslookup: workers['D'],
    closing: workers['C'], logging: workers['L'], finishing: workers['G'],
    cleanup: workers['I'], slot: workers['.']
};

// Renvoyer la chaîne JSON
return JSON.stringify(output);
```

Valeur acceptée par la fonction :

```
HTTP/1.1 200 OK
Date: Mon, 27 Mar 2023 11:08:39 GMT
Server: Apache/2.4.52 (Ubuntu)
Vary: Accept-Encoding
Content-Encoding: gzip
Content-Length: 405
Content-Type: text/plain; charset=ISO-8859-1

127.0.0.1
ServerVersion: Apache/2.4.52 (Ubuntu)
ServerMPM: prefork
Server Built: 2023-03-08T17:32:01
CurrentTime: Monday, 27-Mar-2023 14:08:39 EEST
```

```
RestartTime: Monday, 27-Mar-2023 12:19:59 EEST
ParentServerConfigGeneration: 1
ParentServerMPMGeneration: 0
ServerUptimeSeconds: 6520
ServerUptime: 1 hour 48 minutes 40 seconds
Load1: 0.56
Load5: 0.33
Load15: 0.28
Total Accesses: 2476
Total kBytes: 8370
Total Duration: 52718
CPUUser: 8.16
CPUSystem: 3.44
CPUChildrenUser: 0
CPUChildrenSystem: 0
CPULoad: .177914
Uptime: 6520
ReqPerSec: .379755
BytesPerSec: 3461.58
BytesPerReq: 3461.58
DurationPerReq: 21.2916
BusyWorkers: 2
IdleWorkers: 6
Scoreboard: ____KW__.....................................................................................
```

Valeur renvoyée par la fonction :


```
{
    "Date": "Mon, 27 Mar 2023 11:08:39 GMT",
    "Server": "Apache/2.4.52 (Ubuntu)",
    "Vary": "Accept-Encoding",
    "Encoding": "gzip",
    "Length": 405,
    "Type": "text/plain; charset=ISO-8859-1",
    "ServerVersion": "Apache/2.4.52 (Ubuntu)",
    "ServerMPM": "prefork",
    "Server Built": "2023-03-08T17:32:01",
    "CurrentTime": "Monday, 27-Mar-2023 14:08:39 EEST",
    "RestartTime": "Monday, 27-Mar-2023 12:19:59 EEST",
    "ParentServerConfigGeneration": 1,
    "ParentServerMPMGeneration": 0,
    "ServerUptimeSeconds": 6520,
    "ServerUptime": "1 hour 48 minutes 40 seconds",
    "Load1": 0.56,
    "Load5": 0.33,
    "Load15": 0.28,
    "Total Accesses": 2476,
    "Total kBytes": 8370,
    "Total Duration": 52718,
    "CPUUser": 8.16,
    "CPUSystem": 3.44,
    "CPUChildrenUser": 0,
    "CPUChildrenSystem": 0,
    "CPULoad": 0.177914,
    "Uptime": 6520,
    "ReqPerSec": 0.379755,
    "BytesPerSec": 1314.55,
    "BytesPerReq": 3461.58,
    "DurationPerReq": 21.2916,
    "BusyWorkers": 2,
    "IdleWorkers": 6,
    "Scoreboard": "____KW__.....................................................................................
    "Workers": {
```

```
        "waiting": 6,
        "starting": 0,
        "reading": 0,
        "sending": 1,
        "keepalive": 1,
        "dnslookup": 0,
        "closing": 0,
        "logging": 0,
        "finishing": 0,
        "cleanup": 0,
        "slot": 142
    }
}
```

Objets JavaScript supplémentaires

Vue d'ensemble

Cette section décrit les ajouts de Zabbix au langage JavaScript implémenté avec Duktape, ainsi que les fonctions JavaScript globales prises en charge.

Attention:
N'utilisez pas d'affectations à des variables non déclarées dans le JavaScript de prétraitement. Utilisez `var` pour déclarer les variables locales.

Objets intégrés

Zabbix

L'objet Zabbix permet d'interagir avec les fonctionnalités internes de Zabbix.

Méthode	Description
`log(loglevel, message)`	Écrit <message> dans le journal Zabbix en utilisant le niveau de journalisation <loglevel> (voir le paramètre DebugLevel du fichier de configuration).

Exemple :

```
Zabbix.log(3, "this is a log entry written with 'Warning' log level")
```

Vous pouvez utiliser les alias suivants :

Alias	Alias de
console.log(object)	Zabbix.log(4, JSON.stringify(object))
console.warn(object)	Zabbix.log(3, JSON.stringify(object))
console.error(object)	Zabbix.log(2, JSON.stringify(object))

Attention:
La taille totale de tous les messages consignés est limitée à 8 Mo par exécution de script.

Méthode	Description
`sleep(delay)`	Retarde l'exécution de JavaScript de `delay` millisecondes.

Exemple (retarder l'exécution de 15 secondes) :

```
Zabbix.sleep(15000)
```

HttpRequest

Cet objet encapsule un handle cURL et permet d'effectuer simplement des requêtes HTTP. Les erreurs sont générées sous forme d'exceptions.

> **Attention:**
> L'initialisation de plusieurs objets `HttpRequest` est limitée à 10 par exécution du script.

Method	Description
`addHeader(value)`	Ajoute un champ d'en-tête HTTP. Ce champ est utilisé pour toutes les requêtes suivantes jusqu'à sa suppression avec la méthode `clearHeader()`. La longueur totale des champs d'en-tête pouvant être ajoutés à un même objet `HttpRequest` est limitée à 128 Ko (caractères spéciaux et noms d'en-tête inclus).
`clearHeader()`	Efface l'en-tête HTTP. Si aucun champ d'en-tête n'est défini, `HttpRequest` définit Content-Type sur application/json si les données envoyées sont au format JSON, et sur text/plain dans le cas contraire.
`connect(url)`	Envoie une requête HTTP CONNECT à l'URL et renvoie la réponse.
`customRequest(method, url, data)`	Permet de spécifier n'importe quelle méthode HTTP dans le premier paramètre. Envoie la requête correspondant à la méthode à l'URL avec une charge utile *data* facultative et renvoie la réponse.
`delete(url, data)`	Envoie une requête HTTP DELETE à l'URL avec une charge utile *data* facultative et renvoie la réponse.
`getHeaders(<asArray>)`	Renvoie l'objet contenant les champs d'en-tête HTTP reçus. Le paramètre `asArray` peut être défini sur "true" (par exemple, `getHeaders(true)`), "false" ou être indéfini. S'il est défini sur "true", les valeurs des champs d'en-tête reçus sont renvoyées sous forme de tableaux ; cette option doit être utilisée pour récupérer les valeurs de plusieurs en-têtes portant le même nom. S'il n'est pas défini ou s'il est défini sur "false", les valeurs des champs d'en-tête HTTP reçus sont renvoyées sous forme de chaînes.
`get(url, data)`	Envoie une requête HTTP GET à l'URL avec une charge utile *data* facultative et renvoie la réponse.
`head(url)`	Envoie une requête HTTP HEAD à l'URL et renvoie la réponse.
`options(url)`	Envoie une requête HTTP OPTIONS à l'URL et renvoie la réponse.
`patch(url, data)`	Envoie une requête HTTP PATCH à l'URL avec une charge utile *data* facultative et renvoie la réponse.
`put(url, data)`	Envoie une requête HTTP PUT à l'URL avec une charge utile *data* facultative et renvoie la réponse.
`post(url, data)`	Envoie une requête HTTP POST à l'URL avec une charge utile *data* facultative et renvoie la réponse.
`getStatus()`	Renvoie le code d'état de la dernière requête HTTP.
`setProxy(proxy)`	Définit le proxy HTTP sur la valeur "proxy". Si ce paramètre est vide, aucun proxy n'est utilisé.

`setHttpAuth(bitmask, username, password)`	Définit les méthodes d'authentification HTTP activées (HTTPAUTH_BASIC, HTTPAUTH_DIGEST, HTTPAUTH_NEGOTIATE, HTTPAUTH_NTLM, HTTPAUTH_NONE) dans le paramètre 'bitmask'.
	L'indicateur HTTPAUTH_NONE permet de désactiver l'authentification HTTP.
	Exemples :
	`request.setHttpAuth(HTTPAUTH_NTLM | HTTPAUTH_BASIC, username, password)`
	`request.setHttpAuth(HTTPAUTH_NONE)`

`trace(url, data)`	Envoie une requête HTTP TRACE à l'URL avec une charge utile *data* facultative et renvoie la réponse.

Exemple :

```
try {
    Zabbix.log(4, 'jira webhook script value='+value);

    var result = {
        'tags': {
            'endpoint': 'jira'
        }
    },
    params = JSON.parse(value),
    req = new HttpRequest(),
    fields = {},
    resp;
```

```
    req.addHeader('Content-Type: application/json');
    req.addHeader('Authorization: Basic '+params.authentication);

    fields.summary = params.summary;
    fields.description = params.description;
    fields.project = {"key": params.project_key};
    fields.issuetype = {"id": params.issue_id};
    resp = req.post('https://jira.example.com/rest/api/2/issue/',
        JSON.stringify({"fields": fields})
    );

    if (req.getStatus() != 201) {
        throw 'Response code: '+req.getStatus();
    }

    resp = JSON.parse(resp);
    result.tags.issue_id = resp.id;
    result.tags.issue_key = resp.key;
} catch (error) {
    Zabbix.log(4, 'jira issue creation failed json : '+JSON.stringify({"fields": fields}));
    Zabbix.log(4, 'jira issue creation failed : '+error);

    result = {};
}

return JSON.stringify(result);
```

XML

L'objet XML permet le traitement des données XML dans le prétraitement des éléments, de la découverte de bas niveau et des webhooks.

Attention:
Pour utiliser l'objet XML, le serveur/proxy doit être compilé avec la prise en charge de libxml2.

Method	Description
`XML.query(data, expression)`	Récupère le contenu d'un nœud à l'aide de XPath. Renvoie null si le nœud est introuvable. **expression** - une expression XPath ; **data** - des données XML sous forme de chaîne.
`XML.toJson(data)`	Convertit des données au format XML en JSON.
`XML.fromJson(object)`	Convertit des données au format JSON en XML.

Exemple :

Entrée :

```
<menu>
    <food type = "breakfast">
        <name>Chocolate</name>
        <price>$5.95</price>
        <description></description>
        <calories>650</calories>
    </food>
</menu>
```

Sortie :

```
{
    "menu": {
        "food": {
            "@type": "breakfast",
            "name": "Chocolate",
            "price": "$5.95",
```

```
            "description": null,
            "calories": "650"
        }
    }
}
```

Règles de sérialisation

La conversion de XML en JSON sera effectuée selon les règles suivantes (pour les conversions de JSON en XML, les règles inverses sont appliquées) :

1. Les attributs XML seront convertis en clés dont le nom est préfixé par '@'.

Exemple :

Entrée :

```
<xml foo="FOO">
  <bar>
    <baz>BAZ</baz>
  </bar>
</xml>
```

Sortie :

```
{
  "xml": {
    "@foo": "FOO",
    "bar": {
      "baz": "BAZ"
    }
  }
}
```

2. Les éléments auto-fermants (<foo/>) seront convertis avec la valeur 'null'.

Exemple :

Entrée :

```
<xml>
  <foo/>
</xml>
```

Sortie :

```
{
  "xml": {
    "foo": null
  }
}
```

3. Les attributs vides (avec la valeur "") seront convertis avec une valeur de chaîne vide ('').

Exemple :

Entrée :

```
<xml>
  <foo bar="" />
</xml>
```

Sortie :

```
{
  "xml": {
    "foo": {
      "@bar": ""
    }
  }
}
```

4. Plusieurs nœuds enfants portant le même nom d'élément seront convertis en une seule clé dont la valeur sera un tableau de valeurs.

Exemple :

Entrée :

```
<xml>
  <foo>BAR</foo>
  <foo>BAZ</foo>
  <foo>QUX</foo>
</xml>
```

Sortie :

```
{
  "xml": {
    "foo": ["BAR", "BAZ", "QUX"]
  }
}
```

5. Si un élément de texte n'a ni attributs ni enfants, il sera converti en chaîne de caractères.

Exemple :

Entrée :

```
<xml>
    <foo>BAZ</foo>
</xml>
```

Sortie :

```
{
  "xml": {
    "foo": "BAZ"
   }
}
```

6. Si un élément de texte n'a pas d'enfants mais possède des attributs, le contenu textuel sera converti en un élément avec la clé '#text' et le contenu comme valeur ; les attributs seront convertis comme décrit dans la règle de sérialisation 1.

Exemple :

Entrée :

```
<xml>
  <foo bar="BAR">
    BAZ
  </foo>
</xml>
```

Sortie :

```
{
  "xml": {
    "foo": {
      "@bar": "BAR",
      "#text": "BAZ"
    }
  }
}
```

Fonctions JavaScript globales

Des fonctions JavaScript globales supplémentaires ont été implémentées avec Duktape :

- `btoa(data)` - encode les données en chaîne base64.
- `atob(base64_string)` - décode une chaîne base64 en tampon Uint8Array.

```
try {
    b64 = btoa("test string");
    buffer = atob(b64);
```

```
    // Notez que la logique de décodage dépend du format des données du tampon.
    decoded = String.fromCharCode.apply(this, [].slice.call(buffer));
}
catch (error) {
    return {'error.name' : error.name, 'error.message' : error.message};
}
```

- `md5(data)` - calcule le hachage MD5 des données.
- `sha256(data)` - calcule le hachage SHA256 des données.
- `hmac('<hash type>',key,data)` - renvoie un hachage HMAC sous forme de chaîne au format hexadécimal ; `md5` et `sha256` sont pris en charge comme `hash type` ; les paramètres `key` et `data` prennent en charge les données binaires.

  Exemples :

  - `hmac('md5',key,data)`
  - `hmac('sha256',key,data)`

- `sign(hash,key,data)` - renvoie la signature calculée (signature RSA avec SHA-256) sous forme de chaîne, où :
 hash - seul `sha256` est autorisé, sinon une erreur est générée.
 key - la clé privée. Elle doit être conforme à la norme PKCS#1 ou PKCS#8. La clé peut être fournie sous différentes formes :

  - avec des espaces à la place des sauts de ligne
  - avec `\n` échappés ou non échappés à la place des sauts de ligne
  - sans aucun saut de ligne, sous forme de chaîne sur une seule ligne
  - sous forme de chaîne au format JSON

  La clé peut également être chargée à partir d'une macro utilisateur / macro secrète / coffre-fort.

  data - les données qui seront signées. Il peut s'agir d'une chaîne (les données binaires sont également prises en charge) ou d'un tampon (Uint8Array/ArrayBuffer).

  Exemple :

  - `sign('sha256',key,data)`

  OpenSSL ou GnuTLS est utilisé pour calculer les signatures. Si Zabbix a été compilé sans l'une de ces bibliothèques de chiffrement, une erreur sera générée ('missing OpenSSL or GnuTLS library').

Objets JavaScript de l'élément de navigateur

Aperçu

Cette section décrit les ajouts de Zabbix au langage JavaScript implémenté avec Duktape pour une utilisation dans le script de l'élément Browser. Ces ajouts complètent les objets JavaScript décrits sur la page Objets JavaScript supplémentaires.

Browser

L'objet `Browser` gère les sessions WebDriver, en initialisant une session lors de sa création et en y mettant fin lors de sa destruction. Un seul script peut prendre en charge jusqu'à quatre objets `Browser`.

Pour construire un objet `Browser`, utilisez la syntaxe `new Browser(options)`. Le paramètre `options` (*objet JSON*) spécifie les options du navigateur, généralement le résultat de la méthode d'options WebDriver (par exemple, `Browser.chromeOptions()`).

Les méthodes suivantes sont prises en charge avec l'objet `Browser`.

Method	Description
`navigate(url)`	Accéder à l'URL spécifiée. Paramètres : `url` - (string) URL vers laquelle accéder.
`getUrl()`	Renvoie une chaîne contenant l'URL de la page ouverte.
`getPageSource()`	Renvoie une chaîne contenant le code source de la page ouverte.

Method	Description
`findElement(strategy, selector)`	Renvoie un objet `Element` contenant un élément de la page ouverte (ou renvoie `null` si aucun élément ne correspond à `strategy` et `selector`). Paramètres : `strategy` - (string, css selector/link text/partial link text/tag name/xpath) Stratégie de localisation ; `selector` - (string) Sélecteur d'élément utilisant la stratégie de localisation spécifiée.
`findElements(strategy, target)`	Renvoie un tableau d'objets `Element` contenant plusieurs éléments de la page ouverte (ou renvoie un tableau vide si aucun élément ne correspond à la stratégie de localisation et à la cible). Paramètres : `strategy` - (string, css selector/link text/partial link text/tag name/xpath) Stratégie de localisation ; `target` - (string) Sélecteur d'élément utilisant la stratégie de localisation spécifiée.
`getCookies()`	Renvoie un tableau d'objets `Cookie`.
`addCookie(cookie)`	Définit des cookies. Paramètres : `cookie` - (objet `Cookie`) Cookie à définir.
`getScreenshot()`	Renvoie une chaîne encodée en base64 représentant une image de la zone d'affichage du navigateur.
`setScreenSize(x,y)`	Définit la taille de la zone d'affichage du navigateur. Paramètres : `x` - (string) Largeur en pixels ; `y` - (string) Hauteur en pixels.
`setScriptTimeout(timeout)`	Définit le délai d'expiration du chargement du script. Paramètres : `timeout` - (integer) Valeur du délai d'expiration en millisecondes.
`setSessionTimeout(timeout)`	Définit le délai d'expiration de la session (chargement de page). Paramètres : `timeout` - (integer) Valeur du délai d'expiration en millisecondes.
`setElementWaitTimeout(timeout)`	Définit le délai d'expiration (implicite) de la stratégie de localisation des éléments. Paramètres : `timeout` - (integer) Valeur du délai d'expiration en millisecondes.
`collectPerfEntries(mark)`	Collecte les entrées de performance à récupérer avec la méthode `getResult()`. Paramètres : `mark` - (string, optional) Marque d'instantané de performance.
`getRawPerfEntries()`	Renvoie un tableau d'objets d'entrée de performance.
`getResult()`	Renvoie un objet `Result` contenant les statistiques de la session du navigateur (informations d'erreur, instantanés de performance, etc.).
`getError()`	Renvoie un objet `BrowserError` contenant les erreurs du navigateur (ou renvoie `null` s'il n'y a pas d'erreurs du navigateur).
`setError(message)`	Définit un message d'erreur personnalisé à inclure dans l'objet `Result`. Paramètres : `message` - (string) Message d'erreur.
`discardError()`	Supprime l'erreur à renvoyer dans l'objet `Result`.
`getAlert()`	Renvoie un objet `Alert` contenant les alertes du navigateur (ou renvoie `null` s'il n'y a pas d'alertes du navigateur).
`chromeOptions()`	Renvoie un objet `chromeOptions` contenant des options prédéfinies du navigateur Chrome.
`firefoxOptions()`	Renvoie un objet `firefoxOptions` contenant des options prédéfinies du navigateur Firefox.
`safariOptions()`	Renvoie un objet `safariOptions` contenant des options prédéfinies du navigateur Safari.

Method	Description
`edgeOptions()`	Renvoie un objet `edgeOptions` contenant des options prédéfinies du navigateur Edge.
`switchFrame(target)`	Bascule vers le cadre spécifié. Paramètres : `target` - (élément du navigateur ou integer, optional) Cadre cible. Pour sélectionner un cadre par élément, transmettez l'élément. Pour sélectionner un cadre par index, transmettez le nombre. Si laissé vide, basculera vers le contexte de navigation de niveau supérieur.

Toutes les méthodes de Browser peuvent générer les erreurs suivantes :

- `BrowserError` - dérivé de l'objet `Error`, généré si le constructeur `Browser` échoue ; contient une propriété supplémentaire `browser` avec un objet `Browser` qui a généré ce `BrowserError`.
- `WebdriverError` - dérivé de `BrowserError` ; contient les mêmes propriétés que l'objet `BrowserError`, qui indiquent si l'erreur a été générée en réponse à une erreur dans la réponse WebDriver.

Element

L'objet `Element` est renvoyé par les méthodes `findElement()`/`findElements()` de l'objet `Browser` et ne peut pas être construit directement.

L'objet `Element` représente un élément dans la page web et fournit des méthodes pour interagir avec lui.

Les méthodes suivantes sont prises en charge avec l'objet `Element`.

Method	Description
`getAttribute(name)`	Renvoie une chaîne de valeur d'attribut de l'attribut de l'élément (ou renvoie `null` si l'attribut spécifié n'a pas été trouvé). Parameters: `name` - (string) Nom de l'attribut.
`getProperty(name)`	Renvoie une chaîne de valeur de propriété de la propriété de l'élément (ou renvoie `null` si la propriété spécifiée n'a pas été trouvée). Parameters: `name` - (string) Nom de la propriété.
`getText()`	Renvoie une chaîne de valeur de texte du texte de l'élément.
`click()`	Clique sur un élément.
`clear()`	Efface le contenu d'un élément modifiable.
`sendKeys(keys)`	Envoie des touches. Parameters: `keys` - (string) Touches à envoyer.

Cookie

L'objet `Cookie` est renvoyé par la méthode `getCookies()` de l'objet `Browser` et transmis à la méthode `addCookie()`.

Bien que l'objet `Cookie` n'ait aucune méthode, il peut contenir les propriétés suivantes :

Propriété	Type	Description
name	string	Nom du cookie.
value	string	Valeur du cookie.
path	string	Chemin pour lequel le cookie est valide. Par défaut, `"/"` si omis lors de l'ajout d'un cookie.
domain	string	Domaine auquel le cookie est visible. Par défaut, le domaine URL du document actif du contexte de navigation actuel de la session si omis lors de l'ajout d'un cookie.
secure	boolean	Indique si le cookie est sécurisé. Par défaut, `false` si omis lors de l'ajout d'un cookie.

Propriété	Type	Description
httpOnly	boolean	Indique si le cookie est de type HTTP-only. Par défaut, `false` si omis lors de l'ajout d'un cookie.
expiry	integer	Heure d'expiration du cookie (en secondes depuis l'époque Unix). Ne doit pas être défini s'il est omis lors de l'ajout d'un cookie.
sameSite	string	L'attribut `sameSite` du cookie, qui contrôle si le cookie doit être limité à un contexte propriétaire ou de même site. Peut être défini sur `"Lax"` ou `"Strict"`. Par défaut, `"None"` si omis lors de l'ajout d'un cookie.

Alerte

L'objet `Alert` représente une alerte de page web, est renvoyé par la méthode `getAlert()` de l'objet `Browser` et ne peut pas être construit directement.

L'objet `Alert` contient la propriété `text` avec le texte de l'alerte (ou `null` s'il n'y a pas d'alertes).

Les méthodes suivantes sont prises en charge avec l'objet `Alert`.

Method	Description
`accept()`	Accepte l'alerte.
`dismiss()`	Ignore l'alerte.

Résultat

L'objet `Result` contient des statistiques de session et est renvoyé par la méthode `getResult()` de l'objet `Browser`.

En général, l'objet `Result` est converti en chaîne et renvoyé par le script, puis analysé en valeurs d'éléments dépendants via le prétraitement.

Bien que l'objet `Result` ne possède aucune méthode, il peut contenir les propriétés suivantes.

Propriété		Type	Description
duration		string	Durée de la session entre la création de la session et la récupération du résultat.
error		object	Informations sur l'erreur.
	http_status	integer	Code d'état HTTP renvoyé par le WebDriver (ou 0 s'il n'y a pas d'erreurs WebDriver).
	error_code	string	Erreur renvoyée par le WebDriver (ou une chaîne vide s'il n'y a pas d'erreurs WebDriver).
	message	string	Message d'erreur WebDriver (ou une chaîne vide s'il n'y a pas d'erreurs WebDriver).
performance_data		object	Statistiques de performance.
	summary	object	Résumé des performances.
		navigation object	Résumé de la navigation.
		resource object	Résumé des ressources.
	details	array of objects	Statistiques de performance après chaque opération ayant pu entraîner une navigation.
		mark string	*(facultatif)* Marque d'instantané de performance spécifiée avec la méthode `collectPerfEntries()`.
		navigation object	Statistiques de navigation.
		resource object	Résumé des ressources pour cette étape.
		user array of objects	Tableau de statistiques de type mark/measure.
	marks	array of objects	Index des instantanés de performance marqués.
		name string	Nom de la marque de l'instantané de performance.
		index integer	Index de l'instantané de performance dans le tableau details.

Prétraitement CSV vers JSON

Présentation

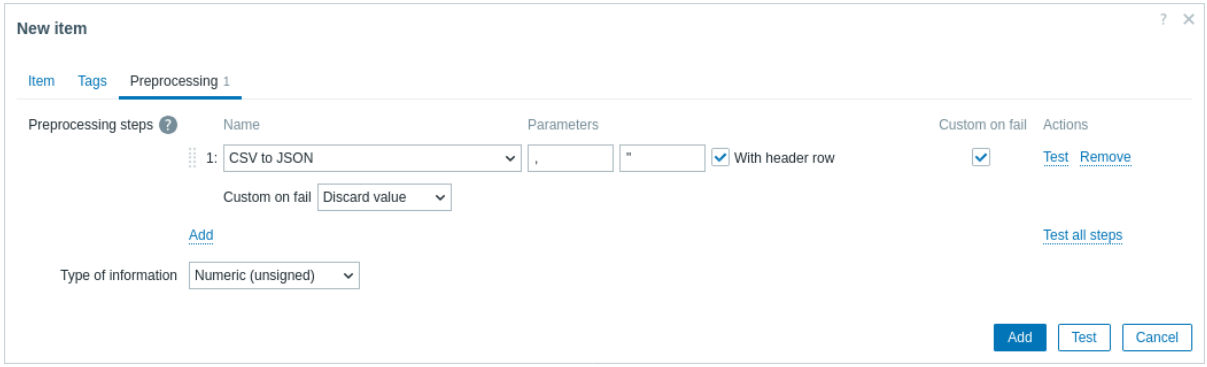
Dans cette étape de prétraitement, il est possible de convertir les données d'un fichier CSV au format JSON. Cette fonctionnalité est prise en charge dans :

- les éléments (prototypes d'éléments)
- les règles de découverte de bas niveau

Configuration

Pour configurer une étape de prétraitement CSV vers JSON :

1. Accédez à l'onglet *Prétraitement* dans la configuration de l'élément/de la règle de découverte.
2. Cliquez sur *Ajouter*.
3. Sélectionnez l'option *CSV vers JSON*.



Le premier paramètre permet de définir un délimiteur personnalisé. Notez que si la première ligne de l'entrée CSV commence par « Sep= » et est suivie d'un seul caractère UTF-8, ce caractère sera utilisé comme délimiteur si le premier paramètre n'est pas défini. Si le premier paramètre n'est pas défini et qu'aucun délimiteur n'est récupéré depuis la ligne « Sep= », une virgule est utilisée comme séparateur.

Le deuxième paramètre facultatif permet de définir un symbole de guillemet.

Si la case *Avec ligne d'en-tête* est cochée, les valeurs de la ligne d'en-tête seront interprétées comme des noms de colonnes (voir Traitement de l'en-tête pour plus d'informations).

Si la case *Personnalisé en cas d'échec* est cochée, l'élément ne deviendra pas non pris en charge en cas d'échec d'une étape de prétraitement. En outre, des options personnalisées de gestion des erreurs peuvent être définies : ignorer la valeur, définir une valeur spécifiée ou définir un message d'erreur spécifié.

Traitement d'en-tête

La ligne d'en-tête du fichier CSV peut être traitée de deux manières différentes :

- Si la case *Avec ligne d'en-tête* est cochée, les valeurs de la ligne d'en-tête sont interprétées comme des noms de colonne. Dans ce cas, les noms de colonne doivent être uniques et la ligne de données ne doit pas contenir plus de colonnes que la ligne d'en-tête ;
- Si la case *Avec ligne d'en-tête* n'est pas cochée, la ligne d'en-tête est interprétée comme des données. Les noms de colonnes sont générés automatiquement (1,2,3,4...)

Exemple de fichier CSV :

```
Nr,Item name,Key,Qty
1,active agent item,agent.hostname,33
"2","passive agent item","agent.version","44"
3,"active,passive agent items",agent.ping,55
```

Note:
Un guillemet dans un champ entre guillemets dans l'entrée doit être échappé en le faisant précéder d'un autre guillemet.

Traitement de la ligne d'en-tête

Sortie JSON lorsqu'une ligne d'en-tête est attendue :

```
[
   {
      "Nr":"1",
      "Item name":"active agent item",
      "Key":"agent.hostname",
      "Qty":"33"
   },
   {
      "Nr":"2",
```

```
      "Item name":"passive agent item",
      "Key":"agent.version",
      "Qty":"44"
   },
   {
      "Nr":"3",
      "Item name":"active,passive agent items",
      "Key":"agent.ping",
      "Qty":"55"
   }
]
```

Pas de traitement de ligne d'en-tête

Sortie JSON lorsqu'une ligne d'en-tête n'est pas attendue :

```
[
   {
      "1":"Nr",
      "2":"Item name",
      "3":"Key"
      "4":"Qty"
   },
   {
      "1":"1",
      "2":"active agent item",
      "3":"agent.hostname"
      "4":"33"
   },
   {
      "1":"2",
      "2":"passive agent item",
      "3":"agent.version"
      "4":"44"
   },
   {
      "1":"3",
      "2":"active,passive agent items",
      "3":"agent.ping"
      "4":"55"
   }
]
```

Types d'élément

Vue d'ensemble

Les types d'éléments couvrent différentes méthodes d'acquisition de données à partir de votre système. Chaque type d'élément possède son propre ensemble de clés d'élément prises en charge et de paramètres requis.

Les types d'éléments suivants sont actuellement proposés par Zabbix :

- Zabbix agent
  - Surveillance des fichiers journaux
- Vérification simple
  - Surveillance VMware
- SNMP agent
- Interruption SNMP
- Zabbix interne
- Zabbix trapper
- Vérification externe
- Surveillance de base de données
- HTTP agent
  - Vérification Prometheus
- IPMI agent

- SSH agent
- Telnet agent
- JMX agent
- Élément calculé
- Élément dépendant
- Élément script
- Élément navigateur
- Requête de télémétrie

Les détails de tous les types d'éléments sont inclus dans les sous-pages de cette section. Bien que les types d'éléments offrent de nombreuses options pour la collecte de données, d'autres options sont disponibles via les paramètres utilisateur ou les modules chargeables.

Certaines vérifications sont effectuées uniquement par le serveur Zabbix (en tant que surveillance sans agent), tandis que d'autres nécessitent un Zabbix agent ou même un Zabbix Java gateway (avec la surveillance JMX).

> **Attention:**
> Si un type d'élément particulier nécessite une interface particulière (par exemple, une vérification IPMI nécessite une interface IPMI sur l'hôte), cette interface doit exister dans la définition de l'hôte.

Plusieurs interfaces peuvent être définies dans la définition de l'hôte : Zabbix agent, SNMP agent, JMX et IPMI. Si un élément peut utiliser plusieurs interfaces, il recherchera, parmi les interfaces disponibles de l'hôte (dans l'ordre : Agent→SNMP→JMX→IPMI), la première interface appropriée à laquelle être associé.

Tous les éléments qui renvoient du texte (types d'informations caractère, journal et texte) peuvent également renvoyer uniquement des espaces blancs, le cas échéant, en définissant la valeur renvoyée sur une chaîne vide (pris en charge depuis la version 2.0).

agent Zabbix

Vue d'ensemble

Cette section fournit des détails sur les clés d'élément qui utilisent la communication avec l'agent Zabbix pour la collecte des données.

Il existe des vérifications d'agent passives et actives. Lors de la configuration d'un élément, vous pouvez sélectionner le type requis :

- *Zabbix agent* - pour les vérifications passives
- *Zabbix agent (active)* - pour les vérifications actives

Notez que toutes les clés d'élément prises en charge par Zabbix agent sont également prises en charge par la nouvelle génération Zabbix agent 2. Consultez les clés d'élément supplémentaires que vous pouvez utiliser uniquement avec agent 2.

Clés d'éléments prises en charge

Les clés d'éléments que vous pouvez utiliser avec l'agent Zabbix sont répertoriées ci-dessous.

Les clés d'éléments sont répertoriées sans paramètres ni informations supplémentaires. Cliquez sur la clé d'élément pour afficher tous les détails.

Clé d'élément	Description	Groupe d'éléments
kernel.maxfiles	Le nombre maximal de fichiers ouverts pris en charge par le système d'exploitation.	Noyau
kernel.maxproc	Le nombre maximal de processus pris en charge par le système d'exploitation.	
kernel.openfiles	Le nombre de descripteurs de fichiers actuellement ouverts.	
log	La surveillance d'un fichier journal.	Surveillance des journaux
log.count	Le nombre de lignes correspondantes dans un fichier journal surveillé.	
logrt	La surveillance d'un fichier journal soumis à une rotation.	
logrt.count	Le nombre de lignes correspondantes dans un fichier journal surveillé et soumis à une rotation.	
modbus.get	Lit les données Modbus.	Modbus
net.dns	Vérifie l'état d'un service DNS.	Réseau
net.dns.perf	Vérifie les performances d'un service DNS.	
net.dns.record	Effectue une requête DNS.	

Clé d'élément	Description	Groupe d'éléments
net.if.collisions	Le nombre de collisions hors fenêtre.	
net.if.discovery	La liste des interfaces réseau.	
net.if.get	La liste des interfaces réseau avec des informations détaillées.	
net.if.in	Les statistiques du trafic entrant sur une interface réseau.	
net.if.out	Les statistiques du trafic sortant sur une interface réseau.	
net.if.total	La somme des statistiques du trafic entrant et sortant sur une interface réseau.	
net.tcp.listen	Vérifie si ce port TCP est à l'état LISTEN.	
net.tcp.port	Vérifie s'il est possible d'établir une connexion TCP vers le port spécifié.	
net.tcp.service	Vérifie si un service est en cours d'exécution et accepte les connexions TCP.	
net.tcp.service.perf	Vérifie les performances d'un service TCP.	
net.tcp.socket.count	Renvoie le nombre de sockets TCP correspondant aux paramètres.	
net.udp.listen	Vérifie si ce port UDP est à l'état LISTEN.	
net.udp.service	Vérifie si un service est en cours d'exécution et répond aux requêtes UDP.	
net.udp.service.perf	Vérifie les performances d'un service UDP.	
net.udp.socket.count	Renvoie le nombre de sockets UDP correspondant aux paramètres.	
proc.cpu.util	Le pourcentage d'utilisation du processeur par le processus.	Processus
proc.get	La liste des processus du système d'exploitation et de leurs paramètres.	
proc.mem	La mémoire utilisée par le processus, en octets.	
proc.num	Le nombre de processus.	
sensor	La valeur relevée par un capteur matériel.	Capteurs
system.boottime	L'heure de démarrage du système.	Système
system.cpu.discovery	La liste des processeurs et cœurs de processeur détectés.	
system.cpu.intr	Les interruptions des périphériques.	
system.cpu.load	La charge du processeur.	
system.cpu.num	Le nombre de processeurs.	
system.cpu.switches	Le nombre de changements de contexte.	
system.cpu.util	Le pourcentage d'utilisation du processeur.	
system.hostname	Le nom d'hôte du système.	
system.hw.chassis	Les informations relatives au châssis.	
system.hw.cpu	Les informations relatives au processeur.	
system.hw.devices	La liste des périphériques PCI ou USB.	
system.hw.macaddr	La liste des adresses MAC.	
system.localtime	L'heure du système.	
system.run	Exécute la commande spécifiée sur l'hôte.	
system.stat	Les statistiques du système.	
system.sw.arch	Les informations relatives à l'architecture logicielle.	
system.sw.os	Les informations relatives au système d'exploitation.	
system.sw.os.get	Des informations détaillées sur le système d'exploitation (version, type, nom de la distribution, version mineure et majeure, etc.).	
system.sw.packages	La liste des paquets installés.	
system.sw.packages.get	Une liste détaillée des paquets installés.	
system.swap.in	Les statistiques de swap-in (du périphérique vers la mémoire).	
system.swap.out	Les statistiques de swap-out (de la mémoire vers le périphérique).	
system.swap.size	La taille de l'espace d'échange en octets ou en pourcentage du total.	
system.uname	L'identification du système.	
system.uptime	La durée de fonctionnement du système, en secondes.	
system.users.num	Le nombre d'utilisateurs connectés.	
vfs.dev.discovery	La liste des périphériques bloc et de leur type.	Systèmes de fichiers virtuels
vfs.dev.get	La liste des périphériques bloc avec des informations détaillées.	
vfs.dev.read	Les statistiques de lecture du disque.	
vfs.dev.write	Les statistiques d'écriture du disque.	
vfs.dir.count	Le nombre d'entrées du répertoire.	
vfs.dir.get	La liste des entrées du répertoire.	
vfs.dir.size	La taille du répertoire.	
vfs.file.cksum	La somme de contrôle du fichier, calculée à l'aide de l'algorithme UNIX cksum.	
vfs.file.contents	Récupère le contenu d'un fichier.	
vfs.file.exists	Vérifie si le fichier existe.	
vfs.file.get	Renvoie des informations sur un fichier.	
vfs.file.md5sum	La somme de contrôle MD5 du fichier.	

Clé d'élément	Description	Groupe d'éléments
vfs.file.owner	Récupère le propriétaire d'un fichier.	
vfs.file.permissions	Renvoie une chaîne de 4 chiffres contenant le nombre octal correspondant aux permissions UNIX.	
vfs.file.regexp	Récupère une chaîne dans le fichier.	
vfs.file.regmatch	Recherche une chaîne dans le fichier.	
vfs.file.size	La taille du fichier.	
vfs.file.time	Les informations temporelles du fichier.	
vfs.fs.discovery	La liste des systèmes de fichiers montés avec leur type et leurs options de montage.	
vfs.fs.get	La liste des systèmes de fichiers montés avec leur type et leurs options de montage et, en mode *full*, l'espace disque disponible et les statistiques des inodes.	
vfs.fs.inode	Le nombre ou le pourcentage d'inodes.	
vfs.fs.size	L'espace disque en octets ou en pourcentage du total.	
vm.memory.size	La taille de la mémoire en octets ou en pourcentage du total.	Mémoire virtuelle
web.page.get	Récupère le contenu d'une page web.	Surveillance web
web.page.perf	Le temps de chargement d'une page web complète.	
web.page.regexp	Recherche une chaîne sur la page web.	
agent.hostmetadata	Les métadonnées de l'hôte de l'agent.	Zabbix
agent.hostname	Le nom d'hôte de l'agent.	
agent.ping	La vérification de la disponibilité de l'agent.	
agent.variant	La variante de l'agent Zabbix (agent Zabbix ou agent Zabbix 2).	
agent.version	La version de l'agent Zabbix.	
zabbix.stats	Renvoie un ensemble de métriques internes du serveur ou du proxy Zabbix. Si `ip` et `port` sont spécifiés, les métriques sont récupérées depuis l'instance distante ; sinon, depuis l'instance locale.	
zabbix.stats (queue)	Renvoie le nombre d'éléments surveillés dans la file d'attente qui sont en retard sur le serveur ou le proxy Zabbix. Si `ip` et `port` sont spécifiés, les métriques sont récupérées depuis l'instance distante ; sinon, depuis l'instance locale.	

Plateformes prises en charge

Sauf indication contraire dans les détails de l'élément, les éléments d'agent (et tous les paramètres) sont pris en charge sur :

- **Linux**
- **FreeBSD**
- **Solaris**
- **HP-UX**
- **AIX**
- **MacOS X**
- **OpenBSD**
- **NetBSD**

De nombreux éléments d'agent sont également pris en charge sous **Windows**. Consultez la page Éléments d'agent Windows pour plus de détails.

Détails de la clé d'élément

Les paramètres sans crochets angulaires sont obligatoires. Les paramètres marqués par des crochets angulaires **< >** sont facultatifs.

kernel.maxfiles

 Le nombre maximal de fichiers ouverts pris en charge par le système d'exploitation.
 Valeur de retour : *Integer*.
 Plateformes prises en charge : Linux, FreeBSD, MacOS X, OpenBSD, NetBSD.

kernel.maxproc

 Le nombre maximal de processus pris en charge par le système d'exploitation.
 Valeur de retour : *Integer*.
 Plateformes prises en charge : Linux 2.6 et versions ultérieures, FreeBSD, Solaris, MacOS X, OpenBSD, NetBSD.

kernel.openfiles

 Le nombre de descripteurs de fichiers actuellement ouverts.
 Valeur de retour : *Integer*.
 Plateformes prises en charge : Linux (l'élément peut fonctionner sur d'autres plateformes de type UNIX).

log[file,<regexp>,<encoding>,<maxlines>,<mode>,<output>,<maxdelay>,<options>,<persistent dir>]

 La surveillance d'un fichier journal.
 Valeur de retour: *Log*.
 Voir les plates-formes prises en charge.

Paramètres:

- **file** - le chemin complet et le nom d'un fichier journal;
- **regexp** - une expression régulière décrivant le motif requis;
- **encoding** - l'identifiant de la page de code;
- **maxlines** - le nombre maximal de nouvelles lignes par seconde que l'agent enverra au serveur ou au proxy Zabbix. Ce paramètre remplace la valeur de `MaxLinesPerSecond` dans `zabbix_agentd.conf`;
- **mode** - valeurs possibles:
  - *all* (par défaut);
  - *skip* - ignorer le traitement des données plus anciennes (n'affecte que les éléments nouvellement créés);
- **output** - un modèle de formatage de sortie facultatif. La séquence d'échappement **\0** est remplacée par la partie correspondante du texte (du premier caractère où la correspondance commence jusqu'au caractère où elle se termine), tandis que la séquence d'échappement **\N** (où N=1...9) est remplacée par le N-ième groupe correspondant (ou par une chaîne vide si N dépasse le nombre de groupes capturés);
- **maxdelay** - le délai maximal en secondes. Type: float. Valeurs:
  - 0 - (par défaut) n'ignore jamais les lignes du fichier journal;
  - > 0.0 - ignorer les lignes plus anciennes afin d'obtenir les lignes les plus récentes analysées dans un délai de "maxdelay" secondes. Lisez les remarques sur maxdelay avant de l'utiliser!
- **options** - options supplémentaires:
mtime-noreread - enregistrements non uniques, relire uniquement si la taille du fichier change (ignorer le changement de l'heure de modification). (Ce paramètre est obsolète depuis la version 5.0.2, car mtime est désormais ignoré.)
- **persistent dir** (uniquement dans zabbix_agentd sur les systèmes Unix; non pris en charge dans Zabbix agent 2) - le chemin absolu du répertoire où stocker les fichiers persistants. Voir aussi les remarques supplémentaires sur les fichiers persistants.

Commentaires:

- L'élément doit être configuré comme une vérification active.
- Si le fichier est manquant ou si les permissions n'autorisent pas l'accès, l'élément passe en non pris en charge.
- Si `output` est laissé vide, la ligne entière contenant le texte correspondant est renvoyée. Notez que tous les types d'expressions régulières globales, sauf *Matches regular expression*, renvoient toujours la ligne entière correspondante et que le paramètre `output` est ignoré.
- L'extraction du contenu à l'aide du paramètre `output` s'effectue sur l'agent.

Exemples:

```
log[/var/log/syslog]
log[/var/log/syslog,error]
log[/home/zabbix/logs/logfile,,,100]
```

Exemple d'utilisation du paramètre `output` pour extraire un nombre d'un enregistrement de journal:

```
log[/app1/app.log,"task run [0-9.]+ sec, processed ([0-9]+) records, [0-9]+ errors",,,,\1] #cet élément co
```

Exemple d'utilisation du paramètre `output` pour réécrire un enregistrement de journal avant de l'envoyer au serveur:

```
log[/app1/app.log,"([0-9 :-]+) task run ([0-9.]+) sec, processed ([0-9]+) records, ([0-9]+) errors",,,,"\1
```

log.count[file,<regexp>,<encoding>,<maxproclines>,<mode>,<maxdelay>,<options>,<persistent dir>]

 Le nombre de lignes correspondantes dans un fichier journal surveillé.
 Valeur de retour: *Integer*.
 Voir les plates-formes prises en charge.

Paramètres:

- **file** - le chemin complet et le nom du fichier journal;
- **regexp** - une expression régulière décrivant le motif requis;
- **encoding** - l'identifiant de page de code;
- **maxproclines** - le nombre maximal de nouvelles lignes par seconde que l'agent analysera (ne peut pas dépasser 10000). La valeur par défaut est 10*`MaxLinesPerSecond` dans `zabbix_agentd.conf`.
- **mode** - valeurs possibles:
  - *all* (par défaut);
  - *skip* - ignorer le traitement des données plus anciennes (n'affecte que les éléments nouvellement créés).
- **maxdelay** - le délai maximal en secondes. Type: float. Valeurs:

    - 0 - (par défaut) ne jamais ignorer les lignes du fichier journal;
    - > 0.0 - ignorer les lignes plus anciennes afin d'obtenir les lignes les plus récentes analysées dans un délai de "maxdelay" secondes. Lisez les notes sur maxdelay avant de l'utiliser!
- **options** - options supplémentaires:
mtime-noreread - enregistrements non uniques, relire uniquement si la taille du fichier change (ignorer le changement de l'heure de modification). (Ce paramètre est obsolète depuis la version 5.0.2, car mtime est désormais ignoré.)
- **persistent dir** (uniquement dans `zabbix_agentd` sur les systèmes Unix; non pris en charge dans Zabbix agent 2) - le chemin absolu du répertoire où stocker les fichiers persistants. Voir aussi les notes supplémentaires sur les fichiers persistants.

Commentaires:

- L'élément doit être configuré comme une vérification active.
- Les lignes correspondantes sont comptées parmi les nouvelles lignes depuis la dernière vérification du journal par l'agent, et dépendent donc de l'intervalle de mise à jour de l'élément.
- Si le fichier est manquant ou si les autorisations ne permettent pas d'y accéder, l'élément passe en état non pris en charge.

logrt[file regexp,<regexp>,<encoding>,<maxlines>,<mode>,<output>,<maxdelay>,<options>,<persistent dir>]

 La surveillance d'un fichier journal qui est roté.
 Valeur de retour: *Log*.
 Voir les plateformes prises en charge.

Paramètres:

- **file regexp** - le chemin absolu vers le fichier, avec le nom de fichier spécifié à l'aide d'une expression régulière. Notez que l'expression régulière s'applique uniquement au nom du fichier et n'a pas besoin de correspondre au nom entier (par exemple, `/path/to/agent` correspondra à `zabbix_agentd.log`).
- **regexp** - une expression régulière décrivant le motif de contenu requis.
- **encoding** - l'identifiant de page de code identifiant.
- **maxlines** - le nombre maximal de nouvelles lignes par seconde que l'agent enverra au serveur Zabbix ou au proxy. Ce paramètre remplace la valeur de `MaxLinesPerSecond` dans `zabbix_agentd.conf`.
- **mode** - valeurs possibles:
    - *all* (par défaut);
    - *skip* - ignorer le traitement des données plus anciennes (n'affecte que les éléments nouvellement créés).
- **output** - un modèle de formatage de sortie facultatif. La séquence d'échappement **\0** est remplacée par la partie correspondante du texte (du premier caractère où la correspondance commence jusqu'au caractère où la correspondance se termine), tandis qu'une séquence d'échappement **\N** (où N=1...9) est remplacée par le N-ième groupe correspondant (ou une chaîne vide si N dépasse le nombre de groupes capturés).
- **maxdelay** - le délai maximal en secondes. Type: float. Valeurs:
    - 0 - (par défaut) ne jamais ignorer les lignes du fichier journal;
    - > 0.0 - ignorer les lignes plus anciennes afin d'obtenir les lignes les plus récentes analysées dans un délai de "maxdelay" secondes. Lisez les notes sur maxdelay avant de l'utiliser!
- **options** - le type de rotation du fichier journal et d'autres options. Valeurs possibles:
    - *rotate* (par défaut).
    - *copytruncate* - notez que *copytruncate* ne peut pas être utilisé avec *maxdelay*. Dans ce cas, *maxdelay* doit être 0 ou non spécifié; voir les notes copytruncate.
    - *mtime-reread* - enregistrements non uniques, relire si l'heure de modification ou la taille change (utilisé par défaut si aucune option `mtime-*` n'est explicitement définie).
    - *mtime-noreread* - enregistrements non uniques, relire uniquement si la taille change (ignorer le changement de l'heure de modification).
- **persistent dir** (uniquement dans `zabbix_agentd` sur les systèmes Unix; non pris en charge dans Zabbix agent 2) - le chemin absolu du répertoire où stocker les fichiers persistants. Voir aussi les notes supplémentaires sur les fichiers persistants.

Commentaires:

- L'élément doit être configuré comme une vérification active.
- La rotation des journaux est basée sur la dernière heure de modification des fichiers.
- Notez que `logrt` est conçu pour fonctionner avec un seul fichier journal actuellement actif, avec plusieurs autres fichiers inactifs correspondants rotés. Si, par exemple, un répertoire contient de nombreux fichiers journaux actifs, un élément `logrt` distinct doit être créé pour chacun d'eux. Sinon, si un seul élément `logrt` prend en charge trop de fichiers, cela peut entraîner un épuisement de la mémoire et un plantage de la surveillance.
- Si `output` est laissé vide, la ligne entière contenant le texte correspondant est renvoyée. Notez que tous les types d'expressions régulières globales, sauf *Matches regular expression*, renvoient toujours la ligne entière correspondante et que le paramètre `output` est ignoré.
- L'extraction du contenu à l'aide du paramètre `output` s'effectue sur l'agent.
- Dans le paramètre `file regexp`, le chemin du répertoire des journaux et l'expression régulière du fichier journal doivent être séparés par le bon séparateur de répertoire:

- Sous Windows, le séparateur doit être une barre oblique inverse (\); les barres obliques (/) peuvent être tolérées à d'autres positions, sauf celle qui sépare le chemin du répertoire des journaux et l'expression régulière du fichier journal (voir les exemples ci-dessous).
- Sur les autres systèmes, il doit s'agir d'une barre oblique (/).

Exemples pour Windows:

```
logrt["c:/dir1/dir2/dir3\filename.*\.log","pattern_to_match"] #cet élément collectera des données à partir
logrt["//example.com/share/dir1/dir2/dir3\filename.*\.log","pattern_to_match"] #cet élément collectera des
```

Exemples pour les autres systèmes:

```
logrt["/home/zabbix/logs/^logfile[0-9]{1,3}$",,,100] #cet élément correspondra à un fichier comme "logfile
logrt["/home/user/^logfile_.*_[0-9]{1,3}$","pattern_to_match","UTF-8",100] #cet élément collectera des don
```

Exemple d'utilisation du paramètre `output` pour extraire un nombre d'un enregistrement de journal:

```
logrt[/app1/^test.*log$,"task run [0-9.]+ sec, processed ([0-9]+) records, [0-9]+ errors",,,,\1] #cet élém
```

Exemple d'utilisation du paramètre `output` pour réécrire un enregistrement de journal avant de l'envoyer au serveur:

```
logrt[/app1/^test.*log$,"([0-9 :-]+) task run ([0-9.]+) sec, processed ([0-9]+) records, ([0-9]+) errors",
```

logrt.count[file regexp,<regexp>,<encoding>,<maxproclines>,<mode>,<maxdelay>,<options>,<persistent dir>]

 Le nombre de lignes correspondantes dans un fichier journal surveillé qui est rotatif.
 Valeur de retour: *Integer*.
 Voir les plateformes prises en charge.

Paramètres:

- **file regexp** - le chemin absolu vers le fichier, avec le nom de fichier spécifié à l'aide d'une expression régulière. Notez que l'expression régulière s'applique uniquement au nom du fichier et n'a pas besoin de correspondre au nom entier (par exemple, `/path/to/agent` correspondra à `zabbix_agentd.log`).
- **regexp** - une expression régulière décrivant le motif requis.
- **encoding** - l'identifiant de page de code.
- **maxproclines** - le nombre maximal de nouvelles lignes par seconde que l'agent analysera (ne peut pas dépasser 10000). La valeur par défaut est 10*`MaxLinesPerSecond` dans `zabbix_agentd.conf`.
- **mode** - valeurs possibles:
  - *all* (par défaut);
  - *skip* - ignore le traitement des données plus anciennes (n'affecte que les éléments nouvellement créés).
- **maxdelay** - le délai maximal en secondes. Type: float. Valeurs:
  - 0 - (par défaut) n'ignore jamais les lignes du fichier journal;
  - > 0.0 - ignore les lignes plus anciennes afin que les lignes les plus récentes soient analysées dans un délai de "maxdelay" secondes. Lisez les notes sur `maxdelay` avant de l'utiliser!
- **options** - le type de rotation du fichier journal et d'autres options. Valeurs possibles:
  - *rotate* (par défaut).
  - *copytruncate* - notez que `copytruncate` ne peut pas être utilisé avec `maxdelay`. Dans ce cas, `maxdelay` doit être 0 ou non spécifié; voir les notes sur `copytruncate`.
  - *mtime-reread* - enregistrements non uniques, relus si l'heure de modification ou la taille change (par défaut).
  - *mtime-noreread* - enregistrements non uniques, relus uniquement si la taille change (ignorer le changement de l'heure de modification).
- **persistent dir** (uniquement dans `zabbix_agentd` sur les systèmes Unix; non pris en charge dans Zabbix agent 2) - le chemin absolu du répertoire où stocker les fichiers persistants. Voir aussi les notes supplémentaires sur les fichiers persistants.

Commentaires:

- L'élément doit être configuré comme une vérification active.
- Les lignes correspondantes sont comptées dans les nouvelles lignes depuis la dernière vérification du journal par l'agent, et dépendent donc de l'intervalle de mise à jour de l'élément.
- La rotation des journaux est basée sur la dernière heure de modification des fichiers.
- Dans le paramètre `file regexp`, le chemin du répertoire des journaux et l'expression régulière du fichier journal doivent être séparés par le bon séparateur de répertoire:
  - Sous Windows, le séparateur doit être une barre oblique inverse (\); les barres obliques peuvent être tolérées à d'autres positions, sauf celle qui sépare le chemin du répertoire des journaux et l'expression régulière du fichier journal (voir les exemples ci-dessous).
  - Sur les autres systèmes, il doit s'agir d'une barre oblique (/).

Exemples pour Windows:

```
logrt.count["c:/dir1/dir2/dir3\filename.*\.log","pattern_to_match"] #cet élément comptera le nombre de cor
logrt.count["//example.com/share/dir1/dir2/dir3\filename.*\.log","pattern_to_match"] #cet élément comptera
```

Exemples pour les autres systèmes:

```
logrt.count["/home/zabbix/logs/.*","^logfile[0-9]{1,3}$",,100] #cet élément comptera le nombre de correspo
logrt.count["/home/user/^logfile_.*_[0-9]{1,3}$","pattern_to_match","UTF-8",100] #cet élément comptera le
```

modbus.get[endpoint,<slave id>,<function>,<address>,<count>,<type>,<endianness>,<offset>]

 Lit les données Modbus.
 Valeur de retour: *chaîne JSON*.
 Plateformes prises en charge: Linux, Windows.

Paramètres:

- **endpoint** - le point de terminaison défini comme `protocol://connection_string`;
- **slave id** - l'ID esclave;
- **function** - la fonction Modbus;
- **address** - l'adresse du premier registre, coil ou input;
- **count** - le nombre d'enregistrements à lire;
- **type** - le type de données;
- **endianness** - la configuration de l'ordre des octets;
- **offset** - le nombre de registres, à partir de `address`, dont les résultats seront ignorés.

Sous Windows, cet élément suit les chemins UNC dans le paramètre `endpoint`, ce qui peut présenter un risque de sécurité.

Voir une description détaillée des paramètres.

net.dns[<ip>,name,<type>,<timeout>,<count>,<protocol>]

 Vérifie l'état d'un service DNS.
 Valeurs de retour : - 0 - la résolution DNS a échoué (le serveur DNS n'a pas répondu ou a renvoyé une erreur) ; - 1 - la résolution DNS a réussi.
 Voir les plates-formes prises en charge.

Paramètres :

- **ip** (ignoré sous Windows sauf si vous utilisez Zabbix agent 2) - l'adresse IP du serveur DNS (laissez vide pour utiliser le serveur DNS par défaut) ;
- **name** - le nom DNS à interroger ;
- **type** - le type d'enregistrement à interroger (la valeur par défaut est `SOA`) ;
- **timeout** (ignoré sous Windows sauf si vous utilisez Zabbix agent 2) - le délai d'attente de la requête, en secondes (la valeur par défaut est 1 seconde) ;
- **count** (ignoré sous Windows sauf si vous utilisez Zabbix agent 2) - le nombre de tentatives pour la requête (la valeur par défaut est 2) ;
- **protocol** - le protocole utilisé pour effectuer les requêtes DNS : *udp* (par défaut) ou *tcp*.

Commentaires :

- Les valeurs possibles pour `type` sont : *ANY*, *A*, *NS*, *CNAME*, *MB*, *MG*, *MR*, *PTR*, *MD*, *MF*, *MX*, *SOA*, *NULL*, *WKS* (non pris en charge pour Zabbix agent sous Windows, Zabbix agent 2 sur tous les OS), *HINFO*, *MINFO*, *TXT*, *SRV*.
- Pour les recherches DNS inversées (lorsque `type` est défini sur *PTR*), vous pouvez fournir le nom DNS au format inversé ou non inversé (voir les exemples ci-dessous). Notez que lorsqu'un enregistrement PTR est demandé, le nom DNS est en réalité une adresse IP.
- Les noms de domaine internationalisés ne sont pas pris en charge ; veuillez utiliser à la place des noms encodés en IDNA.

Exemples :

```
net.dns[198.51.100.1,example.com,MX,2,1]

net.dns[,198.51.100.1,PTR]
net.dns[,1.100.51.198.in-addr.arpa,PTR]

net.dns[,2a00:1450:400f:800::200e,PTR]
net.dns[,e.0.0.2.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.8.0.f.0.0.4.0.5.4.1.0.0.a.2.ip6.arpa,PTR]
```

net.dns.perf[<ip>,name,<type>,<timeout>,<count>,<protocol>]

 Vérifie les performances d'un service DNS.
 Valeur de retour: *Float* (0 - le service est arrêté ; secondes - nombre de secondes passées à attendre une réponse du service).
 Voir les plateformes prises en charge.

Paramètres:

- **ip** (ignoré sous Windows sauf si vous utilisez Zabbix agent 2) - l'adresse IP du serveur DNS (laissez vide pour utiliser le serveur DNS par défaut) ;

- **name** - le nom DNS à interroger ;
- **type** - le type d'enregistrement à interroger (la valeur par défaut est SOA) ;
- **timeout** (ignoré sous Windows sauf si vous utilisez Zabbix agent 2) - le délai d'attente de la requête, en secondes (la valeur par défaut est 1 seconde) ;
- **count** (ignoré sous Windows sauf si vous utilisez Zabbix agent 2) - le nombre de tentatives pour la requête (la valeur par défaut est 2) ;
- **protocol** - le protocole utilisé pour effectuer les requêtes DNS: *udp* (par défaut) ou *tcp*.

Commentaires:

- Les valeurs possibles pour type sont:
ANY, *A*, *NS*, *CNAME*, *MB*, *MG*, *MR*, *PTR*, *MD*, *MF*, *MX*, *SOA*, *NULL*, *WKS* (non pris en charge pour Zabbix agent sous Windows, Zabbix agent 2 sur tous les OS), *HINFO*, *MINFO*, *TXT*, *SRV*.
- Pour les recherches DNS inversées (lorsque type est défini sur *PTR*), vous pouvez fournir le nom DNS au format inversé ou non inversé (voir les exemples ci-dessous). Notez que lorsqu'un enregistrement PTR est demandé, le nom DNS est en réalité une adresse IP.
- Les noms de domaine internationalisés ne sont pas pris en charge, veuillez utiliser à la place des noms encodés en IDNA.
- L'élément renvoie un temps de réponse au lieu de 0 lorsque le serveur DNS répond avec un code d'erreur (par exemple, NXDOMAIN ou SERVFAIL).

Exemples:

```
net.dns.perf[198.51.100.1,example.com,MX,2,1]

net.dns.perf[,198.51.100.1,PTR]
net.dns.perf[,1.100.51.198.in-addr.arpa,PTR]

net.dns.perf[,2a00:1450:400f:800::200e,PTR]
net.dns.perf[,e.0.0.2.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.8.0.f.0.0.4.0.5.4.1.0.0.a.2.ip6.arpa,PTR]
```

net.dns.record[<ip>,name,<type>,<timeout>,<count>,<protocol>]

 Effectue une requête DNS.

Zabbix agent 2 fournit également net.dns.get, qui offre des fonctionnalités supplémentaires, telles que davantage de types d'enregistrements et un meilleur contrôle de la surveillance DNS. Valeur de retour: une chaîne de caractères contenant le type d'information requis.
 Voir les plates-formes prises en charge.

Paramètres:

- **ip** (ignoré sous Windows sauf si vous utilisez Zabbix agent 2) - l'adresse IP du serveur DNS (laissez vide pour utiliser le serveur DNS par défaut);
- **name** - le nom DNS à interroger;
- **type** - le type d'enregistrement à interroger (la valeur par défaut est SOA);
- **timeout** (ignoré sous Windows sauf si vous utilisez Zabbix agent 2) - le délai d'attente de la requête, en secondes (la valeur par défaut est 1 seconde);
- **count** (ignoré sous Windows sauf si vous utilisez Zabbix agent 2) - le nombre de tentatives pour la requête (la valeur par défaut est 2);
- **protocol** - le protocole utilisé pour effectuer les requêtes DNS: *udp* (par défaut) ou *tcp*.

Commentaires:

- Les valeurs possibles pour type sont:
ANY, *A*, *NS*, *CNAME*, *MB*, *MG*, *MR*, *PTR*, *MD*, *MF*, *MX*, *SOA*, *NULL*, *WKS* (non pris en charge pour Zabbix agent sous Windows, Zabbix agent 2 sur tous les OS), *HINFO*, *MINFO*, *TXT*, *SRV*.
- Pour les recherches DNS inversées (lorsque type est défini sur *PTR*), vous pouvez fournir le nom DNS au format inversé ou non inversé (voir les exemples ci-dessous). Notez que lorsque l'enregistrement PTR est demandé, le nom DNS est en réalité une adresse IP.
- Les noms de domaine internationalisés ne sont pas pris en charge; veuillez utiliser à la place des noms encodés en IDNA.

Exemples:

```
net.dns.record[198.51.100.1,example.com,MX,2,1]

net.dns.record[,198.51.100.1,PTR]
net.dns.record[,1.100.51.198.in-addr.arpa,PTR]

net.dns.record[,2a00:1450:400f:800::200e,PTR]
net.dns.record[,e.0.0.2.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.8.0.f.0.0.4.0.5.4.1.0.0.a.2.ip6.arpa,PTR]
```

net.if.collisions[if]

 Le nombre de collisions hors fenêtre.
 Valeur de retour : *Integer*.
 Plateformes prises en charge : Linux, FreeBSD, Solaris, AIX, MacOS X, OpenBSD, NetBSD. Les privilèges root sont requis sur NetBSD.

Paramètre :

- **if** - nom de l'interface réseau

net.if.discovery

 La liste des interfaces réseau. Utilisé pour la découverte de bas niveau.
 Valeur de retour : *chaîne JSON*.
 Plateformes prises en charge : Linux, FreeBSD, Solaris, HP-UX, AIX, OpenBSD, NetBSD, Windows.

net.if.get[<regexp>]

 La liste des interfaces réseau avec des informations détaillées.
 Utilisé pour la découverte de bas niveau.
 Cet élément est une version étendue de l'élément agent Zabbix `net.if.discovery`.
 Valeur de retour : *objet JSON*.
 Plateformes prises en charge : Linux.

Paramètres :

- **regexp** - une expression régulière correspondant aux noms des interfaces réseau.

Commentaires :

- Renvoie un objet JSON contenant les interfaces réseau dont les noms correspondent à l'expression régulière donnée (voir plus de détails).

net.if.in[if,<mode>]

 Les statistiques du trafic entrant sur une interface réseau.
 Valeur de retour: *Integer*.
 Plateformes prises en charge: Linux, FreeBSD, Solaris[5], HP-UX, AIX, MacOS X, OpenBSD, NetBSD, Windows. Les privilèges root sont requis sur NetBSD.

Paramètres:

- **if** - nom de l'interface réseau (Unix); description complète de l'interface réseau ou adresse IPv4; ou, si entre accolades, GUID de l'interface réseau (Windows);
- **mode** - valeurs possibles:
  - *bytes* - nombre d'octets (par défaut);
  - *packets* - nombre de paquets;
  - *errors* - nombre d'erreurs;
  - *dropped* - nombre de paquets abandonnés;
  - *overruns (fifo)* - nombre d'erreurs de tampon FIFO;
  - *frame* - nombre d'erreurs de tramage des paquets;
  - *compressed* - nombre de paquets compressés reçus par le pilote de périphérique;
  - *multicast* - nombre de trames multicast reçues par le pilote de périphérique.

Commentaires:

- Vous pouvez utiliser cette clé avec l'étape de prétraitement *Change per second* afin d'obtenir les statistiques en octets par seconde.
- Le mode *dropped* est pris en charge uniquement sur Linux, FreeBSD, HP-UX, MacOS X, OpenBSD, NetBSD.
- Les modes *overruns*, *frame*, *compressed*, *multicast* sont pris en charge uniquement sur Linux.
- Sur HP-UX, cet élément ne fournit pas de détails sur les interfaces loopback (par exemple lo0).

Exemples:

```
net.if.in[eth0]
net.if.in[eth0,errors]
```

net.if.out[if,<mode>]

 Les statistiques du trafic sortant sur une interface réseau.
 Valeur de retour: *Integer*.
 Plateformes prises en charge: Linux, FreeBSD, Solaris[5], HP-UX, AIX, MacOS X, OpenBSD, NetBSD, Windows. Les privilèges root sont requis sur NetBSD.

Paramètres:

- **if** - nom de l'interface réseau (Unix); description complète de l'interface réseau ou adresse IPv4; ou, si entre accolades, GUID de l'interface réseau (Windows);
- **mode** - valeurs possibles:
  - *bytes* - nombre d'octets (par défaut);
  - *packets* - nombre de paquets;
  - *errors* - nombre d'erreurs;
  - *dropped* - nombre de paquets abandonnés;
  - *overruns (fifo)* - nombre d'erreurs de tampon FIFO;

  - *collisions (colls)* - nombre de collisions détectées sur l'interface;
  - *carrier* - nombre de pertes de porteuse détectées par le pilote de périphérique;
  - *compressed* - nombre de paquets compressés transmis par le pilote de périphérique.

Commentaires:

- Vous pouvez utiliser cette clé avec l'étape de prétraitement *Change per second* afin d'obtenir les statistiques en octets par seconde.
- Le mode *dropped* est pris en charge uniquement sur Linux, HP-UX.
- Les modes *overruns*, *collision*, *carrier*, *compressed* sont pris en charge uniquement sur Linux.
- Sur HP-UX, cet élément ne fournit pas de détails sur les interfaces de bouclage (par exemple lo0).

Exemples:

```
net.if.out[eth0]
net.if.out[eth0,errors]
```

net.if.total[if,<mode>]

 La somme des statistiques du trafic entrant et sortant sur une interface réseau.
 Valeur de retour : *Integer*.
 Plateformes prises en charge : Linux, FreeBSD, Solaris[5], HP-UX, AIX, MacOS X, OpenBSD, NetBSD, Windows. Les privilèges root sont requis sur NetBSD.

Paramètres :

- **if** - nom de l'interface réseau (Unix) ; description complète de l'interface réseau ou adresse IPv4 ; ou, si entre accolades, GUID de l'interface réseau (Windows) ;
- **mode** - valeurs possibles :
  - *bytes* - nombre d'octets (par défaut) ;
  - *packets* - nombre de paquets ;
  - *errors* - nombre d'erreurs ;
  - *dropped* - nombre de paquets abandonnés ;
  - *overruns (fifo)* - nombre d'erreurs de tampon FIFO ;
  - *collisions (colls)* - nombre de collisions détectées sur l'interface ;
  - *compressed* - nombre de paquets compressés transmis ou reçus par le pilote de périphérique.

Commentaires :

- Vous pouvez utiliser cette clé avec l'étape de prétraitement *Change per second* afin d'obtenir les statistiques en octets par seconde.
- Le mode *dropped* est pris en charge uniquement sur Linux, HP-UX. Les paquets abandonnés sont pris en charge uniquement si `net.if.in` et `net.if.out` fonctionnent tous deux pour les paquets abandonnés sur votre plateforme.
- Les modes *overruns*, *collision*, *compressed* sont pris en charge uniquement sur Linux.
- Sur HP-UX, cet élément ne fournit pas de détails sur les interfaces loopback (par ex. lo0).

Exemples :

```
net.if.total[eth0]
net.if.total[eth0,errors]
```

net.tcp.listen[port]

 Vérifie si ce port TCP est à l'état LISTEN.
 Valeurs de retour: 0 - il n'est pas à l'état LISTEN; 1 - il est à l'état LISTEN.
 Plateformes prises en charge: Linux, FreeBSD, Solaris, MacOS X, Windows.

Paramètre:

- **port** - numéro de port TCP

Sur les noyaux Linux 2.6.14 et versions ultérieures, les informations sur les sockets TCP en écoute sont obtenues à partir de l'interface NETLINK du noyau, si possible. Sinon, les informations sont récupérées à partir des fichiers `/proc/net/tcp` et `/roc/net/tcp6`.

Exemple:

```
net.tcp.listen[80]
```

net.tcp.port[<ip>,port]

 Vérifie s'il est possible d'établir une connexion TCP vers le port spécifié.
 Valeurs de retour : 0 - impossible de se connecter ; 1 - connexion possible.
 Voir les plateformes prises en charge.

Paramètres :

- **ip** - l'adresse IP ou le nom DNS (la valeur par défaut est `127.0.0.1`) ;
- **port** - le numéro de port.

Commentaires :

- Pour un test simple des performances TCP, utilisez `net.tcp.service.perf[tcp,<ip>,<port>]`.
- Ces vérifications peuvent entraîner des messages supplémentaires dans les fichiers journaux du démon système (les sessions SMTP et SSH sont généralement consignées).

Exemple :

```
net.tcp.port[,80] #cet élément peut être utilisé pour tester la disponibilité du serveur web exécuté sur l
```

net.tcp.service[service,<ip>,<port>]

 Vérifie si un service est en cours d'exécution et accepte des connexions TCP.
 Valeurs de retour : 0 - le service est arrêté ; 1 - le service est en cours d'exécution.
 Voir les plates-formes prises en charge.

Paramètres :

- **service** - *ssh*, *ldap*, *smtp*, *ftp*, *http*, *pop*, *nntp*, *imap*, *tcp*, *https* ou *telnet* (voir les détails) ;
- **ip** - l'adresse IP ou le nom DNS (par défaut `127.0.0.1`) ;
- **port** - le numéro de port (par défaut, le numéro de port standard du service est utilisé).

Commentaires :

- Ces vérifications peuvent entraîner des messages supplémentaires dans les fichiers journaux des démons système (les sessions SMTP et SSH sont généralement consignées).
- La vérification des protocoles chiffrés (comme IMAP sur le port 993 ou POP sur le port 995) n'est actuellement pas prise en charge. Comme solution de contournement, veuillez utiliser `net.tcp.port[]` pour ce type de vérifications.
- La vérification de LDAP et HTTPS sous Windows n'est prise en charge que par Zabbix agent 2.
- La vérification telnet recherche une invite de connexion (':' à la fin).

Exemple :

```
net.tcp.service[ftp,,45] #this item can be used to test the availability of FTP server on TCP port 45
```

net.tcp.service.perf[service,<ip>,<port>]

 Vérifie les performances d'un service TCP.
 Valeurs de retour: *Float* (0 - le service est hors ligne; secondes - nombre de secondes passées à attendre une réponse du service).
 Voir les plates-formes prises en charge.

Paramètres:

- **service** - *ssh*, *ldap*, *smtp*, *ftp*, *http*, *pop*, *nntp*, *imap*, *tcp*, *https*, ou *telnet* (voir les détails);
- **ip** - l'adresse IP ou le nom DNS (par défaut `127.0.0.1`);
- **port** - le numéro de port (par défaut, le numéro de port standard du service est utilisé).

Commentaires:

- La vérification des protocoles chiffrés (comme IMAP sur le port 993 ou POP sur le port 995) n'est actuellement pas prise en charge. Comme solution de contournement, veuillez utiliser `net.tcp.service.perf[tcp,<ip>,<port>]` pour ce type de vérification.
- La vérification telnet recherche une invite de connexion (':' à la fin).

Exemple:

```
net.tcp.service.perf[ssh] #cet élément peut être utilisé pour tester la vitesse de la réponse initiale du
```

net.tcp.socket.count[<laddr>,<lport>,<raddr>,<rport>,<state>]

 Renvoie le nombre de sockets TCP correspondant aux paramètres.
 Valeur de retour : *Integer*.
 Plateformes prises en charge : Linux, Windows.

Paramètres :

- **laddr** - l'adresse IPv4/6 locale ou le sous-réseau CIDR ;
- **lport** - le numéro de port local ou le nom du service ;
- **raddr** - l'adresse IPv4/6 distante ou le sous-réseau CIDR ;
- **rport** - le numéro de port distant ou le nom du service ;
- **state** - l'état de la connexion (*established*, *syn_sent*, *syn_recv*, *fin_wait1*, *fin_wait2*, *time_wait*, *close*, *close_wait*, *last_ack*, *listen*, *closing*).

Exemple :

```
net.tcp.socket.count[,80,,,established] #le nombre de connexions vers le port TCP local 80 à l'état establi
```

net.udp.listen[port]

 Vérifie si ce port UDP est à l'état LISTEN.
 Valeurs de retour : 0 - il n'est pas à l'état LISTEN ; 1 - il est à l'état LISTEN.
 Plateformes prises en charge : Linux, FreeBSD, Solaris, MacOS X.

Paramètre :

- **port** - numéro de port UDP

Exemple :

```
net.udp.listen[68]
```

net.udp.service[service,<ip>,<port>]

 Vérifie si un service est en cours d'exécution et répond aux requêtes UDP.
 Valeurs de retour : 0 - le service est arrêté ; 1 - le service est en cours d'exécution.
 Voir les plates-formes prises en charge.

Paramètres :

- **service** - *ntp* (voir détails) ;
- **ip** - l'adresse IP ou le nom DNS (par défaut `127.0.0.1`) ;
- **port** - le numéro de port (par défaut, le numéro de port standard du service est utilisé).

Exemple :

```
net.udp.service[ntp,,45] #cet élément peut être utilisé pour tester la disponibilité du service NTP sur le
```

net.udp.service.perf[service,<ip>,<port>]

 Vérifie les performances d'un service UDP.
 Valeurs de retour: *Float* (0 - le service est arrêté ; secondes - le nombre de secondes passées à attendre une réponse du service).
 Voir les plates-formes prises en charge.

Paramètres:

- **service** - *ntp* (voir les détails);
- **ip** - l'adresse IP ou le nom DNS (par défaut, `127.0.0.1` est utilisé) ;
- **port** - le numéro de port (par défaut, le numéro de port standard du service est utilisé).

Exemple:

```
net.udp.service.perf[ntp] #this item can be used to test response time from NTP service
```

net.udp.socket.count[<laddr>,<lport>,<raddr>,<rport>,<state>]

 Renvoie le nombre de sockets UDP correspondant aux paramètres.
 Valeur de retour : *Integer*.
 Plateformes prises en charge : Linux, Windows.

Paramètres :

- **laddr** - l'adresse IPv4/6 locale ou le sous-réseau CIDR ;
- **lport** - le numéro de port local ou le nom du service ;
- **raddr** - l'adresse IPv4/6 distante ou le sous-réseau CIDR ;
- **rport** - le numéro de port distant ou le nom du service ;
- **state** - l'état de la connexion (*established*, *unconn*).

Exemple :

```
net.udp.socket.count[,,,,established] #renvoie le nombre de sockets UDP à l'état connecté
```

proc.cpu.util[<name>,<user>,<type>,<cmdline>,<mode>,<zone>]

 Le pourcentage d'utilisation CPU du processus.
 Valeur de retour: *Float*.
 Plateformes prises en charge: Linux, Solaris[6].

Paramètres:

- **name** - le nom du processus (la valeur par défaut est *all processes*);
- **user** - le nom de l'utilisateur (la valeur par défaut est *all users*);
- **type** - le type d'utilisation CPU: *total* (par défaut), *user* ou *system*;
- **cmdline** - filtrer par ligne de commande (il s'agit d'une expression régulière);
- **mode** - le mode de collecte des données: *avg1* (par défaut), *avg5* ou *avg15*;
- **zone** - la zone cible: *current* (par défaut) ou *all*. Ce paramètre est pris en charge uniquement sur Solaris.

Commentaires:

- La valeur retournée est basée sur le pourcentage d'utilisation d'un seul coeur CPU. Par exemple, l'utilisation CPU d'un processus utilisant pleinement deux coeurs est de 200%.
- Les données d'utilisation CPU du processus sont collectées par un collecteur qui prend en charge un maximum de 1024 requêtes uniques (par nom, utilisateur et ligne de commande). Les requêtes non consultées au cours des dernières 24 heures sont supprimées du collecteur.
- Lors de la définition du paramètre `zone` sur *current* (ou par défaut), si l'agent a été compilé sur un Solaris sans prise en charge des zones, mais s'exécute sur une version plus récente de Solaris où les zones sont prises en charge, alors l'agent renverra NOTSUPPORTED (l'agent ne peut pas limiter les résultats à la zone actuelle uniquement). Cependant, *all* est pris en charge dans ce cas.

Exemples:

```
proc.cpu.util[,root] #utilisation CPU de tous les processus exécutés sous l'utilisateur "root"
proc.cpu.util[zabbix_server,zabbix] #utilisation CPU de tous les processus zabbix_server exécutés sous l'u
```

proc.get[<name>,<user>,<cmdline>,<mode>]

 La liste des processus du système d'exploitation et de leurs paramètres. Peut être utilisée pour la découverte de bas niveau.
 Valeur de retour: *chaîne JSON*.
 Plateformes prises en charge: Linux, FreeBSD, Windows, OpenBSD, NetBSD.

Paramètres:

- **name** - le nom du processus (par défaut *tous les processus*).
- **user** - le nom de l'utilisateur (par défaut *tous les utilisateurs*).
- **cmdline** - filtre par ligne de commande (c'est une expression régulière). Ce paramètre n'est pas pris en charge pour Windows; sur les autres plateformes, il n'est pas pris en charge si mode est défini sur 'summary'.
- **mode** - valeurs possibles:
  - *process* (par défaut);
  - *thread* (non pris en charge pour NetBSD);
  - *summary*. Voir une liste des paramètres de processus renvoyés pour chaque mode et système d'exploitation.

Commentaires:

- Si une valeur ne peut pas être récupérée, par exemple en raison d'une erreur (le processus est déjà terminé, absence d'autorisations, échec d'un appel système), −1 sera renvoyé.
- Voir les notes sur la sélection des processus avec les paramètres `name` et `cmdline` (spécifique à Linux).

Exemples:

```
proc.get[zabbix_server,zabbix,,process] #liste de tous les processus zabbix_server exécutés sous l'utilisa
proc.get[java,,,thread] #liste de tous les processus Java, renvoie une entrée par thread
proc.get[,zabbix,,summary] #données combinées pour les processus de chaque type exécutés sous l'utilisateu
```

proc.mem[<name>,<user>,<mode>,<cmdline>,<memtype>]

 La mémoire utilisée par le processus, en octets.
 Valeur de retour : - *Integer* - avec `mode` défini sur *max*, *min*, *sum* ; - *Float* - avec `mode` défini sur *avg*
 Plateformes prises en charge : Linux, FreeBSD, Solaris, AIX, OpenBSD, NetBSD.

Paramètres :

- **name** - le nom du processus (par défaut : *all processes*) ;
- **user** - le nom d'utilisateur (par défaut : *all users*) ;
- **mode** - valeurs possibles : *avg*, *max*, *min* ou *sum* (par défaut) ;
- **cmdline** - filtre par ligne de commande (il s'agit d'une expression régulière) ;
- **memtype** - le type de mémoire utilisé par le processus

Commentaires :

- Le paramètre `memtype` est pris en charge uniquement sur Linux, FreeBSD, Solaris[6], AIX.
- Lorsque plusieurs processus utilisent de la mémoire partagée, la somme de la mémoire utilisée par les processus peut produire des valeurs élevées et irréalistes.

Voir les notes sur la sélection des processus avec les paramètres `name` et `cmdline` (spécifique à Linux).

Lorsque cet élément est invoqué depuis la ligne de commande et contient un paramètre de ligne de commande (par exemple, en utilisant le mode de test de l'agent : `zabbix_agentd -t proc.mem[,,,apache2]`), un processus supplémentaire sera compté, car l'agent se comptera lui-même.

Exemples :

```
proc.mem[,root] #la mémoire utilisée par tous les processus exécutés sous l'utilisateur "root"
proc.mem[zabbix_server,zabbix] #la mémoire utilisée par tous les processus zabbix_server exécutés sous l'u
proc.mem[,oracle,max,oracleZABBIX] #la mémoire utilisée par le processus le plus gourmand en mémoire exécu
```

proc.num[<name>,<user>,<state>,<cmdline>,<zone>]

 Le nombre de processus.
 Valeur de retour: *Integer*.
 Plateformes prises en charge: Linux, FreeBSD, Solaris[6], HP-UX, AIX, OpenBSD, NetBSD, Windows.

Paramètres:

- **name** - le nom du processus (la valeur par défaut est *all processes*).
- **user** - le nom d'utilisateur (la valeur par défaut est *all users*).
- **state** - valeurs possibles:
  - *all* (par défaut);
  - *disk* - sommeil ininterruptible;
  - *run* - en cours d'exécution;
  - *sleep* - sommeil interruptible;
  - *trace* - arrêté;
  - *zomb* - zombie.
- **cmdline** - filtre par ligne de commande (c'est une expression régulière).
- **zone** - la zone cible: *current* (par défaut), ou *all*. Ce paramètre est pris en charge uniquement sur Solaris.

Commentaires:

- Les paramètres d'état *disk* et *trace* sont pris en charge uniquement sur Linux, FreeBSD, OpenBSD, NetBSD.
- Sous Windows, seuls les paramètres `name` et `user` sont pris en charge.
- Lorsque cet élément est appelé depuis la ligne de commande et contient un paramètre de ligne de commande (par exemple, en utilisant le mode de test de l'agent: `zabbix_agentd -t proc.num[,,,apache2]`), un processus supplémentaire sera compté, car l'agent se comptera lui-même.
- Lors de la définition du paramètre `zone` sur `current` (ou par défaut), si l'agent a été compilé sur un Solaris sans prise en charge des zones, mais s'exécute sur une version plus récente de Solaris où les zones sont prises en charge, alors l'agent renverra NOTSUPPORTED (l'agent ne peut pas limiter les résultats à la zone actuelle uniquement). Cependant, *all* est pris en charge dans ce cas.
- Voir les notes sur la sélection des processus avec les paramètres `name` et `cmdline` (spécifique à Linux).

Exemples:

```
proc.num[,mysql] #le nombre de processus exécutés sous l'utilisateur mysql
proc.num[apache2,www-data] #le nombre de processus apache2 exécutés sous l'utilisateur www-data
proc.num[,oracle,sleep,oracleZABBIX] #le nombre de processus à l'état sleep exécutés sous Oracle et ayant
```

sensor[device,sensor,<mode>]

 Lecture d'un capteur matériel.
 Valeur de retour: *Float*.
 Plateformes prises en charge: Linux, OpenBSD.

Paramètres:

- **device** - le nom du périphérique;
- **sensor** - le nom du capteur;
- **mode** - valeurs possibles: *avg*, *max* ou *min*.

Commentaires:

- Lit `/proc/sys/dev/sensors` sur Linux 2.4.
- Lit `/sys/class/hwmon` sur Linux 2.6+. Voir une description plus détaillée de l'élément sensor sur Linux.
- Lit le MIB `hw.sensors` sur OpenBSD.
- Sur Linux 2.4, si vous spécifiez mode, alors device et sensor sont traités comme des expressions régulières.
- Sur Linux 2.6+ et OpenBSD, ou si mode est omis, device et sensor sont traités littéralement.

Exemples:

```
sensor[w83781d-i2c-0-2d,temp1]
sensor[cpu0,temp0] #la température d'un CPU
sensor["cpu[0-2]$",temp,avg] #la température moyenne des trois premiers CPU
```

system.boottime

 L'heure de démarrage du système.
 Valeur de retour : *Integer (horodatage Unix)*.
 Plateformes prises en charge : Linux, FreeBSD, Solaris, MacOS X, OpenBSD, NetBSD.

system.cpu.discovery

 La liste des CPU/cœurs de CPU détectés. Utilisé pour la découverte de bas niveau.
 Valeur de retour : *chaîne JSON*.
 Voir les plateformes prises en charge.

system.cpu.intr

 Les interruptions du périphérique.
 Valeur de retour : *Integer*.
 Plateformes prises en charge : Linux, FreeBSD, Solaris, AIX, OpenBSD, NetBSD.

system.cpu.load[<cpu>,<mode>]

 La charge CPU.
 Valeur de retour : *Float*.
 Voir les plateformes prises en charge.

Paramètres :

- **cpu** - valeurs possibles : *all* (par défaut) ou *percpu* (la charge totale divisée par le nombre de CPU en ligne) ;
- **mode** - valeurs possibles : *avg1* (moyenne sur une minute, par défaut), *avg5* ou *avg15*.

Exemple :

```
system.cpu.load[,avg5]
```

system.cpu.num[<type>]

 Le nombre de processeurs.
 Valeur de retour : *Integer*.
 Plateformes prises en charge : Linux, FreeBSD, Solaris, HP-UX, AIX, MacOS X, OpenBSD, NetBSD, Windows.

Paramètre :

- **type** - valeurs possibles : *online* (par défaut) ou *max*

Le paramètre de type *max* est pris en charge uniquement sous Linux, FreeBSD, Solaris, MacOS X.

Exemple :

```
system.cpu.num
```

system.cpu.switches

 Le nombre de changements de contexte.
 Valeur de retour : *Integer*.
 Plateformes prises en charge : Linux, FreeBSD, Solaris, AIX, OpenBSD, NetBSD.

system.cpu.util[<cpu>,<type>,<mode>,<logical or physical>]

 Le pourcentage d'utilisation du processeur.
 Valeur de retour : *Float*.
 Plateformes prises en charge : Linux, FreeBSD, Solaris, HP-UX, AIX, OpenBSD, NetBSD, Windows.

Paramètres :

- **cpu** - *<numéro de CPU>* ou *all* (par défaut) ;
- **type** - valeurs possibles : *user* (par défaut), *idle*, *nice*, *system*, *iowait*, *interrupt*, *softirq*, *steal*, *spin* (sur OpenBSD), *guest* (sur les noyaux Linux 2.6.24 et supérieurs), ou *guest_nice* (sur les noyaux Linux 2.6.33 et supérieurs) ;
- **mode** - valeurs possibles : *avg1* (moyenne sur une minute, par défaut), *avg5* ou *avg15* ;
- **logical or physical** - valeurs possibles : *logical* (par défaut) ou *physical*. Ce paramètre est pris en charge uniquement sur AIX.

Commentaires :

- Le paramètre de type *nice* est pris en charge uniquement sur Linux, FreeBSD, HP-UX, OpenBSD, NetBSD.
- Le paramètre de type *iowait* est pris en charge uniquement sur Linux 2.6 et versions ultérieures, Solaris, AIX.
- Le paramètre de type *interrupt* est pris en charge uniquement sur Linux 2.6 et versions ultérieures, FreeBSD, OpenBSD.
- Les paramètres de type *softirq*, *steal*, *guest*, *guest_nice* sont pris en charge uniquement sur Linux 2.6 et versions ultérieures.
- Les paramètres de mode *avg5* et *avg15* sont pris en charge sur Linux, FreeBSD, Solaris, HP-UX, AIX, OpenBSD, NetBSD.

Exemple :

```
system.cpu.util[0,user,avg5]
```

system.hostname[<type>,<transform>]

 Le nom d'hôte du système.
 Valeur de retour: *String*.
 Voir les plateformes prises en charge.

Paramètres:

- **type** - valeurs possibles:
  - *netbios* (par défaut sous Windows);
  - *host* (par défaut sous Linux);
  - *shorthost* (renvoie la partie du nom d'hôte avant le premier point, une chaîne complète pour les noms sans point);
  - *fqdn* (renvoie le nom de domaine pleinement qualifié).
- **transform** - valeurs possibles:

- *none* (par défaut);
- *lower* (convertit en minuscules).

La valeur est obtenue en prenant `nodename` à partir de la sortie de l'API système uname().

Exemples de valeurs renvoyées:

```
system.hostname → linux-w7x1
system.hostname → example.com
system.hostname[shorthost] → example
system.hostname → WIN-SERV2008-I6
system.hostname[host] → Win-Serv2008-I6LonG
system.hostname[host,lower] → win-serv2008-i6long
system.hostname[fqdn,lower] → blog.zabbix.com
```

system.hw.chassis[<info>]

 Les informations sur le châssis.
 Valeur de retour : *String*.
 Plateformes prises en charge : Linux.

Paramètre :

- **info** - valeurs possibles : *full* (par défaut), *model*, *serial*, *type* ou *vendor*

Commentaires :

- Cette clé d'élément dépend de la disponibilité de la table SMBIOS.
- Elle tentera de lire la table DMI depuis sysfs ; si l'accès à sysfs échoue, elle essaiera alors de lire directement depuis la mémoire.
- Les **permissions root** sont requises, car la valeur est obtenue par lecture depuis sysfs ou la mémoire.

Exemple :

```
system.hw.chassis[full] → Hewlett-Packard HP Pro 3010 Small Form Factor PC CZXXXXXXXX Desktop
```

system.hw.cpu[<cpu>,<info>]

 Les informations sur le processeur.
 Valeur de retour : *String* ou *Integer*.
 Plateformes prises en charge : Linux.

Paramètres :

- **cpu** - <*numéro de CPU*> ou *all* (par défaut) ;
- **info** - valeurs possibles : *full* (par défaut), *curfreq*, *maxfreq*, *model* ou *vendor*.

Commentaires :

- Récupère les informations depuis `/proc/cpuinfo` et `/sys/devices/system/cpu/[cpunum]/cpufreq/cpuinfo_max_freq`.
- Si un numéro de CPU et *curfreq* ou *maxfreq* sont spécifiés, une valeur numérique est renvoyée (Hz).

Exemple :

```
system.hw.cpu[0,vendor] → AuthenticAMD
```

system.hw.devices[<type>]

 La liste des périphériques PCI ou USB.
 Valeur de retour : *Texte*.
 Plateformes prises en charge : Linux.

Paramètre :

- **type** - *pci* (par défaut) ou *usb*

Renvoie la sortie de l'utilitaire lspci ou lsusb (exécuté sans aucun paramètre).

Exemple :

```
system.hw.devices → 00:00.0 Host bridge: Advanced Micro Devices [AMD] RS780 Host Bridge
```

system.hw.macaddr[<interface>,<format>]

 Liste des adresses MAC.
 Valeur de retour: *String*.
 Plateformes prises en charge: Linux.

Paramètres:

- **interface** - *all* (par défaut) ou une expression régulière;
- **format** - *full* (par défaut) ou *short*.

Commentaires:

- Liste les adresses MAC des interfaces dont le nom correspond à l'expression régulière `interface` donnée (*all* liste toutes les interfaces).
- Si `format` est spécifié comme *short*, les noms d'interface et les adresses MAC identiques ne sont pas listés.

Exemple:

```
system.hw.macaddr["eth0$",full] → [eth0] 00:11:22:33:44:55
```

system.localtime[<type>]

 L'heure du système.
 Valeur de retour: *Integer* - avec `type` défini sur *utc*; *String* - avec `type` défini sur *local*.
 Voir les plates-formes prises en charge.

Paramètres:

- **type** - valeurs possibles:
  - *utc* - (par défaut) le temps écoulé depuis l'Epoch (00:00:00 UTC, 1er janvier 1970), mesuré en secondes;
  - *local* - l'heure au format 'yyyy-mm-dd,hh:mm:ss.nnn,+hh:mm'.

Doit être utilisé comme une vérification passive pour Zabbix agent; sur Zabbix agent 2, il peut être configuré comme une vérification active.

Exemple:

```
system.localtime[local] #créez un élément à l'aide de cette clé, puis utilisez-le pour afficher l'heure de
```

system.run[command,<mode>]

 Exécute la commande spécifiée sur l'hôte.
 Valeur de retour: résultat *Text* de la commande ou 1 - avec `mode` défini sur *nowait* (quel que soit le résultat de la commande).
 Voir les plateformes prises en charge.

Paramètres:

- **command** - commande à exécuter;
- **mode** - valeurs possibles:
  - *wait* - attendre la fin de l'exécution (par défaut);
  - *nowait* - ne pas attendre.

Commentaires:

- Cet élément est désactivé par défaut. Découvrez comment les activer.
- La valeur de retour de l'élément est la sortie standard ainsi que l'erreur standard produites par la commande. La vérification du code de sortie n'est pas effectuée.
- Pour être traité correctement, la valeur de retour de la commande doit être de type de données `text`. Un résultat vide est également autorisé.
- La valeur de retour est limitée à 16 Mo (y compris les espaces blancs de fin qui sont tronqués); les limites de la base de données s'appliquent également.
- Voir aussi: Exécution de commandes.

Exemple:

```
system.run[ls -l /] #retourne une liste détaillée des fichiers du répertoire racine
```

system.stat[resource,<type>]

 Les statistiques système.
 Valeur de retour: *Integer* ou *float*.
 Plateformes prises en charge: AIX.

Paramètres:

- **ent** - le nombre d'unités de processeur que cette partition est autorisée à recevoir (float).
- **kthr,<type>** - informations sur les états des threads du noyau:
  - *r* - nombre moyen de threads du noyau exécutables (float);
  - *b* - nombre moyen de threads du noyau placés dans la file d'attente d'attente du Gestionnaire de mémoire virtuelle (float).
- **memory,<type>** - informations sur l'utilisation de la mémoire virtuelle et réelle:
  - *avm* - pages virtuelles actives (integer);
  - *fre* - taille de la liste libre (integer).
- **page,<type>** - informations sur les défauts de page et l'activité de pagination:
  - *fi* - entrées de pages de fichiers par seconde (float);
  - *fo* - sorties de pages de fichiers par seconde (float);
  - *pi* - pages paginées depuis l'espace de pagination (float);
  - *po* - pages paginées vers l'espace de pagination (float);
  - *fr* - pages libérées (remplacement de pages) (float);

    - *sr* - pages analysées par l'algorithme de remplacement de pages (float).
- **faults,<type>** - taux de traps et d'interruptions:
    - *in* - interruptions de périphériques (float);
    - *sy* - appels système (float);
    - *cs* - changements de contexte des threads du noyau (float).
- **cpu,<type>** - répartition du pourcentage d'utilisation du temps processeur:
    - *us* - temps utilisateur (float);
    - *sy* - temps système (float);
    - *id* - temps d'inactivité (float);
    - *wa* - temps d'inactivité pendant lequel le système avait des requêtes d'E/S disque/NFS en attente (float);
    - *pc* - nombre de processeurs physiques consommés (float);
    - *ec* - pourcentage de capacité autorisée consommée (float);
    - *lbusy* - indique le pourcentage d'utilisation du ou des processeurs logiques survenu lors de l'exécution au niveau utilisateur et système (float);
    - *app* - indique les processeurs physiques disponibles dans le pool partagé (float).
- **disk,<type>** - statistiques de disque:
    - *bps* - indique la quantité de données transférées (lues ou écrites) vers le lecteur, en octets par seconde (integer);
    - *tps* - indique le nombre de transferts par seconde envoyés au disque physique/bande (float).

Commentaires:

- Veuillez noter les limitations suivantes pour ces éléments:
    - `system.stat[cpu,app]` - pris en charge uniquement sur les LPAR AIX de type "Shared";
    - `system.stat[cpu,ec]` - pris en charge sur les LPAR AIX de type "Shared" et "Dedicated" ("Dedicated" renvoie toujours 100 (pour cent));
    - `system.stat[cpu,lbusy]` - pris en charge uniquement sur les LPAR AIX de type "Shared";
    - `system.stat[cpu,pc]` - pris en charge sur les LPAR AIX de type "Shared" et "Dedicated";
    - `system.stat[ent]` - pris en charge sur les LPAR AIX de type "Shared" et "Dedicated".

system.sw.arch

 Les informations sur l'architecture logicielle.
 Valeur de retour : *String*.
 Voir les plateformes prises en charge.

Les informations sont obtenues à partir de la fonction `uname()`.

Exemple :

```
system.sw.arch → i686
```

system.sw.os[<info>]

 Informations sur le système d'exploitation.
 Valeur de retour: *String*.
 Plateformes prises en charge: Linux, Windows.

Paramètre:

- **info** - valeurs possibles: *full* (par défaut), *short*, ou *name*

Les informations de info sont obtenues à partir de (notez que tous les fichiers et options ne sont pas présents dans toutes les distributions):

- `/proc/version` (*full*) sur Linux;
- `/proc/version_signature` (*short*) sur Linux;
- le paramètre PRETTY_NAME de `/etc/os-release` sur les systèmes Linux qui le prennent en charge ou `/etc/issue.net` (*name*);
- la clé de registre `HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion` sur Windows.

Exemples:

```
system.sw.os[short] → Ubuntu 2.6.35-28.50-generic 2.6.35.11
system.sw.os[full] → [s|Windows 10 Enterprise 22621.1.asd64fre.ni_release.220506-1250 Build 22621.963]
```

system.sw.os.get

 Informations détaillées sur le système d'exploitation (version, type, nom de la distribution, version mineure et majeure, etc.).
 Valeur de retour : *chaîne JSON*.
 Plateformes prises en charge : Linux, Windows.

system.sw.packages[<regexp>,<manager>,<format>]

 La liste des paquets installés.
 Valeur de retour: *Text*.
 Plateformes prises en charge: Linux.

Paramètres:

- **regexp** - *all* (par défaut) ou une expression régulière;

- **manager** - *all* (par défaut) ou un gestionnaire de paquets;
- **format** - *full* (par défaut) ou *short*.

Commentaires:

- Liste (par ordre alphabétique) les paquets installés dont le nom correspond à l'expression régulière donnée (*all* les liste tous).
- Gestionnaires de paquets pris en charge (commande exécutée):
  - pkg (`dpkg --get-selections`);
  - pkgtool (`ls /var/log/packages`);
  - rpm (`rpm -qa`);
  - pacman (`pacman -Q`);
  - portage.
- Si `format` est spécifié comme *full*, les paquets sont regroupés par gestionnaire de paquets (chaque gestionnaire sur une ligne séparée commençant par son nom entre crochets).
- Si `format` est spécifié comme *short*, les paquets ne sont pas regroupés et sont listés sur une seule ligne.

Exemple:

```
system.sw.packages[mini,dpkg,short] → python-minimal, python2.6-minimal, ubuntu-minimal
```

system.sw.packages.get[<regexp>,<manager>]

 Une liste détaillée des paquets installés.
 Valeur de retour: *chaîne JSON*.
 Plateformes prises en charge: Linux.

Paramètres:

- **regexp** - *all* (par défaut) ou une expression régulière;
- **manager** - *all* (par défaut) ou un gestionnaire de paquets (valeurs possibles: *rpm*, *dpkg*, *pkgtool*, *pacman* ou *portage*).

Commentaires:

- Renvoie un JSON non formaté contenant les paquets installés dont le nom correspond à l'expression régulière donnée.
- La sortie est un tableau d'objets contenant chacun les clés suivantes: name, manager, version, size, architecture, buildtime et installtime (voir plus de détails).

system.swap.in[<device>,<type>]

 Les statistiques de swap-in (du périphérique vers la mémoire).
 Valeur de retour : *Integer*.
 Plateformes prises en charge : Linux, FreeBSD, OpenBSD.

Paramètres :

- **device** - spécifie le périphérique utilisé pour le swap (Linux uniquement) ou *all* (par défaut) ;
- **type** - valeurs possibles :
  - *count* (nombre de swap-ins, par défaut sur les plateformes non Linux) ;
  - *sectors* (secteurs échangés) ;
  - *pages* (pages échangées, par défaut sur Linux).

Commentaires :

- La source de ces informations est :
  - `/proc/swaps`, `/proc/partitions`, `/proc/stat` (Linux 2.4) ;
  - `/proc/swaps`, `/proc/diskstats`, `/proc/vmstat` (Linux 2.6).
- Notez que *pages* ne fonctionnera que si le périphérique n'a pas été spécifié.
- Le paramètre de type *sectors* est pris en charge uniquement sur Linux.

Exemple :

```
system.swap.in[,pages]
```

system.swap.out[<device>,<type>]

 Les statistiques de swap-out (de la mémoire vers le périphérique).
 Valeur de retour: *Integer*.
 Plateformes prises en charge: Linux, FreeBSD, OpenBSD.

Paramètres:

- **device** - spécifiez le périphérique utilisé pour le swap (Linux uniquement) ou *all* (par défaut);
- **type** - valeurs possibles:
  - *count* (nombre de swap-outs, par défaut sur les plateformes non Linux);
  - *sectors* (secteurs échangés);
  - *pages* (pages échangées, par défaut sur Linux).

Commentaires:

- La source de ces informations est:
  - `/proc/swaps`, `/proc/partitions`, `/proc/stat` (Linux 2.4);
  - `/proc/swaps`, `/proc/diskstats`, `/proc/vmstat` (Linux 2.6).
- Notez que *pages* ne fonctionnera que si device n'a pas été spécifié.
- Le paramètre de type *sectors* est pris en charge uniquement sur Linux.

Exemple:

```
system.swap.out[,pages]
```

system.swap.size[<device>,<type>]

 La taille de l'espace d'échange en octets ou en pourcentage du total.
 Valeur de retour: *Integer* - pour les octets; *Float* - pour le pourcentage.
 Plateformes prises en charge: Linux, FreeBSD, Solaris, AIX, OpenBSD, Windows.

Paramètres:

- **device** - spécifie le périphérique utilisé pour l'échange (FreeBSD uniquement) ou *all* (par défaut);
- **type** - valeurs possibles:
  - *free* (espace d'échange libre, par défaut);
  - *pfree* (espace d'échange libre, en pourcentage);
  - *pused* (espace d'échange utilisé, en pourcentage);
  - *total* (espace d'échange total);
  - *used* (espace d'échange utilisé).

Commentaires:

- Notez que *pfree*, *pused* ne sont pas pris en charge sous Windows si la taille de l'espace d'échange est 0.
- Si device n'est pas spécifié, l'agent Zabbix ne prendra en compte que les périphériques d'échange (fichiers), la mémoire physique sera ignorée. Par exemple, sur les systèmes Solaris, la commande `swap -s` inclut une partie de la mémoire physique et des périphériques d'échange (contrairement à `swap -l`).

Exemple:

```
system.swap.size[,pfree] → pourcentage d'espace d'échange libre
```

system.uname

 Identification du système.
 Valeur de retour : *String*.
 Voir les plateformes prises en charge.

Commentaires :

- Sous UNIX, la valeur de cet élément est obtenue à l'aide de l'appel système uname().
- Sous Windows, l'élément renvoie l'architecture du système d'exploitation, tandis que sous UNIX il renvoie l'architecture du processeur.

Exemples :

```
system.uname → FreeBSD localhost 4.2-RELEASE FreeBSD 4.2-RELEASE #0: Mon Nov i386
system.uname → Windows ZABBIX-WIN 6.0.6001 Microsoft® Windows Server® 2008 Standard Service Pack 1 x86
```

system.uptime

 Le temps de fonctionnement du système en secondes.
 Valeur de retour : *Integer*.
 Plateformes prises en charge : Linux, FreeBSD, Solaris, AIX, MacOS X, OpenBSD, NetBSD, Windows.

Dans la configuration de l'élément, utilisez les unités **s** ou **uptime** pour obtenir des valeurs lisibles.

system.users.num

 Le nombre d'utilisateurs connectés.
 Valeur de retour : *Integer*.
 Voir les plateformes prises en charge.

La commande **who** est utilisée du côté de l'agent pour obtenir la valeur.

vfs.dev.discovery

 La liste des périphériques de bloc et de leur type. Utilisé pour la découverte de bas niveau.
 Valeur de retour : *chaîne JSON*.
 Plateformes prises en charge : Linux.

vfs.dev.get[<mode>,<regexp>]

 La liste des périphériques de bloc avec des informations détaillées. Utilisé pour la découverte de bas niveau.
 Valeur de retour: *chaîne JSON*.
 Plateformes prises en charge: Linux, AIX.

Paramètres:

- **mode** - valeurs possibles:
  - *disks* (par défaut) - renvoie la liste des disques;
  - *disk_stats* - renvoie les statistiques pour chaque disque;
  - *devices* (Linux uniquement) - renvoie la liste des périphériques de bloc (disques et partitions);
  - *device_stats* (Linux uniquement) - renvoie les statistiques pour chaque périphérique (disques et partitions).
- **regexp** - une expression régulière utilisée pour filtrer les périphériques de bloc par leur nom.

vfs.dev.read[<device>,<type>,<mode>]

 Les statistiques de lecture du disque.
 Valeur de retour : - *Integer* - avec `type` à *sectors*, *operations*, *bytes* ; - *Float* - avec `type` à *sps*, *ops*, *bps*.
 Plateformes prises en charge : Linux, FreeBSD, Solaris, AIX, OpenBSD.

Paramètres :

- **device** - périphérique de disque (par défaut : *all* [3]);
- **type** - valeurs possibles : *sectors*, *operations*, *bytes*, *sps*, *ops*, ou *bps* (*sps*, *ops*, *bps* signifient respectivement : secteurs, opérations, octets par seconde) ;
- **mode** - valeurs possibles : *avg1* (moyenne sur une minute, par défaut), *avg5*, ou *avg15*. Ce paramètre est pris en charge uniquement avec `type` à : sps, ops, bps.

Commentaires :

- Si vous utilisez un intervalle de mise à jour de trois heures ou plus[2], cet élément renverra toujours 0.
- Les paramètres de type *sectors* et *sps* sont pris en charge uniquement sur Linux.
- Le paramètre de type *ops* est pris en charge uniquement sur Linux et FreeBSD.
- Le paramètre de type *bps* est pris en charge uniquement sur FreeBSD.
- Le paramètre de type *bytes* est pris en charge uniquement sur FreeBSD, Solaris, AIX, OpenBSD.
- Le paramètre `mode` est pris en charge uniquement sur Linux, FreeBSD.
- Vous pouvez utiliser des noms de périphériques relatifs (par exemple, `sda`) ainsi qu'un préfixe `/dev/` facultatif (par exemple, `/dev/sda`).
- Les volumes logiques LVM sont pris en charge.
- Les valeurs par défaut du paramètre 'type' selon les différents systèmes d'exploitation :
  - *AIX* - operations;
  - *FreeBSD* - bps;
  - *Linux* - sps;
  - *OpenBSD* - operations;
  - *Solaris* - bytes.
- *sps*, *ops* et *bps* sur les plateformes prises en charge sont limités à 1024 périphériques (1023 individuels et un pour *all*).

Exemple :

```
vfs.dev.read[,operations]
```

vfs.dev.write[<device>,<type>,<mode>]

 Les statistiques d'écriture du disque.
 Valeur de retour: *Integer* - avec `type` dans *sectors*, *operations*, *bytes*; *Float* - avec `type` dans *sps*, *ops*, *bps*.
 Plateformes prises en charge: Linux, FreeBSD, Solaris, AIX, OpenBSD.

Paramètres:

- **device** - périphérique de disque (la valeur par défaut est *all* [3]);
- **type** - valeurs possibles: *sectors*, *operations*, *bytes*, *sps*, *ops*, ou *bps* (*sps*, *ops*, *bps* signifient respectivement: secteurs, opérations, octets par seconde);
- **mode** - valeurs possibles: *avg1* (moyenne sur une minute, par défaut), *avg5*, ou *avg15*. Ce paramètre n'est pris en charge qu'avec `type` dans: sps, ops, bps.

Commentaires:

- Si vous utilisez un intervalle de mise à jour de trois heures ou plus[2], cet élément renverra toujours 0.
- Les paramètres de type *sectors* et *sps* ne sont pris en charge que sur Linux.
- Le paramètre de type *ops* n'est pris en charge que sur Linux et FreeBSD.
- Le paramètre de type *bps* n'est pris en charge que sur FreeBSD.
- Le paramètre de type *bytes* n'est pris en charge que sur FreeBSD, Solaris, AIX, OpenBSD.
- Le paramètre `mode` n'est pris en charge que sur Linux, FreeBSD.
- Vous pouvez utiliser des noms de périphériques relatifs (par exemple, `sda`) ainsi qu'un préfixe `/dev/` facultatif (par exemple, `/dev/sda`).
- Les volumes logiques LVM sont pris en charge.
- Les valeurs par défaut du paramètre `type` selon les différents systèmes d'exploitation:
  - *AIX* - operations
  - *FreeBSD* - bps

    - *Linux* - sps
    - *OpenBSD* - operations
    - *Solaris* - bytes
- *sps*, *ops* et *bps* sur les plateformes prises en charge sont limités à 1024 périphériques (1023 individuels et un pour *all*).

Exemple:

```
vfs.dev.write[,operations]
```

vfs.dir.count[dir,<regex incl>,<regex excl>,<types incl>,<types excl>,<max depth>,<min size>,<max size>,<min age>,<max age>,<regex excl dir>]

 Le nombre d'entrées du répertoire.
 Valeur retournée : *Integer*.
 Voir les plateformes prises en charge.

Paramètres :

- **dir** - le chemin absolu vers le répertoire ;
- **regex incl** - une expression régulière décrivant le modèle de nom de l'entité (fichier, répertoire, lien symbolique) à inclure ; tout inclure si vide (valeur par défaut) ;
- **regex excl** - une expression régulière décrivant le modèle de nom de l'entité (fichier, répertoire, lien symbolique) à exclure ; ne rien exclure si vide (valeur par défaut) ;
- **types incl** - les types d'entrées de répertoire à compter, valeurs possibles :
    - *file* - fichier ordinaire ;
    - *dir* - sous-répertoire ;
    - *sym* - lien symbolique ;
    - *sock* - socket ;
    - *bdev* - périphérique bloc ;
    - *cdev* - périphérique caractère ;
    - *fifo* - FIFO ;
    - *dev* - synonyme de "bdev,cdev" ;
    - *all* - tous les types (par défaut), c'est-à-dire "file,dir,sym,sock,bdev,cdev,fifo". Plusieurs types doivent être séparés par des virgules et placés entre guillemets.
- **types excl** - les types d'entrées de répertoire (voir `types incl`) à NE PAS compter. Si un type d'entrée figure à la fois dans `types incl` et `types excl`, les entrées de répertoire de ce type NE sont PAS comptées.
- **max depth** - la profondeur maximale des sous-répertoires à parcourir :
    - **-1** (par défaut) - illimitée ;
    - **0** - ne pas descendre dans les sous-répertoires.
- **min size** - la taille minimale (en octets) du fichier à compter. Les fichiers plus petits ne seront pas comptés. Les suffixes de mémoire peuvent être utilisés.
- **max size** - la taille maximale (en octets) du fichier à compter. Les fichiers plus grands ne seront pas comptés. Les suffixes de mémoire peuvent être utilisés.
- **min age** - l'ancienneté minimale (en secondes) de l'entrée de répertoire à compter. Les entrées plus récentes ne seront pas comptées. Les suffixes de temps peuvent être utilisés.
- **max age** - l'ancienneté maximale (en secondes) de l'entrée de répertoire à compter. Les entrées aussi anciennes ou plus anciennes ne seront pas comptées (heure de modification). Les suffixes de temps peuvent être utilisés.
- **regex excl dir** - une expression régulière décrivant le modèle de nom du répertoire à exclure. Tout le contenu du répertoire sera exclu (contrairement à `regex_excl`).

Commentaires :

- Sous Windows, cet élément suit les chemins UNC, ce qui peut constituer un risque de sécurité.
- Les variables d'environnement, par exemple `%APP_HOME%`, `$HOME` et `%TEMP%`, ne sont pas prises en charge.
- Les pseudo-répertoires "." et ".." ne sont jamais comptés.
- Les liens symboliques ne sont jamais suivis lors du parcours des répertoires.
- `regex incl` et `regex excl` sont tous deux appliqués aux fichiers et aux répertoires lors du calcul du nombre d'entrées, mais sont ignorés lors de la sélection des sous-répertoires à parcourir (si `regex incl` est `(?i)^.+\.zip$` et que `max depth` n'est pas défini, tous les sous-répertoires seront parcourus, mais seuls les fichiers de type zip seront comptés).
- Le temps d'exécution est limité par la valeur du délai d'attente dans la configuration de l'agent (3 secondes). Comme le parcours de grands répertoires peut prendre plus de temps, aucune donnée ne sera renvoyée et l'élément deviendra non pris en charge. Le décompte partiel ne sera pas renvoyé.
- Lors du filtrage par taille, seuls les fichiers ordinaires ont des tailles significatives. Sous Linux et BSD, les répertoires ont également des tailles non nulles (généralement quelques Ko). Les périphériques ont une taille nulle ; par exemple, la taille de `/dev/sda1` ne reflète pas la taille de la partition correspondante. Par conséquent, lors de l'utilisation de `<min_size>` et `<max_size>`, il est recommandé de spécifier `<types_incl>` comme "*file*", afin d'éviter toute surprise.

Exemples :

```
vfs.dir.count[/dev] #surveille le nombre de périphériques dans /dev (Linux)
vfs.dir.count["C:\Users\ADMINI~1\AppData\Local\Temp"] #surveille le nombre de fichiers dans un répertoire
```

vfs.dir.get[dir,<regex incl>,<regex excl>,<types incl>,<types excl>,<max depth>,<min size>,<max size>,<min age>,<max age>,<regex excl dir>]

 La liste des entrées du répertoire.
 Valeur retournée : *chaîne JSON*.
 Voir les plateformes prises en charge.

Paramètres :

- **dir** - le chemin absolu vers le répertoire ;
- **regex incl** - une expression régulière décrivant le modèle de nom de l'entité (fichier, répertoire, lien symbolique) à inclure ; tout inclure si vide (valeur par défaut) ;
- **regex excl** - une expression régulière décrivant le modèle de nom de l'entité (fichier, répertoire, lien symbolique) à exclure ; ne rien exclure si vide (valeur par défaut) ;
- **types incl** - les types d'entrées de répertoire à lister, valeurs possibles :
  - *file* - fichier ordinaire ;
  - *dir* - sous-répertoire ;
  - *sym* - lien symbolique ;
  - *sock* - socket ;
  - *bdev* - périphérique bloc ;
  - *cdev* - périphérique caractère ;
  - *fifo* - FIFO ;
  - *dev* - synonyme de « bdev,cdev » ;
  - *all* - tous les types (par défaut), c'est-à-dire « file,dir,sym,sock,bdev,cdev,fifo ». Plusieurs types doivent être séparés par une virgule et placés entre guillemets.
- **types excl** - les types d'entrées de répertoire (voir `types incl`) à NE PAS lister. Si un type d'entrée figure à la fois dans `types incl` et `types excl`, les entrées de répertoire de ce type NE sont PAS listées.
- **max depth** - la profondeur maximale des sous-répertoires à parcourir :
  - **-1** (par défaut) - illimitée ;
  - **0** - ne pas descendre dans les sous-répertoires.
- **min size** - la taille minimale (en octets) d'un fichier à lister. Les fichiers plus petits ne seront pas listés. Les suffixes de mémoire peuvent être utilisés.
- **max size** - la taille maximale (en octets) d'un fichier à lister. Les fichiers plus grands ne seront pas listés. Les suffixes de mémoire peuvent être utilisés.
- **min age** - l'ancienneté minimale (en secondes) d'une entrée de répertoire à lister. Les entrées plus récentes ne seront pas listées. Les suffixes de temps peuvent être utilisés.
- **max age** - l'ancienneté maximale (en secondes) d'une entrée de répertoire à lister. Les entrées ayant cette ancienneté ou une ancienneté supérieure ne seront pas listées (heure de modification). Les suffixes de temps peuvent être utilisés.
- **regex excl dir** - une expression régulière décrivant le modèle de nom du répertoire à exclure. Tout le contenu du répertoire sera exclu (contrairement à `regex excl`).

Commentaires :

- Sous Windows, cet élément suit les chemins UNC, ce qui peut constituer un risque de sécurité.
- Les variables d'environnement, telles que `%APP_HOME%`, `$HOME` et `%TEMP%`, ne sont pas prises en charge.
- Les pseudo-répertoires « . » et « .. » ne sont jamais listés.
- Les liens symboliques ne sont jamais suivis lors du parcours des répertoires.
- `regex incl` et `regex excl` sont tous deux appliqués aux fichiers et aux répertoires lors de la génération de la liste des entrées, mais sont ignorés lors de la sélection des sous-répertoires à parcourir (si `regex incl` est `(?i)^.+\.zip$` et que `max depth` n'est pas défini, tous les sous-répertoires seront parcourus, mais seuls les fichiers de type zip seront comptabilisés).
- Le temps d'exécution est limité par la valeur de délai d'attente dans la configuration de l'agent. Comme le parcours d'un répertoire volumineux peut prendre plus de temps, aucune donnée ne sera retournée et l'élément deviendra non pris en charge. La liste partielle ne sera pas retournée.
- Lors du filtrage par taille, seuls les fichiers ordinaires ont des tailles significatives. Sous Linux et BSD, les répertoires ont également des tailles non nulles (généralement quelques Ko). Les périphériques ont une taille nulle ; par exemple, la taille de `/dev/sda1` ne reflète pas la taille de la partition correspondante. Par conséquent, lors de l'utilisation de `min size` et `max size`, il est recommandé de spécifier `types incl` comme « *file* », afin d'éviter les surprises.

Exemples :

```
vfs.dir.get[/dev] #récupère la liste des périphériques dans /dev (Linux)
vfs.dir.get["C:\Users\ADMINI~1\AppData\Local\Temp"] #récupère la liste des fichiers dans un répertoire tem
```

vfs.dir.size[dir,<regex incl>,<regex excl>,<mode>,<max depth>,<regex excl dir>]

 La taille du répertoire (en octets).
 Valeur de retour: *Integer*.
 Plateformes prises en charge: Linux, Windows. L'élément peut fonctionner sur d'autres plateformes de type UNIX.

Paramètres:

- **dir** - le chemin absolu vers le répertoire;
- **regex incl** - une expression régulière décrivant le motif du nom de l'entité (fichier, répertoire, lien symbolique) à inclure; inclure tout si vide (valeur par défaut);
- **regex excl** - une expression régulière décrivant le motif du nom de l'entité (fichier, répertoire, lien symbolique) à exclure; n'exclure aucun élément si vide (valeur par défaut);
- **mode** - valeurs possibles:
  - *apparent* (par défaut) - obtient les tailles apparentes des fichiers plutôt que l'utilisation disque (agit comme `du -sb dir`);
  - *disk* - obtient l'utilisation disque (agit comme `du -s -B1 dir`). Contrairement à la commande du, l'élément vfs.dir.size prend en compte les fichiers cachés lors du calcul de la taille du répertoire (agit comme `du -sb .[^.]* * ` dans dir).
- **max depth** - profondeur maximale de parcours des sous-répertoires:
  - **-1** (par défaut) - illimitée;
  - **0** - pas de descente dans les sous-répertoires.
- **regex excl dir** - une expression régulière décrivant le motif du nom du répertoire à exclure. Tout le contenu du répertoire sera exclu (contrairement à `regex excl`)

Commentaires:

- Sous Windows, cet élément suit les chemins UNC, ce qui peut constituer un risque de sécurité.
- Seuls les répertoires disposant au minimum de l'autorisation de lecture pour l'utilisateur *zabbix* sont calculés. Pour les répertoires disposant uniquement de l'autorisation de lecture, seule la taille du répertoire lui-même est calculée. Les répertoires disposant des autorisations de lecture et d'exécution sont calculés en incluant leur contenu.
- Pour les répertoires volumineux ou les disques lents, cet élément peut expirer en raison du paramètre Timeout dans les fichiers de configuration agent et serveur/proxy. Augmentez les valeurs de timeout si nécessaire.
- La limite de taille de fichier dépend de la prise en charge des grands fichiers.

Exemples:

```
vfs.dir.size[/tmp,log] #calcule la taille de tous les fichiers dans /tmp contenant 'log' dans leur nom
vfs.dir.size[/tmp,log,^.+\.old$] #calcule la taille de tous les fichiers dans /tmp contenant 'log' dans le
```

vfs.file.cksum[file,<mode>]

 La somme de contrôle du fichier, calculée par l'algorithme UNIX cksum.
 Valeur de retour: *Integer* - avec `mode` à *crc32*, *String* - avec `mode` à *md5*, *sha256*.
 Voir les plateformes prises en charge.

Paramètres:

- **file** - le chemin complet vers le fichier;
- **mode** - *crc32* (par défaut), *md5* ou *sha256*.

Commentaires:

- Sous Windows, cet élément suit les chemins UNC, ce qui peut constituer un risque de sécurité.
- La limite de taille du fichier dépend de la prise en charge des fichiers volumineux.

Exemple:

```
vfs.file.cksum[/etc/passwd]
```

Exemples de valeurs retournées (crc32/md5/sha256 respectivement):

```
675436101
9845acf68b73991eb7fd7ee0ded23c44
ae67546e4aac995e5c921042d0cf0f1f7147703aa42bfbfb65404b30f238f2dc
```

vfs.file.contents[file,<encoding>]

 Récupération du contenu d'un fichier[7].
 Valeur de retour: *Text*.
 Voir plates-formes prises en charge.

Paramètres:

- **file** - le chemin complet vers le fichier;
- **encoding** - l'identifiant de page de code identifiant.

Commentaires:

- Sous Windows, cet élément suit les chemins UNC, ce qui peut constituer un risque de sécurité.
- La valeur de retour est limitée à 16 Mo (y compris les espaces blancs de fin qui sont tronqués); les limites de la base de données s'appliquent également.
- Une chaîne vide est renvoyée si le fichier est vide ou ne contient que des caractères LF/CR.
- Le marqueur d'ordre des octets (BOM) est exclu de la sortie.

Exemple:

```
vfs.file.contents[/etc/passwd]
```

vfs.file.exists[file,<types incl>,<types excl>]

 Vérifie si le fichier existe.
 Valeur de retour: 0 - introuvable; 1 - fichier du type spécifié existant.
 Voir les plateformes prises en charge.

Paramètres:

- **file** - le chemin complet vers le fichier;
- **types incl** - la liste des types de fichiers à inclure, valeurs possibles:
  - *file* (fichier ordinaire, valeur par défaut (si `types_excl` n'est pas défini));
  - *dir* (répertoire);
  - *sym* (lien symbolique);
  - *sock* (socket);
  - *bdev* (périphérique bloc);
  - *cdev* (périphérique caractère);
  - *fifo* (FIFO);
  - *dev* (synonyme de "bdev,cdev");
  - *all* (tous les types mentionnés, valeur par défaut si `types_excl` est défini).
- **types excl** - la liste des types de fichiers à exclure, voir `types_incl` pour les valeurs possibles (par défaut, aucun type n'est exclu)

Commentaires:

- Sous Windows, cet élément suit les chemins UNC, ce qui peut constituer un risque de sécurité.
- Plusieurs types doivent être séparés par une virgule et l'ensemble doit être placé entre guillemets "".
- Si le même type figure à la fois dans <types_incl> et <types_excl>, les fichiers de ce type sont exclus.
- La limite de taille de fichier dépend de la prise en charge des gros fichiers.

Exemples:

```
vfs.file.exists[/tmp/application.pid]
vfs.file.exists[/tmp/application.pid,"file,dir,sym"]
vfs.file.exists[/tmp/application_dir,dir]
```

vfs.file.get[file]

 Renvoie des informations sur un fichier.
 Valeur de retour: *chaîne JSON*.
 Voir plates-formes prises en charge.

Paramètre:

- **file** - le chemin complet vers le fichier

Commentaires:

- Sous Windows, cet élément suit les chemins UNC, ce qui peut présenter un risque de sécurité.
- Types de fichiers pris en charge sur les systèmes de type UNIX: fichier régulier, répertoire, lien symbolique, socket, périphérique bloc, périphérique caractère, FIFO.
- La limite de taille de fichier dépend de la prise en charge des fichiers volumineux.

Exemple:

```
vfs.file.get[/etc/passwd] #return a JSON with information about the /etc/passwd file (type, user, permissi
```

vfs.file.md5sum[file]

 La somme de contrôle MD5 du fichier.
 Valeur de retour: chaîne de caractères (hachage MD5 du fichier).
 Voir les plateformes prises en charge.

Paramètre:

- **file** - le chemin complet vers le fichier

Commentaires:

- Sous Windows, cet élément suit les chemins UNC, ce qui peut constituer un risque de sécurité.
- La limite de taille de fichier dépend de la prise en charge des fichiers volumineux.

Exemple:

```
vfs.file.md5sum[/usr/local/etc/zabbix_agentd.conf]
```

Exemple de valeur renvoyée:

```
b5052decb577e0fffd622d6ddc017e82
```

vfs.file.owner[file,<ownertype>,<resulttype>]

 Récupère le propriétaire d'un fichier.
 Valeur de retour: *String*.
 Voir les plates-formes prises en charge.

Paramètres:

- **file** - le chemin complet vers le fichier;
- **ownertype** - *user* (par défaut) ou *group* (Unix uniquement);
- **resulttype** - *name* (par défaut) ou *id*; pour id - renvoie uid/gid sur Unix, SID sur Windows.

Commentaires:

- Sous Windows, cet élément suit les chemins UNC, ce qui peut constituer un risque de sécurité.
- La limite de taille de fichier dépend de la prise en charge des grands fichiers.

Exemples:

```
vfs.file.owner[/tmp/zabbix_server.log] #renvoie le propriétaire du fichier /tmp/zabbix_server.log
vfs.file.owner[/tmp/zabbix_server.log,,id] #renvoie l'ID du propriétaire du fichier /tmp/zabbix_server.log
```

vfs.file.permissions[file]

 Renvoie une chaîne de 4 chiffres contenant le nombre octal des permissions UNIX.
 Valeur de retour : *String*.
 Plateformes prises en charge : Linux. L'élément peut fonctionner sur d'autres plateformes de type UNIX.

Paramètres :

- **file** - le chemin complet vers le fichier

La limite de taille du fichier dépend de la prise en charge des fichiers volumineux.

Exemple :

```
vfs.file.permissions[/etc/passwd] #renvoie les permissions de /etc/passwd, par exemple, '0644'
```

vfs.file.regexp[file,regexp,<encoding>,<start line>,<end line>,<output>]

 Récupère une chaîne dans le fichier[7].
 Valeur de retour: la ligne contenant la chaîne correspondante, ou selon ce qui est spécifié par le paramètre facultatif `output`.
 Voir les plateformes prises en charge.

Paramètres:

- **file** - le chemin complet vers le fichier;
- **regexp** - une expression régulière décrivant le motif requis;
- **encoding** - l'identifiant de la page de code;
- **start line** - le numéro de la première ligne à rechercher (première ligne du fichier par défaut);
- **end line** - le numéro de la dernière ligne à rechercher (dernière ligne du fichier par défaut);
- **output** - un modèle de formatage de sortie facultatif. La séquence d'échappement **\0** est remplacée par la partie correspondante du texte (de la première caractère où la correspondance commence jusqu'au caractère où la correspondance se termine), tandis que la séquence d'échappement **\N** (où N=1...9) est remplacée par le N-ième groupe correspondant (ou une chaîne vide si N dépasse le nombre de groupes capturés).

Commentaires:

- Sous Windows, cet élément suit les chemins UNC, ce qui peut présenter un risque de sécurité.
- La limite de taille du fichier dépend de la prise en charge des fichiers volumineux.
- Seule la première ligne correspondante est renvoyée.
- Une chaîne vide est renvoyée si aucune ligne ne correspond à l'expression.
- Le marqueur d'ordre des octets (BOM) est exclu de la sortie.
- L'extraction du contenu à l'aide du paramètre `output` s'effectue sur l'agent.

Exemples:

```
vfs.file.regexp[/etc/passwd,zabbix]
vfs.file.regexp[/path/to/some/file,"([0-9]+)$",,3,5,\1]
vfs.file.regexp[/etc/passwd,"^zabbix:.:([0-9]+)",,,,\1] → obtention de l'ID de l'utilisateur *zabbix*
```

vfs.file.regmatch[file,regexp,<encoding>,<start line>,<end line>]

 Trouve une chaîne dans le fichier[7].
 Valeurs de retour: 0 - correspondance non trouvée; 1 - trouvée.
 Voir les plates-formes prises en charge.

Paramètres:

- **file** - le chemin complet vers le fichier;
- **regexp** - une expression régulière décrivant le motif requis;
- **encoding** - l'identifiant de page de code;
- **start line** - le numéro de la première ligne à rechercher (première ligne du fichier par défaut);
- **end line** - le numéro de la dernière ligne à rechercher (dernière ligne du fichier par défaut).

Commentaires:

- Sous Windows, cet élément suit les chemins UNC, ce qui peut présenter un risque de sécurité.
- La limite de taille du fichier dépend de la prise en charge des grands fichiers.
- Le BOM (byte order mark) est ignoré.

Exemple:

```
vfs.file.regmatch[/var/log/app.log,error]
```

vfs.file.size[file,<mode>]

 La taille du fichier (en octets).
 Valeur de retour: *Integer*.
 Voir les plateformes prises en charge.

Paramètres:

- **file** - le chemin complet vers le fichier;
- **mode** - valeurs possibles:
  - *bytes* (par défaut);
  - *lines* (les lignes vides sont également comptées).

Commentaires:

- Sous Windows, cet élément suit les chemins UNC, ce qui peut présenter un risque de sécurité.
- Le fichier doit avoir des permissions de lecture pour l'utilisateur *zabbix*.
- La limite de taille du fichier dépend de la prise en charge des fichiers volumineux.

Exemple:

```
vfs.file.size[/var/log/syslog]
```

vfs.file.time[file,<mode>]

 Informations sur l'heure du fichier.
 Valeur de retour: *Integer* (horodatage Unix).
 Voir plateformes prises en charge.

Paramètres:

- **file** - le chemin complet vers le fichier;
- **mode** - valeurs possibles:
  - *modify* (par défaut) - la dernière heure de modification du contenu du fichier (par défaut);
  - *access* - la dernière heure de lecture du fichier;
  - *change* - la dernière heure de modification des propriétés du fichier.

Commentaires:

- Sous Windows, cet élément suit les chemins UNC, ce qui peut représenter un risque de sécurité.
- La limite de taille de fichier dépend de la prise en charge des grands fichiers.

Exemple:

```
vfs.file.time[/etc/passwd,modify]
```

vfs.fs.discovery

 La liste des systèmes de fichiers montés avec leur type et leurs options de montage. Utilisé pour la découverte de bas niveau.
 Valeur de retour : *chaîne JSON*.
 Plateformes prises en charge : Linux, FreeBSD, Solaris, HP-UX, AIX, MacOS X, OpenBSD, NetBSD, Windows.

vfs.fs.get[<mode>,<mountpoint>]

 La liste des systèmes de fichiers montés avec leur type et leurs options de montage et, en mode *full*, l'espace disque disponible et les statistiques des inodes. Peut être utilisé pour la découverte de bas niveau.
 Valeur de retour : *chaîne JSON*.
 Plateformes prises en charge : Linux, FreeBSD, Solaris, HP-UX, AIX, MacOS X, OpenBSD, NetBSD, Windows.

Paramètres :

- **mode** - valeurs possibles :
  - *full* (par défaut) - propriétés du système de fichiers avec l'espace disque et les statistiques des inodes ;
  - *short* - propriétés du système de fichiers uniquement, sans l'espace disque ni les statistiques des inodes ;
- **mountpoint** - point de montage pour lequel les données doivent être renvoyées (par exemple, `/var` ou `C:`). S'il n'est pas spécifié, tous les systèmes de fichiers montés sont renvoyés.

Commentaires :

- En mode *short*, seuls le nom, le type et les options de montage du système de fichiers sont renvoyés (sous Windows : le nom, le type, le libellé et le type de lecteur). Les systèmes de fichiers ne sont pas interrogés pour obtenir leurs statistiques ; sur les systèmes de type Unix, seule la liste des systèmes de fichiers montés est lue (par exemple, `/proc/mounts` sous Linux). Ainsi, la vérification ne déclenche pas le montage automatique et n'est pas affectée par les systèmes de fichiers réseau qui ne répondent pas. Le mode *short* convient donc à la découverte de bas niveau.
- La valeur **mountpoint** doit correspondre exactement à la propriété `fsname`. Sous Windows, la comparaison ne tient pas compte de la casse ; spécifiez le point de montage sans la barre oblique inverse finale (par exemple, `C:`).
- Si **mountpoint** est spécifié, les données sont renvoyées uniquement pour ce point de montage. Si aucun système de fichiers monté ne correspond, un tableau vide (`[]`) est renvoyé.
- La sortie peut contenir plusieurs entrées avec le même `fsname` si plusieurs systèmes de fichiers sont montés dans le même répertoire.
- En mode *full*, l'espace disque disponible et les statistiques des inodes sont renvoyés uniquement si l'utilisateur du agent Zabbix dispose des autorisations de parcours sur chaque répertoire du chemin du point de montage ; sinon, le système de fichiers est omis de la sortie. Cela ne s'applique pas au mode *short*, qui peut donc répertorier des systèmes de fichiers absents de la sortie du mode *full*.
- Les propriétés `fslabel` et `fsdrivetype` sont renvoyées uniquement sous Windows ; les propriétés `inodes` et `options` ne sont pas renvoyées sous Windows.
- Les systèmes de fichiers dont le nombre d'inodes est égal à zéro, ce qui peut être le cas des systèmes de fichiers utilisant des inodes dynamiques (par exemple, btrfs), sont également signalés.
- Voir également : Découverte des systèmes de fichiers montés.

Exemples :

```
vfs.fs.get #tous les systèmes de fichiers montés avec leurs statistiques ; équivalent à vfs.fs.get[full]
vfs.fs.get[short] #tous les systèmes de fichiers montés sans statistiques
vfs.fs.get[full,/var] #statistiques du point de montage /var uniquement
vfs.fs.get[short,C:] #propriétés du volume C: uniquement (Windows)
```

vfs.fs.inode[fs,<mode>]

 Le nombre ou le pourcentage d'inodes.
 Valeur de retour : *Integer* - pour le nombre ; *Float* - pour le pourcentage.
 Voir les plateformes prises en charge.

Paramètres :

- **fs** - le système de fichiers ;
- **mode** - valeurs possibles : *total* (par défaut), *free*, *used*, *pfree* (libre, en pourcentage) ou *pused* (utilisé, en pourcentage).

Si le nombre d'inodes est égal à zéro, ce qui peut être le cas pour les systèmes de fichiers avec des inodes dynamiques (par ex. btrfs), les valeurs pfree/pused seront signalées respectivement comme « 100 » et « 0 ».

Exemple :

```
vfs.fs.inode[/,pfree]
```

vfs.fs.size[fs,<mode>]

 L'espace disque en octets ou en pourcentage du total.
 Valeur de retour : *Integer* - pour les octets ; *Float* - pour le pourcentage.
 Voir les plateformes prises en charge.

Paramètres :

- **fs** - le système de fichiers ;
- **mode** - valeurs possibles : *total* (par défaut), *free*, *used*, *pfree* (libre, en pourcentage) ou *pused* (utilisé, en pourcentage).

Commentaires :

- Si le système de fichiers n'est pas monté, renvoie la taille d'un système de fichiers local où se trouve le point de montage.
- L'espace réservé d'un système de fichiers est pris en compte et n'est pas inclus lors de l'utilisation du mode *free*.

Exemple :

```
vfs.fs.size[/tmp,free]
```

vm.memory.size[<mode>]

 La taille de la mémoire en octets ou en pourcentage du total.
 Valeur de retour : *Integer* - pour les octets ; *Float* - pour le pourcentage.
 Voir les plateformes prises en charge.

Paramètre :

- **mode** - valeurs possibles : *total* (par défaut), *active*, *anon*, *buffers*, *cached*, *exec*, *file*, *free*, *inactive*, *pinned*, *shared*, *slab*, *wired*, *used*, *pused* (utilisée, en pourcentage), *available*, ou *pavailable* (disponible, en pourcentage).

Commentaires :

- Cet élément accepte trois catégories de paramètres :
  - *total* - quantité totale de mémoire ;
  - types de mémoire spécifiques à la plateforme : *active*, *anon*, *buffers*, *cached*, *exec*, *file*, *free*, *inactive*, *pinned*, *shared*, *slab*, *wired* ;
  - estimations au niveau utilisateur de la quantité de mémoire utilisée et disponible : *used*, *pused*, *available*, *pavailable*.
- Le paramètre de mode *active* est pris en charge uniquement sur FreeBSD, HP-UX, MacOS X, OpenBSD, NetBSD.
- Les paramètres de mode *anon*, *exec*, *file* sont pris en charge uniquement sur NetBSD.
- Le paramètre de mode *buffers* est pris en charge uniquement sur Linux, FreeBSD, OpenBSD, NetBSD.
- Le paramètre de mode *cached* est pris en charge uniquement sur Linux, FreeBSD, AIX, OpenBSD, NetBSD.
- Les paramètres de mode *inactive*, *wired* sont pris en charge uniquement sur FreeBSD, MacOS X, OpenBSD, NetBSD.
- Le paramètre de mode *pinned* est pris en charge uniquement sur AIX.
- Le paramètre de mode *shared* est pris en charge uniquement sur Linux 2.4, FreeBSD, OpenBSD, NetBSD.
- Voir aussi les détails supplémentaires pour cet élément.

Exemple :

```
vm.memory.size[pavailable]
```

web.page.get[host,<path>,<port>,<redirect_limit>]

 Obtenir le contenu d'une page web.
 Valeur retournée : source de la page web sous forme de texte (y compris les en-têtes).
 Voir les plateformes prises en charge.

Paramètres :

- **host** - le nom d'hôte ou l'URL (au format `scheme://host:port/path`, où seul *host* est obligatoire). Schémas d'URL autorisés : *http*, *https*[4]. Si le schéma est absent, *http* est utilisé. Si une URL est spécifiée, `path` et `port` doivent être vides. La spécification d'un nom d'utilisateur et d'un mot de passe lors de la connexion à des serveurs nécessitant une authentification, par exemple : `http://user:password@www.example.com`, est possible uniquement avec la prise en charge de cURL [4]. Le Punycode est pris en charge dans les noms d'hôte.
- **path** - le chemin vers un document HTML (la valeur par défaut est /) ;
- **port** - le numéro de port (la valeur par défaut est 80 pour HTTP)
- **redirect_limit** - le nombre maximal de redirections à suivre [4], de 0 à 1000 (la valeur par défaut est 0 - les redirections ne sont pas suivies).

Commentaires :

- Cet élément devient non pris en charge si la ressource spécifiée dans `host` n'existe pas ou n'est pas disponible, ou si `redirect_limit` est supérieur à 0 et que le nombre de redirections le dépasse.
- `host` peut être un nom d'hôte, un nom de domaine, une adresse IPv4 ou IPv6. Cependant, pour une adresse IPv6, l'agent Zabbix doit être compilé avec la prise en charge d'IPv6 activée.
- Si les redirections sont suivies, la valeur retournée contient les en-têtes de chaque réponse de redirection, suivis des en-têtes et du corps de la réponse finale.

Exemples :

```
web.page.get[www.example.com,index.php,80]
web.page.get[https://www.example.com]
web.page.get[https://blog.example.com/?s=zabbix]
web.page.get[localhost:80]
web.page.get["[::1]/server-status"]
web.page.get[http://www.example.com,,,5]
```

web.page.perf[host,<path>,<port>,<redirect_limit>]

 Le temps de chargement d'une page web complète (en secondes).
 Valeur renvoyée : *Float*.
 Voir les plateformes prises en charge.

Paramètres :

- **host** - le nom d'hôte ou l'URL (au format `scheme://host:port/path`, où seul *host* est obligatoire). Schémas d'URL autorisés : *http*, *https*[4]. En l'absence de schéma, *http* est utilisé. Si une URL est spécifiée, `path` et `port` doivent être vides. La spécification d'un nom d'utilisateur et d'un mot de passe lors de la connexion à des serveurs nécessitant une authentification, par exemple : `http://user:password@www.example.com`, n'est possible qu'avec la prise en charge de cURL [4]. Le Punycode est pris en charge dans les noms d'hôte.
- **path** - le chemin vers un document HTML (la valeur par défaut est /) ;
- **port** - le numéro de port (la valeur par défaut est 80 pour HTTP) ;
- **redirect_limit** - le nombre maximal de redirections à suivre [4], de 0 à 1000 (la valeur par défaut est 0 - les redirections ne sont pas suivies).

Commentaires :

- Cet élément devient non pris en charge si la ressource spécifiée dans `host` n'existe pas ou n'est pas disponible, ou si `redirect_limit` est supérieur à 0 et que le nombre de redirections le dépasse.
- `host` peut être un nom d'hôte, un nom de domaine, une adresse IPv4 ou IPv6. Cependant, pour une adresse IPv6, l'agent Zabbix doit être compilé avec la prise en charge d'IPv6 activée.
- Si les redirections sont suivies, le temps de chargement inclut le temps consacré à toutes les redirections.

Exemples :

```
web.page.perf[www.example.com,index.php,80]
web.page.perf[https://www.example.com]
web.page.perf[http://www.example.com,,,5]
```

web.page.regexp[host,<path>,<port>,regexp,<length>,<output>,<redirect_limit>]

 Recherche une chaîne sur la page web.
 Valeur de retour : la chaîne correspondante, ou selon les indications du paramètre facultatif `output`.
 Voir les plateformes prises en charge.

Paramètres :

- **host** - le nom d'hôte ou l'URL (au format `scheme://host:port/path`, où seul *host* est obligatoire). Schémas d'URL autorisés : *http*, *https*[4]. Si le schéma est absent, *http* est utilisé. Si une URL est spécifiée, `path` et `port` doivent être vides. La spécification d'un nom d'utilisateur et d'un mot de passe lors de la connexion à des serveurs nécessitant une authentification, par exemple : `http://user:password@www.example.com`, est possible uniquement avec la prise en charge de cURL [4]. Le format Punycode est pris en charge dans les noms d'hôte.
- **path** - le chemin vers un document HTML (la valeur par défaut est /).
- **port** - le numéro de port (la valeur par défaut est 80 pour HTTP).
- **regexp** - une expression régulière décrivant le modèle requis.
- **length** - le nombre maximal de caractères à retourner ;
- **output** - un modèle facultatif de formatage de la sortie. La séquence d'échappement **\0** est remplacée par la partie correspondante du texte (du premier caractère où commence la correspondance jusqu'au caractère où elle se termine), tandis qu'une séquence d'échappement **\N** (où N=1...9) est remplacée par le N-ième groupe correspondant (ou par une chaîne vide si N dépasse le nombre de groupes capturés).
- **redirect_limit** - le nombre maximal de redirections à suivre [4], de 0 à 1000 (la valeur par défaut est 0 - les redirections ne sont pas suivies).

Commentaires :

- Cet élément devient non pris en charge si la ressource spécifiée dans `host` n'existe pas ou n'est pas disponible, ou si `redirect_limit` est supérieur à 0 et que le nombre de redirections le dépasse.
- `host` peut être un nom d'hôte, un nom de domaine, une adresse IPv4 ou IPv6. Cependant, pour une adresse IPv6, l'agent Zabbix doit être compilé avec la prise en charge d'IPv6 activée.
- L'extraction du contenu à l'aide du paramètre `output` est effectuée sur l'agent.
- Si les redirections sont suivies, les en-têtes de chaque réponse de redirection sont également recherchés.

Exemples :

```
web.page.regexp[www.example.com,index.php,80,OK,2]
web.page.regexp[https://www.example.com,,,OK,2]
web.page.regexp[http://www.example.com,,,OK,2,,5]
```

agent.hostmetadata

 Les métadonnées de l'hôte de l'agent.
 Valeur de retour: *String*.
 Voir plates-formes prises en charge.

Renvoie la valeur des paramètres `HostMetadata` ou `HostMetadataItem`, ou une chaîne vide si aucun n'est défini.

agent.hostname

 Le nom d'hôte de l'agent.
 Valeur de retour: *String*.
 Voir plates-formes prises en charge.

Renvoie:

- En tant que vérification passive - le nom du premier hôte répertorié dans le paramètre Hostname du fichier de configuration de l'agent;
- En tant que vérification active - le nom de l'hôte actuel.

agent.ping

 La vérification de disponibilité de l'agent.
 Valeur de retour: Rien - indisponible; 1 - disponible.
 Voir supported platforms.

Utilisez la fonction de déclencheur `nodata()` pour vérifier l'indisponibilité de l'hôte.

agent.variant

 La variante de l'agent Zabbix (agent Zabbix ou agent Zabbix 2).
 Valeur de retour : 1 - agent Zabbix ; 2 - agent Zabbix 2.
 Voir les plateformes prises en charge.

agent.version

 La version de l'agent Zabbix.
 Valeur de retour : *String*.
 Voir les plateformes prises en charge.

Exemple de valeur renvoyée :

```
6.0.3
```

zabbix.stats[<ip>,<port>]

 Renvoie un ensemble de métriques internes du serveur Zabbix ou du proxy. Si `ip` et `port` sont spécifiés, les métriques sont récupérées depuis l'instance distante ; sinon, depuis l'instance locale.
 Valeur de retour : *chaîne JSON*.
 Voir plateformes prises en charge.

Paramètres :

- **ip** - la liste des adresses IP/DNS/masques réseau des serveurs/proxys à interroger à distance (la valeur par défaut est `127.0.0.1`) ;
- **port** - le port du serveur/proxy à interroger à distance (la valeur par défaut est 10051)

Commentaires :

- Un ensemble sélectionné de métriques internes est renvoyé par cet élément. Pour plus de détails, voir Surveillance à distance des statistiques Zabbix.
- Notez que la requête de statistiques ne sera acceptée que depuis les adresses répertoriées dans le paramètre `StatsAllowedIP` serveur/proxy de l'instance cible.

zabbix.stats[<ip>,<port>,queue,<from>,<to>]

 Renvoie le nombre d'éléments surveillés dans la file d'attente qui sont en retard sur le serveur Zabbix ou le proxy. Si `ip` et `port` sont spécifiés, les métriques sont récupérées depuis l'instance distante ; sinon, depuis l'instance locale.
 Valeur de retour : *chaîne JSON*.
 Voir les plates-formes prises en charge.

Paramètres :

- **ip** - la liste des adresses IP/DNS/masques réseau des serveurs/proxys à interroger à distance (la valeur par défaut est `127.0.0.1`) ;
- **port** - le port du serveur/proxy à interroger à distance (la valeur par défaut est 10051)
- **queue** - constante (à utiliser telle quelle)
- **from** - en retard d'au moins (la valeur par défaut est 6 secondes)
- **to** - en retard d'au plus (la valeur par défaut est l'infini)

Notez que la requête de statistiques ne sera acceptée que depuis les adresses répertoriées dans le paramètre 'StatsAllowedIP' serveur/proxy de l'instance cible.

Notes de bas de page

[1] Note spécifique à Linux. Zabbix agent doit disposer d'un accès en lecture seule au système de fichiers `/proc`. Les correctifs du noyau provenant de www.grsecurity.org limitent les droits d'accès des utilisateurs non privilégiés.

[2] `vfs.dev.read[]`, `vfs.dev.write[]` : Zabbix agent mettra fin aux connexions aux périphériques « obsolètes » si les valeurs des éléments ne sont pas consultées pendant plus de 3 heures. Cela peut se produire si un système possède des périphériques dont les chemins changent dynamiquement ou si un périphérique est supprimé manuellement. Notez également que ces éléments, s'ils utilisent un intervalle de mise à jour de 3 heures ou plus, renverront toujours « 0 ».

[3] `vfs.dev.read[]`, `vfs.dev.write[]` : Si la valeur par défaut `all` est utilisée pour le premier paramètre, la clé renverra des statistiques récapitulatives, notamment pour tous les périphériques de type bloc tels que sda et sdb, ainsi que leurs partitions (sda1, sda2, sdb3...) et les périphériques multiples (RAID MD) basés sur ces périphériques/partitions de type bloc et les volumes logiques (LVM) basés sur ces périphériques/partitions de type bloc. Dans ce cas, les valeurs renvoyées doivent être considérées uniquement comme des valeurs relatives (dynamiques dans le temps), et non comme des valeurs absolues.

[4] HTTPS et le suivi des redirections (`redirect_limit` supérieur à 0) sont pris en charge uniquement si Zabbix agent est compilé avec la prise en charge de cURL ; sinon, l'élément devient non pris en charge. Zabbix agent 2 ne nécessite pas cURL pour ces deux fonctionnalités.

[5] Les valeurs `bytes` et `errors` ne sont pas prises en charge pour les interfaces de bouclage sur les systèmes Solaris jusqu'à Solaris 10 6/06 inclus, car les statistiques relatives aux octets, aux erreurs et à l'utilisation ne sont pas stockées et/ou signalées par le noyau. Cependant, si vous supervisez un système Solaris via net-snmp, des valeurs peuvent être renvoyées, car net-snmp contient du code hérité de cmu-snmp datant de 1997 au plus tôt qui, lorsqu'il ne parvient pas à lire les valeurs en octets dans les statistiques de l'interface, renvoie le compteur de paquets (qui existe bien sur les interfaces de bouclage) multiplié par une valeur arbitraire de 308. Cela suppose que la longueur moyenne d'un paquet est de 308 octets, ce qui constitue une estimation très approximative, car la limite MTU sur les systèmes Solaris pour les interfaces de bouclage est de 8892 octets. Ces valeurs ne doivent pas être considérées comme correctes ni même comme approximativement exactes. Il s'agit d'estimations. Zabbix agent n'effectue aucune estimation, mais net-snmp renverra une valeur pour ces champs.

[6] La ligne de commande sur Solaris, obtenue à partir de `/proc/pid/psinfo`, est limitée à 80 octets et contient la ligne de commande telle qu'elle était au démarrage du processus.

[7] Les éléments `vfs.file.contents[]`, `vfs.file.regexp[]`, `vfs.file.regmatch[]` peuvent être utilisés pour récupérer le contenu des fichiers. Si vous souhaitez limiter l'accès à certains fichiers contenant des informations sensibles, exécutez Zabbix agent sous un utilisateur ne disposant d'aucune autorisation d'accès pour consulter ces fichiers.

Utilisation avec les utilitaires en ligne de commande

Notez que lors du test ou de l'utilisation de clés d'élément avec zabbix_agentd ou zabbix_get depuis la ligne de commande, vous devez également tenir compte de la syntaxe du shell.

Par exemple, si un certain paramètre de la clé doit être placé entre guillemets doubles, vous devez échapper explicitement les guillemets doubles, sinon ils seront supprimés par le shell en tant que caractères spéciaux et ne seront pas transmis à l'utilitaire Zabbix.

Exemples :

```
zabbix_agentd -t 'vfs.dir.count[/var/log,,,"file,dir",,0]'

zabbix_agentd -t vfs.dir.count[/var/log,,,\"file,dir\",,0]
```

Paramètres d'encodage

Pour vous assurer que les données acquises ne sont pas corrompues, vous pouvez spécifier l'encodage correct pour le traitement de la vérification (par exemple, `vfs.file.contents`) dans le paramètre `encoding`. La liste des encodages pris en charge (identifiants de pages de codes) est disponible dans la documentation de libiconv (projet GNU) ou dans la documentation du SDK Microsoft Windows pour les "identifiants de pages de codes". Notez que Microsoft indique parfois que certaines pages de codes sont « disponibles uniquement pour les applications managées », ce qui signifie qu'elles sont exposées uniquement via le runtime .NET et ne sont pas disponibles via les API natives Win32. L'agent Zabbix implémente sa propre logique de conversion d'encodage. Par conséquent, ces pages de codes sont prises en charge par l'agent Zabbix même lorsque les fonctions natives de Windows ne les fournissent pas.

Si aucun encodage n'est spécifié dans le paramètre `encoding`, les stratégies de résolution suivantes sont appliquées :

- Si l'encodage n'est pas spécifié (ou s'il s'agit d'une chaîne vide), UTF-8 est utilisé et les données sont traitées « telles quelles » ;
- Analyse du BOM - applicable aux éléments `vfs.file.contents`, `vfs.file.regexp`, `vfs.file.regmatch`. Une tentative est effectuée pour déterminer l'encodage correct en utilisant la marque d'ordre des octets (BOM) au début du fichier. Si aucun BOM n'est présent, la résolution standard (voir ci-dessus) est appliquée à la place.

Dépannage des éléments de l'agent

Dans le cas des vérifications passives, pour éviter que l'élément n'obtienne aucune valeur parce que la requête du serveur vers l'agent expire en premier, il convient de tenir compte des points suivants :

- Lorsque la version de l'agent est antérieure à celle du serveur, la valeur `Timeout` dans la configuration de l'élément (ou le délai d'expiration global) peut devoir être supérieure à la valeur `Timeout` dans le fichier de configuration de l'agent.
- Lorsque la version de l'agent est plus récente que celle du serveur, la valeur `Timeout` dans le fichier de configuration du serveur peut devoir être supérieure à la valeur `Timeout` dans le fichier de configuration de l'agent.

Zabbix agent 2

Zabbix agent 2 prend en charge toutes les clés d'élément prises en charge pour Zabbix agent sur Unix et Windows. Cette page fournit des détails sur les clés d'élément supplémentaires, que vous pouvez utiliser uniquement avec Zabbix agent 2, regroupées par plugin auquel elles appartiennent.

Les clés d'élément sont répertoriées sans paramètres ni informations supplémentaires. Cliquez sur la clé d'élément pour afficher les détails complets.

Item key	Description	Plugin
ceph.df.details	L'utilisation des données du cluster et leur répartition entre les pools.	Ceph (chargeable)
ceph.osd.stats	Statistiques agrégées et par OSD.	
ceph.osd.discovery	La liste des OSD découverts.	
ceph.osd.dump	Les seuils d'utilisation et les états des OSD.	
ceph.ping	Teste si une connexion à Ceph peut être établie.	
ceph.pool.discovery	La liste des pools découverts.	
ceph.status	L'état global du cluster.	
docker.container_info	Informations de bas niveau sur un conteneur.	Docker
docker.container_stats	Les statistiques d'utilisation des ressources du conteneur.	
docker.containers	Renvoie la liste des conteneurs.	
docker.containers.discovery	Renvoie la liste des conteneurs. Utilisé pour la découverte de bas niveau.	
docker.data.usage	Informations sur l'utilisation actuelle des données.	
docker.images	Renvoie la liste des images.	
docker.images.discovery	Renvoie la liste des images. Utilisé pour la découverte de bas niveau.	
docker.info	Les informations système.	
docker.ping	Teste si le démon Docker est actif ou non.	
ember.get	Renvoie le résultat du périphérique requis.	Ember+ (chargeable)
memcached.ping	Teste si une connexion est active ou non.	Memcached
memcached.stats	Obtient la sortie de la commande STATS.	
mongodb.cfg.discovery	Renvoie une liste de serveurs de configuration découverts.	MongoDB (chargeable)
mongodb.collection.stats	Renvoie diverses statistiques de stockage pour une collection donnée.	
mongodb.collections.discovery	Renvoie une liste de collections découvertes.	
mongodb.collections.usage	Renvoie les statistiques d'utilisation des collections.	
mongodb.connpool.stats	Renvoie des informations sur les connexions sortantes ouvertes depuis l'instance de base de données actuelle vers d'autres membres du cluster fragmenté ou de l'ensemble de réplicas.	
mongodb.db.stats	Renvoie les statistiques reflétant l'état d'un système de base de données donné.	
mongodb.db.discovery	Renvoie une liste de bases de données découvertes.	
mongodb.jumbo_chunks.count	Renvoie le nombre de chunks jumbo.	
mongodb.oplog.stats	Renvoie l'état de l'ensemble de réplicas, à partir des données interrogées dans l'oplog.	
mongodb.ping	Teste si une connexion est active ou non.	
mongodb.rs.config	Renvoie la configuration actuelle de l'ensemble de réplicas.	
mongodb.rs.status	Renvoie l'état de l'ensemble de réplicas du point de vue du membre sur lequel la méthode est exécutée.	
mongodb.server.status	Renvoie l'état de la base de données.	
mongodb.sh.discovery	Renvoie la liste des shards découverts présents dans le cluster.	
mongodb.version	Renvoie la version du serveur de base de données.	
mqtt.get	S'abonne à un ou plusieurs sujets spécifiques (avec caractères génériques) du broker fourni et attend les publications.	MQTT
mssql.availability.group.get	Renvoie les groupes de disponibilité.	MSSQL (chargeable)
mssql.custom.query	Renvoie le résultat d'une requête personnalisée.	

Item key	Description	Plugin
mssql.db.get	Renvoie toutes les bases de données MSSQL disponibles.	
mssql.job.status.get	Renvoie l'état des tâches.	
mssql.last.backup.get	Renvoie l'heure de la dernière sauvegarde pour toutes les bases de données.	
mssql.local.db.get	Renvoie les bases de données qui participent à un groupe de disponibilité Always On et à un réplica (principal ou secondaire) et qui se trouvent sur le serveur auquel la connexion a été établie.	
mssql.mirroring.get	Renvoie les informations de mirroring.	
mssql.nonlocal.db.get	Renvoie les bases de données qui participent à un groupe de disponibilité Always On et à un réplica (principal ou secondaire) situés sur d'autres serveurs (la base de données n'est pas locale à l'instance SQL Server à laquelle la connexion a été établie).	
mssql.perfcounter.get	Renvoie les compteurs de performances.	
mssql.ping	Teste si une connexion est active ou non.	
mssql.quorum.get	Renvoie les informations de quorum.	
mssql.quorum.member.get	Renvoie les membres du quorum.	
mssql.replica.get	Renvoie les réplicas.	
mssql.version	Renvoie la version MSSQL.	
mysql.custom.query	Renvoie le résultat d'une requête personnalisée.	MySQL
mysql.db.discovery	Renvoie la liste des bases de données MySQL.	
mysql.db.size	La taille de la base de données en octets.	
mysql.get_status_variables	Valeurs des variables d'état globales.	
mysql.ping	Teste si une connexion est active ou non.	
mysql.replication.discovery	Renvoie la liste des réplications MySQL.	
mysql.replication.get_slave_status	L'état de la réplication.	
mysql.version	La version MySQL.	
net.dns.get	Effectue une requête DNS et renvoie des informations détaillées sur l'enregistrement DNS.	Network
nvml.device.count	Le nombre de périphériques GPU.	NVIDIA GPU (chargeable)
nvml.device.decoder.utilization	Utilisation du décodeur du périphérique GPU.	
nvml.device.ecc.mode	Mode ECC du périphérique GPU.	
nvml.device.encoder.stats.get	Statistiques de l'encodeur du périphérique GPU.	
nvml.device.encoder.utilization	Utilisation de l'encodeur du périphérique GPU.	
nvml.device.energy.consumption	Consommation d'énergie du périphérique GPU.	
nvml.device.errors.memory	Informations sur les erreurs mémoire ECC du périphérique GPU.	
nvml.device.errors.register	Informations sur les erreurs de registre ECC du périphérique GPU.	
nvml.device.fan.speed.avg	Vitesse moyenne du ventilateur du périphérique GPU.	
nvml.device.get	GPU découverts avec UUID et nom.	
nvml.device.graphics.frequency	Fréquence d'horloge graphique du périphérique GPU.	
nvml.device.memory.bar1.get	Informations sur la mémoire BAR1 du périphérique GPU.	
nvml.device.memory.fb.get	Informations sur la mémoire framebuffer du périphérique GPU.	
nvml.device.memory.frequency	Fréquence d'horloge mémoire du périphérique GPU.	
nvml.device.pci.utilization	Utilisation PCI du périphérique GPU.	
nvml.device.performance.state	État de performance du périphérique GPU.	
nvml.device.power.limit	Limite de puissance du périphérique GPU.	
nvml.device.power.usage	Utilisation de la puissance du périphérique GPU.	
nvml.device.serial	Numéro de série du périphérique GPU.	
nvml.device.sm.frequency	Fréquence d'horloge du multiprocesseur de flux du périphérique GPU.	
nvml.device.temperature	Température du périphérique GPU.	
nvml.device.utilization	Statistiques d'utilisation du périphérique GPU.	
nvml.device.video.frequency	Fréquence d'horloge vidéo du périphérique GPU.	
nvml.system.driver.version	La version du pilote NVIDIA.	
nvml.version	La version de la bibliothèque NVML.	
oracle.diskgroups.stats	Renvoie les statistiques des groupes de disques Automatic Storage Management (ASM).	Oracle
oracle.diskgroups.discovery	Renvoie la liste des groupes de disques ASM.	
oracle.archive.info	Les statistiques des journaux d'archive.	
oracle.cdb.info	Les informations sur les bases de données conteneurs (CDB).	
oracle.custom.query	Le résultat d'une requête personnalisée.	
oracle.datafiles.stats	Renvoie les statistiques des fichiers de données.	

Item key	Description	Plugin
oracle.db.discovery	Renvoie la liste des bases de données.	
oracle.fra.stats	Renvoie les statistiques de la zone de récupération rapide (FRA).	
oracle.instance.info	Les statistiques de l'instance.	
oracle.pdb.info	Les informations sur les bases de données enfichables (PDB).	
oracle.pdb.discovery	Renvoie la liste des PDB.	
oracle.pga.stats	Renvoie les statistiques de la Program Global Area (PGA).	
oracle.ping	Teste si une connexion à Oracle peut être établie.	
oracle.proc.stats	Renvoie les statistiques des processus.	
oracle.redolog.info	Les informations sur les fichiers journaux à partir du fichier de contrôle.	
oracle.sga.stats	Renvoie les statistiques de la System Global Area (SGA).	
oracle.sessions.stats	Renvoie les statistiques des sessions.	
oracle.sys.metrics	Renvoie un ensemble de valeurs de métriques système.	
oracle.sys.params	Renvoie un ensemble de valeurs de paramètres système.	
oracle.ts.stats	Renvoie les statistiques des tablespaces.	
oracle.ts.discovery	Renvoie une liste de tablespaces.	
oracle.user.info	Renvoie les informations sur l'utilisateur Oracle.	
oracle.version	Renvoie la version du serveur de base de données.	
pgsql.autovacuum.count	Le nombre de workers autovacuum.	PostgreSQL (chargeable)
pgsql.archive	Les informations sur les fichiers archivés.	
pgsql.bgwriter	Le nombre total de checkpoints pour le cluster de bases de données, réparti par type de checkpoint.	
pgsql.cache.hit	Le taux de réussite du cache tampon PostgreSQL.	
pgsql.connections	Renvoie les connexions par type.	
pgsql.custom.query	Renvoie le résultat d'une requête personnalisée.	
pgsql.db.age	L'ancienneté du FrozenXID le plus ancien de la base de données.	
pgsql.db.bloating_tables	Le nombre de tables gonflées par base de données.	
pgsql.db.discovery	La liste des bases de données PostgreSQL.	
pgsql.db.size	La taille de la base de données en octets.	
pgsql.dbstat	Collecte les statistiques par base de données.	
pgsql.dbstat.sum	Les données récapitulatives pour toutes les bases de données d'un cluster.	
pgsql.locks	Les informations sur les verrous accordés par base de données.	
pgsql.oldest.xid	L'ancienneté du XID le plus ancien.	
pgsql.ping	Teste si une connexion est active ou non.	
pgsql.queries	Métriques de requête par temps d'exécution.	
pgsql.replication.count	Le nombre de serveurs de secours.	
pgsql.replication.process	Le décalage de flush, le décalage d'écriture et le décalage de replay pour chaque processus d'envoi.	
pgsql.replication.process.discovery	La découverte des noms de processus de réplication.	
pgsql.replication.recovery_role	L'état de récupération.	
pgsql.replication.status	L'état de la réplication.	
pgsql.replication_lag.b	Le retard de réplication en octets.	
pgsql.replication_lag.sec	Le retard de réplication en secondes.	
pgsql.uptime	Le temps de fonctionnement de PostgreSQL en millisecondes.	
pgsql.version	Renvoie la version de PostgreSQL.	
pgsql.wal.stat	Les statistiques WAL.	
redis.config	Obtient les paramètres de configuration d'une instance Redis correspondant au modèle.	Redis
redis.info	Obtient la sortie de la commande INFO.	
redis.ping	Teste si une connexion est active ou non.	
redis.slowlog.count	Le nombre d'entrées du journal des requêtes lentes depuis le démarrage de Redis.	
smart.attribute.discovery	Renvoie une liste d'attributs de périphériques S.M.A.R.T.	S.M.A.R.T.
smart.disk.discovery	Renvoie une liste de périphériques S.M.A.R.T.	
smart.disk.get	Renvoie toutes les propriétés disponibles des périphériques S.M.A.R.T.	
systemd.unit.get	Renvoie toutes les propriétés d'une unité systemd.	Systemd
systemd.unit.info	Informations sur l'unité systemd.	
systemd.unit.discovery	La liste des unités systemd et leurs détails.	
web.certificate.get	Valide les certificats et renvoie les détails du certificat.	Certificats Web

Détails de la clé d'élément

Les paramètres sans crochets angulaires sont obligatoires. Les paramètres marqués par des crochets angulaires **< >** sont facultatifs.

ceph.df.details[connString,<user>,<apikey>]

 L'utilisation des données du cluster et leur répartition entre les pools.
 Valeur de retour : *objet JSON*.

Paramètres :

- **connString** - l'URI ou le nom de session ;
- **user, apikey** - le nom d'utilisateur et la clé API pour se connecter à Ceph.

ceph.osd.stats[connString,<user>,<apikey>]

 Statistiques agrégées et par OSD.
 Valeur de retour : *objet JSON*.

Paramètres :

- **connString** - l'URI ou le nom de session ;
- **user, apikey** - le nom d'utilisateur et la clé API pour se connecter à Ceph.

ceph.osd.discovery[connString,<user>,<apikey>]

 La liste des OSD découverts. Utilisé pour la découverte de bas niveau.
 Valeur de retour : *objet JSON*.

Paramètres :

- **connString** - l'URI ou le nom de session ;
- **user, apikey** - le nom d'utilisateur et la clé API pour se connecter à Ceph.

ceph.osd.dump[connString,<user>,<apikey>]

 Les seuils d'utilisation et les statuts des OSD.
 Valeur de retour : *objet JSON*.

Paramètres :

- **connString** - l'URI ou le nom de session ;
- **user, apikey** - le nom d'utilisateur et la clé API pour se connecter à Ceph.

ceph.ping[connString,<user>,<apikey>]

 Teste si une connexion à Ceph peut être établie.
 Valeur de retour : *0* - la connexion est interrompue (en cas d'erreur, y compris les problèmes d'AUTH et de configuration) ; *1* - la connexion a réussi.

Paramètres :

- **connString** - l'URI ou le nom de session ;
- **user, apikey** - le nom d'utilisateur et la clé API pour se connecter à Ceph.

ceph.pool.discovery[connString,<user>,<apikey>]

 La liste des pools découverts. Utilisé pour la découverte de bas niveau.
 Valeur de retour : *objet JSON*.

Paramètres :

- **connString** - l'URI ou le nom de session ;
- **user, apikey** - le nom d'utilisateur et la clé API pour se connecter à Ceph.

ceph.status[connString,<user>,<apikey>]

 L'état global du cluster.
 Valeur de retour : *objet JSON*.

Paramètres :

- **connString** - l'URI ou le nom de session ;
- **user, apikey** - le nom d'utilisateur et la clé API pour se connecter à Ceph.

docker.container_info[<ID>,<info>]

 Informations de bas niveau sur un conteneur.
 Valeur de retour : la sortie de l'appel API ContainerInspect, sérialisée au format JSON.

Paramètres :

- **ID** - l'ID ou le nom du conteneur ;
- **info** - la quantité d'informations renvoyées. Valeurs prises en charge : *short* (par défaut) ou *full*.

L'utilisateur Agent 2 (« zabbix ») doit être ajouté au groupe « docker » afin de disposer des privilèges suffisants. Sinon, la vérification échouera.

docker.container_stats[<ID>]

 Les statistiques d'utilisation des ressources du conteneur.
 Valeur de retour : la sortie de l'appel API ContainerStats ainsi que le pourcentage d'utilisation du CPU, sérialisés au format JSON.

Paramètre :

- **ID** - l'ID ou le nom du conteneur.

L'utilisateur Agent 2 (« zabbix ») doit être ajouté au groupe « docker » afin de disposer des privilèges suffisants. Sinon, la vérification échouera.

docker.containers

 La liste des conteneurs.
 Valeur de retour : la sortie de l'appel API ContainerList sérialisée au format JSON.

L'utilisateur Agent 2 (« zabbix ») doit être ajouté au groupe « docker » afin de disposer des privilèges suffisants. Sinon, la vérification échouera.

docker.containers.discovery[<options>]

 Renvoie la liste des conteneurs. Utilisé pour la découverte de bas niveau.
 Valeur de retour : *objet JSON*.

Paramètre :

- **options** - indique si tous les conteneurs ou uniquement les conteneurs en cours d'exécution doivent être découverts. Valeurs prises en charge : *true* - renvoyer tous les conteneurs ; *false* - renvoyer uniquement les conteneurs en cours d'exécution (par défaut).

L'utilisateur Agent 2 (« zabbix ») doit être ajouté au groupe « docker » afin de disposer des privilèges suffisants. Sinon, la vérification échouera.

docker.data.usage

 Informations sur l'utilisation actuelle des données.
 Valeur de retour : la sortie de l'appel API SystemDataUsage sérialisée au format JSON.

L'utilisateur Agent 2 (« zabbix ») doit être ajouté au groupe « docker » afin de disposer des privilèges suffisants. Sinon, la vérification échouera.

docker.images

 Renvoie la liste des images.
 Valeur de retour : la sortie de l'appel API ImageList sérialisée au format JSON.

L'utilisateur Agent 2 (« zabbix ») doit être ajouté au groupe « docker » afin de disposer des privilèges nécessaires. Sinon, la vérification échouera.

docker.images.discovery

 Renvoie la liste des images. Utilisé pour la découverte de bas niveau.
 Valeur de retour : *objet JSON*.

L'utilisateur Agent 2 ('zabbix') doit être ajouté au groupe 'docker' afin de disposer des privilèges suffisants. Sinon, la vérification échouera.

docker.info

 Les informations du système.
 Valeur de retour : la sortie de l'appel API SystemInfo sérialisée au format JSON.

L'utilisateur d'Agent 2 (« zabbix ») doit être ajouté au groupe « docker » afin de disposer des privilèges suffisants. Sinon, la vérification échouera.

docker.ping

 Teste si le démon Docker est actif ou non.
 Valeur de retour : *1* - la connexion est active ; *0* - la connexion est interrompue.

L'utilisateur Agent 2 ('zabbix') doit être ajouté au groupe 'docker' afin de disposer des privilèges suffisants. Sinon, la vérification échouera.

ember.get[<uri>,<path>]

 Renvoie le résultat du périphérique requis.
 Valeur de retour : *objet JSON*.

Paramètres :

- **uri** - URI du périphérique Ember+. Par défaut : 127.0.0.1:9998;
- **path** - Chemin OID vers le périphérique. Vide par défaut, renvoie les données de la collection racine.

memcached.ping[connString,<user>,<password>]

 Teste si une connexion est active ou non.
 Valeur de retour : *1* - la connexion est active ; *0* - la connexion est interrompue (en cas d'erreur, y compris les problèmes d'AUTH et de configuration).

Paramètres :

- **connString** - l'URI ou le nom de session ;
- **user, password** - les identifiants de connexion Memcached.

memcached.stats[connString,<user>,<password>,<type>]

 Obtient la sortie de la commande STATS.
 Valeur de retour : *JSON* - la sortie est sérialisée au format JSON.

Paramètres :

- **connString** - l'URI ou le nom de session ;
- **user, password** - les identifiants de connexion Memcached ;
- **type** - le type de statistiques à renvoyer : *items*, *sizes*, *slabs* ou *settings* (vide par défaut, renvoie les statistiques générales).

mongodb.collection.stats[connString,<user>,<password>,<database>,collection]

 Renvoie diverses statistiques de stockage pour une collection donnée.
 Valeur de retour : *objet JSON*.

Paramètres :

- **connString** - l'URI ou le nom de session ;
- **user, password** - les identifiants de connexion MongoDB ;
- **database** - le nom de la base de données (par défaut : admin) ;
- **collection** - le nom de la collection.

mongodb.cfg.discovery[connString,<user>,<password>]

 Renvoie une liste des serveurs de configuration découverts.
 Valeur de retour : *objet JSON*.

Paramètres :

- **connString** - l'URI ou le nom de session ;
- **user, password** - les identifiants de connexion MongoDB.

mongodb.collections.discovery[connString,<user>,<password>]

 Renvoie une liste des collections découvertes. Utilisé pour la découverte de bas niveau.
 Valeur de retour : *objet JSON*.

Paramètres :

- **connString** - l'URI ou le nom de session ;
- **user, password** - les identifiants de connexion MongoDB.

mongodb.collections.usage[connString,<user>,<password>]

 Renvoie les statistiques d'utilisation des collections.
 Valeur de retour : *objet JSON*.

Paramètres :

- **connString** - l'URI ou le nom de session ;
- **user, password** - les identifiants de connexion MongoDB.

mongodb.connpool.stats[connString,<user>,<password>]

 Renvoie des informations concernant les connexions sortantes ouvertes depuis l'instance de base de données actuelle vers d'autres membres du cluster shardé ou du replica set.
 Valeur de retour : *objet JSON*.

Paramètres :

- **connString** - l'URI ou le nom de session ;
- **user, password** - les identifiants de connexion MongoDB.

mongodb.db.stats[connString,<user>,<password>,<database>]

 Renvoie les statistiques reflétant l'état d'un système de base de données donné.
 Valeur de retour : *objet JSON*.

Paramètres :

- **connString** - l'URI ou le nom de session ;
- **user, password** - les identifiants de connexion MongoDB ;
- **database** - le nom de la base de données (par défaut : admin).

mongodb.db.discovery[connString,<user>,<password>]

 Renvoie une liste des bases de données découvertes. Utilisé pour la découverte de bas niveau.
 Valeur de retour : *objet JSON*.

Paramètres :

- **connString** - l'URI ou le nom de session ;
- **user, password** - les identifiants de connexion MongoDB.

mongodb.jumbo_chunks.count[connString,<user>,<password>]

 Renvoie le nombre de chunks jumbo.
 Valeur de retour : *objet JSON*.

Paramètres :

- **connString** - l'URI ou le nom de session ;
- **user, password** - les identifiants de connexion MongoDB.

mongodb.oplog.stats[connString,<user>,<password>]

 Renvoie l'état du jeu de réplicas à l'aide des données interrogées depuis l'oplog.
 Valeur de retour : *objet JSON*.

Paramètres :

- **connString** - l'URI ou le nom de session ;
- **user, password** - les identifiants de connexion MongoDB.

mongodb.ping[connString,<user>,<password>]

 Teste si une connexion est active ou non.
 Valeur de retour : *1* - la connexion est active ; *0* - la connexion est interrompue (si une erreur est présente, y compris des problèmes d'AUTH et de configuration).

Paramètres :

- **connString** - l'URI ou le nom de session ;
- **user, password** - les identifiants de connexion MongoDB.

mongodb.rs.config[connString,<user>,<password>]

 Renvoie la configuration actuelle du jeu de réplicas.
 Valeur de retour : *objet JSON*.

Paramètres :

- **connString** - l'URI ou le nom de session ;
- **user, password** - les identifiants de connexion MongoDB.

mongodb.rs.status[connString,<user>,<password>]

 Renvoie l'état du replica set du point de vue du membre sur lequel la méthode est exécutée.
 Valeur de retour : *objet JSON*.

Paramètres :

- **connString** - l'URI ou le nom de session ;
- **user, password** - les identifiants de connexion MongoDB.

mongodb.server.status[connString,<user>,<password>]

 Renvoie l'état de la base de données.
 Valeur de retour : *objet JSON*.

Paramètres :

- **connString** - l'URI ou le nom de session ;
- **user, password** - les identifiants de connexion MongoDB.

mongodb.sh.discovery[connString,<user>,<password>]

 Renvoie la liste des shards découverts présents dans le cluster.
 Valeur de retour : *objet JSON*.

Paramètres :

- **connString** - l'URI ou le nom de session ;
- **user, password** - les identifiants de connexion MongoDB.

mongodb.version[connString,<user>,<password>]

 Renvoie la version du serveur de base de données.
 Valeur de retour : *String*.

Paramètres :

- **connString** - l'URI ou le nom de session ;
- **user, password** - les identifiants de connexion MongoDB.

mqtt.get[<broker url>,topic,<user>,<password>]

 S'abonne à un topic spécifique ou à des topics (avec des caractères génériques) du broker fourni et attend les publications.
 Valeur de retour: selon le contenu du topic. Si des caractères génériques sont utilisés, renvoie le contenu du topic au format JSON.

Paramètres:

- **broker url** - l'URL du broker MQTT au format `protocol://host:port` sans paramètres de requête (protocoles pris en charge: `tcp`, `ssl`, `ws`). Si aucune valeur n'est spécifiée, l'agent utilisera `tcp://localhost:1883`. Si un protocole ou un port sont omis, le protocole par défaut (`tcp`) ou le port (1883) seront utilisés;
- **topic** - le topic MQTT (obligatoire). Les caractères génériques (+,#) sont pris en charge;
- **user, password** - les identifiants d'authentification (si nécessaire).

Commentaires:

- L'élément doit être configuré comme une vérification active (type d'élément 'Zabbix agent (active)');
- Les certificats de chiffrement TLS peuvent être utilisés en les enregistrant dans un emplacement par défaut (par exemple, le répertoire `/etc/ssl/certs/` pour Ubuntu). Pour TLS, utilisez le schéma `tls://`.

mssql.availability.group.get[URI,<user>,<password>]

 Renvoie les groupes de disponibilité.
 Valeur de retour : *objet JSON*.

Paramètres :

- **URI** - URI du serveur MSSQL (le seul schéma pris en charge est `sqlserver://`). Les informations d'identification intégrées seront ignorées. Il est possible de spécifier un nom d'instance comme partie de l'URI, par exemple : `sqlserver://localhost/InstanceName` (sans port). Si le port est spécifié, le nom d'instance est ignoré ;
- **user, password** - nom d'utilisateur, mot de passe à envoyer au serveur MSSQL protégé.

Pour plus d'informations, consultez le fichier readme du plugin MSSQL.

mssql.custom.query[URI,<user>,<password>,queryName,<args...>]

 Renvoie le résultat d'une requête personnalisée.
 Valeur de retour : *objet JSON*.

Paramètres :

- **URI** - URI du serveur MSSQL (le seul schéma pris en charge est `sqlserver://`). Les informations d'identification intégrées seront ignorées. Il est possible de spécifier un nom d'instance comme partie de l'URI, par ex. : `sqlserver://localhost/InstanceNa` (sans port). Si le port est spécifié, le nom d'instance est ignoré ;
- **user, password** - nom d'utilisateur, mot de passe à envoyer au serveur MSSQL protégé ;
- **queryName** - nom d'une requête personnalisée configurée dans `Plugins.MSSQL.CustomQueriesDir` sans l'extension .sql ;
- **args** - un ou plusieurs arguments séparés par des virgules à transmettre à une requête.

Pour plus d'informations, consultez le fichier readme du plugin MSSQL.

mssql.db.get

 Renvoie toutes les bases de données MSSQL disponibles.
 Valeur de retour : *objet JSON*.

Pour plus d'informations, consultez le fichier readme du plugin MSSQL.

mssql.job.status.get

 Renvoie le statut des tâches.
 Valeur de retour : *objet JSON*.

Pour plus d'informations, consultez le fichier readme du plugin MSSQL.

mssql.last.backup.get

 Renvoie l'heure de la dernière sauvegarde pour toutes les bases de données.
 Valeur de retour : *objet JSON*.

Pour plus d'informations, consultez le fichier readme du plugin MSSQL.

mssql.local.db.get

 Renvoie les bases de données qui participent à un groupe de disponibilité Always On et à un réplica (primaire ou secondaire), et qui sont situées sur le serveur auquel la connexion a été établie.
 Valeur de retour : *objet JSON*.

Pour plus d'informations, consultez le fichier readme du plugin MSSQL.

mssql.mirroring.get

 Renvoie les informations de mise en miroir.
 Valeur de retour : *objet JSON*.

Pour plus d'informations, consultez le fichier readme du plugin MSSQL.

mssql.nonlocal.db.get

 Renvoie les bases de données qui participent à un groupe de disponibilité Always On et à un réplica (primaire ou secondaire) situé sur d'autres serveurs (la base de données n'est pas locale à l'instance SQL Server à laquelle la connexion a été établie).
 Valeur de retour : *objet JSON*.

Pour plus d'informations, consultez le fichier readme du plugin MSSQL.

mssql.perfcounter.get

 Renvoie les compteurs de performance.
 Valeur de retour : *objet JSON*.

Pour plus d'informations, consultez le fichier readme du plugin MSSQL.

mssql.ping

 Ping la base de données. Vérifie si la connexion est correctement configurée.
 Valeur de retour : *1* - disponible, *0* - indisponible.

Pour plus d'informations, consultez le fichier readme du plugin MSSQL.

mssql.quorum.get

 Renvoie les informations de quorum.
 Valeur de retour : *objet JSON*.

Pour plus d'informations, consultez le fichier readme du plugin MSSQL.

mssql.quorum.member.get

 Renvoie les membres du quorum.
 Valeur de retour : *objet JSON*.

Pour plus d'informations, consultez le fichier readme du plugin MSSQL.

mssql.replica.get

 Renvoie les réplicas.
 Valeur de retour : *objet JSON*.

Pour plus d'informations, consultez le fichier readme du plugin MSSQL.

mssql.version

 Renvoie la version de MSSQL.
 Valeur de retour : *String*.

Pour plus d'informations, consultez le fichier readme du plugin MSSQL.

mysql.custom.query[connString,<user>,<password>,queryName,<args...>]

 Renvoie le résultat d'une requête personnalisée.
 Valeur de retour : *objet JSON*.

Paramètres :

- **connString** - URI ou nom de session ;
- **user, password** - identifiants de connexion MySQL ;
- **queryName** - nom d'une requête personnalisée ; doit correspondre au nom du fichier SQL sans extension ;
- **args** - un ou plusieurs arguments séparés par des virgules à transmettre à une requête.

Pour plus d'informations, consultez le fichier readme du plugin MySQL, section *Requêtes personnalisées*.

mysql.db.discovery[connString,<user>,<password>]

 Renvoie la liste des bases de données MySQL. Utilisé pour la découverte de bas niveau.
 Valeur de retour : le résultat de la requête SQL « show databases » au format JSON LLD.

Paramètres :

- **connString** - l'URI ou le nom de session ;
- **user, password** - les identifiants de connexion MySQL.

mysql.db.size[connString,<user>,<password>,<database name>]

 La taille de la base de données en octets.
 Valeur de retour : résultat de la requête SQL "select coalesce(sum(data_length + index_length),0) as size from information_schema.tables where table_schema=?" pour la base de données spécifiée, en octets.

Paramètres :

- **connString** - l'URI ou le nom de session ;
- **user, password** - les identifiants de connexion MySQL ;
- **database name** - le nom de la base de données.

mysql.get_status_variables[connString,<user>,<password>]

 Valeurs des variables d'état globales.
 Valeur de retour : résultat de la requête SQL « show global status » au format JSON.

Paramètres :

- **connString** - l'URI ou le nom de session ;
- **user, password** - les identifiants de connexion MySQL.

mysql.ping[connString,<user>,<password>]

 Teste si une connexion est active ou non.
 Valeur de retour : *1* - la connexion est active ; *0* - la connexion est interrompue (si une erreur est présente, y compris les problèmes d'AUTH et de configuration).

Paramètres :

- **connString** - l'URI ou le nom de session ;
- **user, password** - les identifiants de connexion MySQL.

mysql.replication.discovery[connString,<user>,<password>]

 Renvoie la liste des réplications MySQL. Utilisé pour la découverte de bas niveau.
 Valeur de retour : le résultat de la requête SQL `SHOW SLAVE STATUS` ou `SHOW REPLICA STATUS` au format JSON LLD.

Paramètres :

- **connString** - l'URI ou le nom de session ;
- **user, password** - les identifiants de connexion MySQL.

mysql.replication.get_slave_status[connString,<user>,<password>,<master host>]

 Le statut de réplication.
 Valeur de retour : résultat de la requête SQL `SHOW SLAVE STATUS` ou `SHOW REPLICA STATUS` au format JSON.

Paramètres :

- **connString** - l'URI ou le nom de session ;
- **user, password** - les identifiants de connexion MySQL ;
- **master host** - le nom d'hôte du maître de réplication. Si aucun n'est trouvé, une erreur est renvoyée. Si ce paramètre n'est pas spécifié, tous les hôtes sont renvoyés.

mysql.version[connString,<user>,<password>]

 La version de MySQL.
 Valeur de retour : *String* (avec la version de l'instance MySQL).

Paramètres :

- **connString** - l'URI ou le nom de session ;
- **user, password** - les identifiants de connexion MySQL.

net.dns.get[<ip>,name,<type>,<timeout>,<count>,<protocol>,"<flags>"]

Effectue une requête DNS et renvoie des informations détaillées sur l'enregistrement DNS.
 Cet élément est une version étendue de l'élément agent Zabbix `net.dns.record`, avec davantage de types d'enregistrements et la prise en charge de flags personnalisables.
 Valeurs de retour: *objet JSON*

Paramètres:

- **ip** - l'adresse IP du serveur DNS (laissez vide pour utiliser le serveur DNS par défaut);
- **name** - le nom DNS à interroger;
- **type** - le type d'enregistrement à interroger (la valeur par défaut est *SOA*);
- **timeout** - le délai d'attente de la requête, en secondes (la valeur par défaut est 1 seconde);
- **count** - le nombre de tentatives pour la requête (la valeur par défaut est 2);
- **protocol** - le protocole utilisé pour effectuer les requêtes DNS: *udp* (par défaut) ou *tcp*;
- **flags** - un ou plusieurs arguments séparés par des virgules à transmettre à une requête.

Commentaires:

- Les valeurs possibles pour `type` sont: *A*, *NS*, *MD*, *MF*, *CNAME*, *SOA*, *MB*, *MG*, *MR*, *NULL*, *PTR*, *HINFO*, *MINFO*, *MX*, *TXT*, *RP*, *AFSDB*, *X25*, *ISDN*, *RT*, *NSAPPTR*, *SIG*, *KEY*, *PX*, *GPOS*, *AAAA*, *LOC*, *NXT*, *EID*, *NIMLOC*, *SRV*, *ATMA*, *NAPTR*, *KX*, *CERT*, *DNAME*, *OPT*, *APL*, *DS*, *SSHFP*, *IPSECKEY*, *RRSIG*, *NSEC*, *DNSKEY*, *DHCID*, *NSEC3*, *NSEC3PARAM*, *TLSA*, *SMIMEA*, *HIP*, *NINFO*, *RKEY*,

TALINK, *CDS*, *CDNSKEY*, *OPENPGPKEY*, *CSYNC*, *ZONEMD*, *SVCB*, *HTTPS*, *SPF*, *UINFO*, *UID*, *GID*, *UNSPEC*, *NID*, *L32*, *L64*, *LP*, *EUI48*, *EUI64*, *URI*, *CAA*, *AVC*, *AMTRELAY*. Notez que les valeurs doivent être en majuscules uniquement; les valeurs en minuscules ou en casse mixte ne sont pas prises en charge.

- Pour les recherches DNS inversées (lorsque `type` est défini sur *PTR*), vous pouvez fournir le nom DNS au format inversé ou non inversé (voir les exemples ci-dessous). Notez que lorsqu'un enregistrement PTR est demandé, le nom DNS est en réalité une adresse IP.
- Les valeurs possibles pour `flags` sont:
  - *cdflag* ou *nocdflag* (par défaut);
  - *rdflag* (par défaut) ou *nordflag*;
  - *dnssec* ou *nodnssec* (par défaut);
  - *nsid* ou *nonsid* (par défaut);
  - *edns0* (par défaut) ou *noedns0*;
  - *aaflag* ou *noaaflag* (par défaut);
  - *adflag* ou *noadflag* (par défaut). Les `flags` *dnssec* et *nsid* ne peuvent pas être utilisés avec *noedns0*, car ils nécessitent tous deux *edns0*. Notez que les valeurs doivent être en minuscules uniquement; les valeurs en majuscules ou en casse mixte ne sont pas prises en charge.
- Les noms de domaine internationalisés ne sont pas pris en charge, veuillez utiliser à la place des noms encodés en IDNA.
- La sortie est un objet contenant des informations sur l'enregistrement DNS en fonction des paramètres fournis (voir plus de détails).

Exemples:

```
net.dns.get[192.0.2.1,example.com,DNSKEY,3,3,tcp,"cdflag,rdflag,nsid"]

net.dns.get[,198.51.100.1,PTR,,,,"cdflag,rdflag,nsid"]
net.dns.get[,1.100.51.198.in-addr.arpa,PTR,,,,"cdflag,rdflag,nsid"]

net.dns.get[,2a00:1450:400f:800::200e,PTR,,,,"cdflag,rdflag,nsid"]
net.dns.get[,e.0.0.2.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.8.0.f.0.0.4.0.5.4.1.0.0.a.2.ip6.arpa,PTR,,,,"cdflag,rdfla
```

nvml.device.count

 Le nombre de périphériques GPU.
 Valeur de retour : *Integer*.

Pour plus d'informations, consultez le fichier readme du plugin NVIDIA GPU.

nvml.device.decoder.utilization[<deviceUUID>]

 Utilisation du décodeur du périphérique GPU en pourcentage.
 Valeur de retour : *Integer*.

Paramètre :

- **deviceUUID** - UUID du périphérique GPU.

Pour plus d'informations, consultez le fichier readme du plugin NVIDIA GPU.

nvml.device.ecc.mode[<deviceUUID>]

 Informations sur le mode ECC du périphérique GPU (actuel, en attente).
 Valeur de retour : *JSON*.

Paramètre :

- **deviceUUID** - UUID du périphérique GPU.

Pour plus d'informations, consultez le fichier readme du plugin NVIDIA GPU.

nvml.device.encoder.stats.get[<deviceUUID>]

 Statistiques de l'encodeur du périphérique GPU.
 Valeur de retour : *JSON*.

Paramètre :

- **deviceUUID** - UUID du périphérique GPU.

Pour plus d'informations, consultez le fichier readme du plugin NVIDIA GPU.

nvml.device.encoder.utilization[<deviceUUID>]

 Utilisation de l'encodeur du périphérique GPU en pourcentage.
 Valeur de retour : *Integer*.

Paramètre :

- **deviceUUID** - UUID du périphérique GPU.

Pour plus d’informations, consultez le fichier readme du plugin NVIDIA GPU.

nvml.device.energy.consumption[<deviceUUID>]

 Consommation totale d’énergie du périphérique GPU en millijoules (mJ) depuis le dernier rechargement du pilote.
 Valeur de retour : *Integer*.

Paramètre :

- **deviceUUID** - UUID du périphérique GPU.

Pour plus d’informations, consultez le fichier readme du plugin NVIDIA GPU.

nvml.device.errors.memory[<deviceUUID>]

 Informations sur les erreurs de mémoire ECC du périphérique GPU (corrigées, non corrigées).
 Valeur de retour : *JSON*.

Paramètre :

- **deviceUUID** - UUID du périphérique GPU.

Pour plus d’informations, consultez le fichier readme du plugin NVIDIA GPU.

nvml.device.errors.register[<deviceUUID>]

 Informations sur les erreurs de registre ECC du périphérique GPU (corrigées, non corrigées).
 Valeur de retour : *JSON*.

Paramètre :

- **deviceUUID** - UUID du périphérique GPU.

Pour plus d’informations, consultez le fichier readme du plugin NVIDIA GPU.

nvml.device.fan.speed.avg[<deviceUUID>]

 Vitesse moyenne du ventilateur du périphérique GPU en pourcentage de la vitesse maximale.
 Valeur de retour : *Integer*.

Paramètre :

- **deviceUUID** - UUID du périphérique GPU.

Pour plus d’informations, consultez le fichier readme du plugin NVIDIA GPU.

nvml.device.get

 GPU découverts avec UUID et nom.
 Valeur de retour : *JSON*.

Pour plus d’informations, consultez le fichier readme du plugin NVIDIA GPU.

nvml.device.graphics.frequency[<deviceUUID>]

 Fréquence d’horloge graphique du périphérique GPU en MHz.
 Valeur de retour : *Integer*.

Paramètre :

- **deviceUUID** - UUID du périphérique GPU.

Pour plus d’informations, consultez le fichier readme du plugin NVIDIA GPU.

nvml.device.memory.fb.get[<deviceUUID>]

 Statistiques de la mémoire framebuffer du périphérique GPU (totale, réservée, libre, utilisée).
 Valeur de retour : *JSON*.

Paramètre :

- **deviceUUID** - UUID du périphérique GPU.

Pour plus d’informations, consultez le fichier readme du plugin NVIDIA GPU.

nvml.device.memory.bar1.get[<deviceUUID>]

 Statistiques de la mémoire BAR1 du périphérique GPU (total, libre, utilisé).
 Valeur de retour : *JSON*.

Paramètre :

- **deviceUUID** - UUID du périphérique GPU.

Pour plus d’informations, consultez le fichier readme du plugin NVIDIA GPU.

nvml.device.memory.frequency[<deviceUUID>]

 Fréquence d’horloge de la mémoire du périphérique GPU en MHz.
 Valeur de retour : *Integer*.

Paramètre :

- **deviceUUID** - UUID du périphérique GPU.

Pour plus d'informations, consultez le fichier readme du plugin NVIDIA GPU.

nvml.device.pci.utilization[<deviceUUID>]

 Utilisation PCI du périphérique GPU (débit de transmission/réception en Ko/s).
 Valeur de retour : *JSON*.

Paramètre :

- **deviceUUID** - UUID du périphérique GPU.

Pour plus d'informations, consultez le fichier readme du plugin NVIDIA GPU.

nvml.device.performance.state[<deviceUUID>]

 État de performance du périphérique GPU.
 Valeur de retour : *Integer*.

Paramètre :

- **deviceUUID** - UUID du périphérique GPU.

Pour plus d'informations, consultez le fichier readme du plugin NVIDIA GPU.

nvml.device.power.limit[<deviceUUID>]

 Limite de puissance du périphérique GPU en milliwatts.
 Valeur de retour : *Integer*.

Paramètre :

- **deviceUUID** - UUID du périphérique GPU.

Pour plus d'informations, consultez le fichier readme du plugin NVIDIA GPU.

nvml.device.power.usage[<deviceUUID>]

 Consommation électrique actuelle du périphérique GPU en milliwatts.
 Valeur de retour : *Integer*.

Paramètre :

- **deviceUUID** - UUID du périphérique GPU.

Pour plus d'informations, consultez le fichier readme du plugin NVIDIA GPU.

nvml.device.serial[<deviceUUID>]

 Numéro de série du périphérique GPU.
 Valeur de retour : *String*.

Paramètre :

- **deviceUUID** - UUID du périphérique GPU.

Pour plus d'informations, consultez le fichier readme du plugin NVIDIA GPU.

nvml.device.sm.frequency[<deviceUUID>]

 Fréquence d'horloge du multiprocesseur de flux du périphérique GPU en MHz.
 Valeur de retour : *Integer*.

Paramètre :

- **deviceUUID** - UUID du périphérique GPU.

Pour plus d'informations, consultez le fichier readme du plugin NVIDIA GPU.

nvml.device.temperature[<deviceUUID>]

 Température du périphérique GPU en degrés Celsius.
 Valeur de retour : *Integer*.

Paramètre :

- **deviceUUID** - UUID du périphérique GPU.

Pour plus d'informations, consultez le fichier readme du plugin NVIDIA GPU.

nvml.device.utilization[<deviceUUID>]

 Statistiques d'utilisation du périphérique GPU (utilisation du GPU/de la mémoire en pourcentage).
 Valeur de retour : *JSON*.

Paramètre :

- **deviceUUID** - UUID du périphérique GPU.

Pour plus d'informations, consultez le fichier readme du plugin NVIDIA GPU.

nvml.device.video.frequency[<deviceUUID>]

 Fréquence d'horloge vidéo du périphérique GPU en MHz.
 Valeur de retour : *Integer*.

Paramètre :

- **deviceUUID** - UUID du périphérique GPU.

Pour plus d'informations, consultez le fichier readme du plugin NVIDIA GPU.

nvml.system.driver.version

 La version du pilote NVIDIA.
 Valeur de retour : *String*.

Pour plus d'informations, consultez le fichier readme du plugin NVIDIA GPU.

nvml.version

 La version de la bibliothèque NVML.
 Valeur de retour : *String*.

Pour plus d'informations, consultez le fichier readme du plugin NVIDIA GPU.

oracle.diskgroups.stats[connString,<user>,<password>,<service>,<diskgroup>]

 Renvoie les statistiques des groupes de disques Automatic Storage Management (ASM).
 Valeur de retour : *objet JSON*.

Paramètres :

- **connString** - peut être :
  - l'URI - si aucune session portant le nom indiqué n'est trouvée et que `Plugins.Oracle.ResolveTNS` est défini sur false. S'il contient un schéma (par ex. "tcp://"), un port (par ex. 1521) ou les deux, l'option ResolveTNS ne sera pas prise en compte et il sera de toute façon traité comme un URI ;
  - le nom de session - si un tel nom est trouvé dans le fichier de configuration du plugin ;
  - la clé TNS - si l'option ResolveTNS est définie sur true et qu'aucune des conditions ci-dessus ne s'applique ;
  - la valeur TNS - si elle commence par une parenthèse ouvrante "(" (espaces initiaux ignorés). Dans ce cas, l'option ResolveTNS n'est pas prise en compte.
- **user** - le nom d'utilisateur Oracle ; prend en charge l'ajout de l'une des options de connexion `as sysdba`, `as sysoper`, `as sysasm`, `as sysbackup`, `as sysdg`, `as syskm` ou `as sysrac` au format `user as sysdba` (une option de connexion n'est pas sensible à la casse et ne doit pas contenir d'espace final).
- **password** - le mot de passe Oracle ;
- **service** - le nom du service Oracle ;
- **diskgroup** - le nom du groupe de disques ASM à interroger.

oracle.diskgroups.discovery[connString,<user>,<password>,<service>]

 Renvoie la liste des groupes de disques ASM. Utilisé pour la découverte de bas niveau.
 Valeur de retour : *objet JSON*.

Paramètres :

- **connString** - peut être :
  - l'URI - si aucune session portant le nom donné n'est trouvée et que `Plugins.Oracle.ResolveTNS` est défini sur false. S'il contient un schéma (par ex. "tcp://"), un port (par ex. 1521) ou les deux, l'option ResolveTNS ne sera pas prise en compte et il sera de toute façon traité comme un URI ;
  - nom de session - si un tel nom est trouvé dans le fichier de configuration du plugin ;
  - clé TNS - si l'option ResolveTNS est définie sur true et qu'aucune des conditions ci-dessus ne s'applique ;
  - valeur TNS - si elle commence par une parenthèse ouvrante "(" (espaces initiaux ignorés). Dans ce cas, l'option ResolveTNS n'est pas prise en compte.
- **user** - le nom d'utilisateur Oracle ; prend en charge l'ajout de l'une des options de connexion `as sysdba`, `as sysoper`, `as sysasm`, `as sysbackup`, `as sysdg`, `as syskm` ou `as sysrac` au format `user as sysdba` (une option de connexion est insensible à la casse et ne doit pas contenir d'espace final).
- **password** - le mot de passe Oracle ;
- **service** - le nom du service Oracle.

oracle.archive.info[connString,<user>,<password>,<service>,<destination>]

 Les statistiques des journaux d'archive.
 Valeur de retour : *objet JSON*.

Paramètres :

- **connString** - peut être :

    - l'URI - si aucune session portant le nom indiqué n'est trouvée et que `Plugins.Oracle.ResolveTNS` est défini sur false. S'il contient un schéma (par ex., "tcp://"), un port (par ex., 1521) ou les deux, l'option ResolveTNS ne sera pas prise en compte et il sera de toute façon traité comme un URI ;
    - le nom de session - si un tel nom est trouvé dans le fichier de configuration du plugin ;
    - la clé TNS - si l'option ResolveTNS est définie sur true et qu'aucune des conditions ci-dessus ne s'applique ;
    - la valeur TNS - si elle commence par une parenthèse ouvrante "(" (espaces initiaux ignorés). Dans ce cas, l'option ResolveTNS n'est pas prise en compte.
- **user** - le nom d'utilisateur Oracle ; prend en charge l'ajout de l'une des options de connexion `as sysdba`, `as sysoper`, `as sysasm`, `as sysbackup`, `as sysdg`, `as syskm` ou `as sysrac` au format `user as sysdba` (une option de connexion n'est pas sensible à la casse et ne doit pas contenir d'espace final).
- **password** - le mot de passe Oracle ;
- **service** - le nom du service Oracle ;
- **destination** - le nom de la destination à interroger.

oracle.cdb.info[connString,<user>,<password>,<service>,<database>]

 Les informations sur les bases de données conteneur (CDB).
 Valeur de retour : *objet JSON*.

Paramètres :

- **connString** - peut être :
    - l'URI - si aucune session portant le nom indiqué n'est trouvée et que `Plugins.Oracle.ResolveTNS` est défini sur false. S'il contient un schéma (par exemple, "tcp://"), un port (par exemple, 1521) ou les deux, l'option ResolveTNS ne sera pas prise en compte et il sera de toute façon traité comme un URI ;
    - le nom de session - si un tel nom est trouvé dans le fichier de configuration du plugin ;
    - la clé TNS - si l'option ResolveTNS est définie sur true et qu'aucune des conditions ci-dessus ne s'applique ;
    - la valeur TNS - si elle commence par la parenthèse ouvrante "(" (espaces initiaux ignorés). Dans ce cas, l'option ResolveTNS n'est pas prise en compte.
- **user** - le nom d'utilisateur Oracle ; prend en charge l'ajout de l'une des options de connexion `as sysdba`, `as sysoper`, `as sysasm`, `as sysbackup`, `as sysdg`, `as syskm` ou `as sysrac` au format `user as sysdba` (une option de connexion n'est pas sensible à la casse et ne doit pas contenir d'espace final).
- **password** - le mot de passe Oracle ;
- **service** - le nom du service Oracle ;
- **database** - le nom de la base de données à interroger.

oracle.custom.query[connString,<user>,<password>,<service>,queryName,<args...>]

 Le résultat d'une requête personnalisée.
 Valeur de retour : *objet JSON*.

Paramètres :

- **connString** - peut être :
    - l'URI - si aucune session portant le nom indiqué n'est trouvée et que `Plugins.Oracle.ResolveTNS` est défini sur false. S'il contient un schéma (par ex., "tcp://"), un port (par ex., 1521) ou les deux, l'option ResolveTNS ne sera pas prise en compte et il sera de toute façon traité comme un URI ;
    - le nom de session - si un tel nom est trouvé dans le fichier de configuration du plugin ;
    - la clé TNS - si l'option ResolveTNS est définie sur true et qu'aucune des conditions ci-dessus ne s'applique ;
    - la valeur TNS - si elle commence par la parenthèse ouvrante "(" (espaces initiaux ignorés). Dans ce cas, l'option ResolveTNS n'est pas prise en compte.
- **user** - le nom d'utilisateur Oracle ; prend en charge l'ajout de l'une des options de connexion `as sysdba`, `as sysoper`, `as sysasm`, `as sysbackup`, `as sysdg`, `as syskm` ou `as sysrac` au format `user as sysdba` (une option de connexion n'est pas sensible à la casse et ne doit pas contenir d'espace final).
- **password** - le mot de passe Oracle ;
- **service** - le nom du service Oracle ;
- **queryName** - le nom d'une requête personnalisée ; doit correspondre au nom du fichier SQL sans extension ;
- **args** - un ou plusieurs arguments séparés par des virgules à transmettre à la requête.

Commentaires : - Les données renvoyées sont automatiquement converties en JSON. - Évitez de renvoyer directement du JSON depuis les requêtes, car il sera corrompu lorsque le plugin tentera de le convertir à nouveau en JSON.

oracle.datafiles.stats[connString,<user>,<password>,<service>]

 Renvoie les statistiques des fichiers de données.
 Valeur de retour : *objet JSON*.

Paramètres :

- **connString** - peut être :
    - l'URI - si aucune session portant le nom donné n'est trouvée et que `Plugins.Oracle.ResolveTNS` est défini sur false. S'il contient un schéma (par ex., "tcp://"), un port (par ex., 1521) ou les deux, l'option ResolveTNS ne sera pas

prise en compte et il sera de toute façon traité comme un URI ;
  - le nom de session - si un tel nom est trouvé dans le fichier de configuration du plugin ;
  - la clé TNS - si l'option ResolveTNS est définie sur true et qu'aucune des conditions ci-dessus ne s'applique ;
  - la valeur TNS - si elle commence par la parenthèse ouvrante "(" (espaces initiaux ignorés). Dans ce cas, l'option ResolveTNS n'est pas prise en compte.
- **user** - le nom d'utilisateur Oracle ; prend en charge l'ajout de l'une des options de connexion `as sysdba`, `as sysoper`, `as sysasm`, `as sysbackup`, `as sysdg`, `as syskm` ou `as sysrac` au format `user as sysdba` (une option de connexion ne tient pas compte de la casse et ne doit pas contenir d'espace final).
- **password** - le mot de passe Oracle ;
- **service** - le nom du service Oracle.

oracle.db.discovery[connString,<user>,<password>,<service>]

 Renvoie la liste des bases de données. Utilisé pour la découverte de bas niveau.
 Valeur de retour : *objet JSON*.

Paramètres :

- **connString** - peut être :
  - l'URI - si aucune session portant le nom donné n'est trouvée et que `Plugins.Oracle.ResolveTNS` est défini sur false. S'il contient un schéma (par ex. "tcp://"), un port (par ex. 1521) ou les deux, l'option ResolveTNS ne sera pas prise en compte et il sera de toute façon traité comme un URI ;
  - nom de session - si un tel nom est trouvé dans le fichier de configuration du plugin ;
  - clé TNS - si l'option ResolveTNS est définie sur true et qu'aucune des conditions ci-dessus ne s'applique ;
  - valeur TNS - si elle commence par la parenthèse ouvrante "(" (espaces initiaux ignorés). Dans ce cas, l'option ResolveTNS n'est pas prise en compte.
- **user** - le nom d'utilisateur Oracle ; prend en charge l'ajout de l'une des options de connexion `as sysdba`, `as sysoper`, `as sysasm`, `as sysbackup`, `as sysdg`, `as syskm` ou `as sysrac` au format `user as sysdba` (une option de connexion n'est pas sensible à la casse et ne doit pas contenir d'espace final).
- **password** - le mot de passe Oracle ;
- **service** - le nom du service Oracle.

oracle.fra.stats[connString,<user>,<password>,<service>]

 Renvoie les statistiques de la Fast Recovery Area (FRA).
 Valeur de retour : *objet JSON*.

Paramètres :

- **connString** - peut être :
  - l'URI - si aucune session portant le nom donné n'est trouvée et que `Plugins.Oracle.ResolveTNS` est défini sur false. S'il contient un schéma (par ex. "tcp://"), un port (par ex. 1521) ou les deux, l'option ResolveTNS ne sera pas prise en compte et il sera de toute façon traité comme un URI ;
  - le nom de session - si un tel nom est trouvé dans le fichier de configuration du plugin ;
  - la clé TNS - si l'option ResolveTNS est définie sur true et qu'aucune des conditions ci-dessus ne s'applique ;
  - la valeur TNS - si elle commence par la parenthèse ouvrante "(" (espaces initiaux ignorés). Dans ce cas, l'option ResolveTNS n'est pas prise en compte.
- **user** - le nom d'utilisateur Oracle ; prend en charge l'ajout de l'une des options de connexion `as sysdba`, `as sysoper`, `as sysasm`, `as sysbackup`, `as sysdg`, `as syskm` ou `as sysrac` au format `user as sysdba` (une option de connexion n'est pas sensible à la casse et ne doit pas contenir d'espace final).
- **password** - le mot de passe Oracle ;
- **service** - le nom du service Oracle.

oracle.instance.info[connString,<user>,<password>,<service>]

 Les statistiques de l'instance.
 Valeur de retour : *objet JSON*.

Paramètres :

- **connString** - peut être :
  - l'URI - si aucune session portant le nom indiqué n'est trouvée et que `Plugins.Oracle.ResolveTNS` est défini sur false. S'il contient un schéma (par ex., "tcp://"), un port (par ex., 1521) ou les deux, l'option ResolveTNS ne sera pas prise en compte et il sera de toute façon traité comme un URI ;
  - le nom de session - si un tel nom est trouvé dans le fichier de configuration du plugin ;
  - la clé TNS - si l'option ResolveTNS est définie sur true et qu'aucune des conditions ci-dessus ne s'applique ;
  - la valeur TNS - si elle commence par une parenthèse ouvrante "(" (espaces initiaux ignorés). Dans ce cas, l'option ResolveTNS n'est pas prise en compte.
- **user** - le nom d'utilisateur Oracle ; prend en charge l'ajout de l'une des options de connexion `as sysdba`, `as sysoper`, `as sysasm`, `as sysbackup`, `as sysdg`, `as syskm` ou `as sysrac` au format `user as sysdba` (une option de connexion n'est pas sensible à la casse et ne doit pas contenir d'espace final).
- **password** - le mot de passe Oracle ;

- **service** - le nom du service Oracle.

oracle.pdb.info[connString,<user>,<password>,<service>,<database>]

 Les informations sur les bases de données enfichables (PDB).
 Valeur de retour : *objet JSON*.

Paramètres :

- **connString** - peut être :
  - l'URI - si aucune session portant le nom indiqué n'est trouvée et que `Plugins.Oracle.ResolveTNS` est défini sur false. S'il contient un schéma (par exemple, "tcp://"), un port (par exemple, 1521) ou les deux, l'option ResolveTNS ne sera pas prise en compte et il sera de toute façon traité comme un URI ;
  - le nom de session - si un tel nom est trouvé dans le fichier de configuration du plugin ;
  - la clé TNS - si l'option ResolveTNS est définie sur true et qu'aucune des conditions ci-dessus ne s'applique ;
  - la valeur TNS - si elle commence par une parenthèse ouvrante "(" (espaces initiaux ignorés). Dans ce cas, l'option ResolveTNS n'est pas prise en compte.
- **user** - le nom d'utilisateur Oracle ; prend en charge l'ajout de l'une des options de connexion `as sysdba`, `as sysoper`, `as sysasm`, `as sysbackup`, `as sysdg`, `as syskm` ou `as sysrac` au format `user as sysdba` (une option de connexion n'est pas sensible à la casse et ne doit pas contenir d'espace final).
- **password** - le mot de passe Oracle ;
- **service** - le nom du service Oracle ;
- **database** - le nom de la base de données à interroger.

oracle.pdb.discovery[connString,<user>,<password>,<service>]

 Renvoie la liste des PDB. Utilisé pour la découverte de bas niveau.
 Valeur de retour : *objet JSON*.

Paramètres :

- **connString** - peut être :
  - l'URI - si aucune session portant le nom donné n'est trouvée, et que `Plugins.Oracle.ResolveTNS` est défini sur false. S'il contient un schéma (par exemple, "tcp://"), un port (par exemple, 1521) ou les deux, l'option ResolveTNS ne sera pas prise en compte, et il sera de toute façon traité comme un URI ;
  - nom de session - si un tel nom est trouvé dans le fichier de configuration du plugin ;
  - clé TNS - si l'option ResolveTNS est définie sur true, et qu'aucune des conditions ci-dessus ne s'applique ;
  - valeur TNS - si elle commence par la parenthèse ouvrante "(" (espaces initiaux ignorés). Dans ce cas, l'option ResolveTNS n'est pas prise en compte.
- **user** - le nom d'utilisateur Oracle ; prend en charge l'ajout de l'une des options de connexion `as sysdba`, `as sysoper`, `as sysasm`, `as sysbackup`, `as sysdg`, `as syskm` ou `as sysrac` au format `user as sysdba` (une option de connexion n'est pas sensible à la casse et ne doit pas contenir d'espace final).
- **password** - le mot de passe Oracle ;
- **service** - le nom du service Oracle.

oracle.pga.stats[connString,<user>,<password>,<service>]

 Renvoie les statistiques de la Program Global Area (PGA).
 Valeur de retour : *objet JSON*.

Paramètres :

- **connString** - peut être :
  - l'URI - si aucune session portant le nom donné n'est trouvée et que `Plugins.Oracle.ResolveTNS` est défini sur false. S'il contient un schéma (par ex. "tcp://"), un port (par ex. 1521) ou les deux, l'option ResolveTNS ne sera pas prise en compte et il sera de toute façon traité comme un URI ;
  - le nom de session - si un tel nom est trouvé dans le fichier de configuration du plugin ;
  - la clé TNS - si l'option ResolveTNS est définie sur true et qu'aucune des conditions ci-dessus ne s'applique ;
  - la valeur TNS - si elle commence par la parenthèse ouvrante "(" (espaces initiaux ignorés). Dans ce cas, l'option ResolveTNS n'est pas prise en compte.
- **user** - le nom d'utilisateur Oracle ; prend en charge l'ajout de l'une des options de connexion `as sysdba`, `as sysoper`, `as sysasm`, `as sysbackup`, `as sysdg`, `as syskm` ou `as sysrac` au format `user as sysdba` (une option de connexion n'est pas sensible à la casse et ne doit pas contenir d'espace final).
- **password** - le mot de passe Oracle ;
- **service** - le nom du service Oracle.

oracle.ping[connString,<user>,<password>,<service>]

 Teste si une connexion à Oracle peut être établie.
 Valeur de retour : *1* - la connexion a réussi ; *0* - la connexion est interrompue (si une erreur est présente, y compris des problèmes d'AUTH et de configuration).

Paramètres :

- **connString** - peut être :

    - l'URI - si aucune session portant le nom donné n'est trouvée et que `Plugins.Oracle.ResolveTNS` est défini sur false. S'il contient un schéma (par ex., "tcp://"), un port (par ex., 1521) ou les deux, l'option ResolveTNS ne sera pas prise en compte et il sera de toute façon traité comme un URI ;
    - le nom de session - si un tel nom est trouvé dans le fichier de configuration du plugin ;
    - la clé TNS - si l'option ResolveTNS est définie sur true et qu'aucune des conditions ci-dessus ne s'applique ;
    - la valeur TNS - si elle commence par une parenthèse ouvrante "(" (espaces initiaux ignorés). Dans ce cas, l'option ResolveTNS n'est pas prise en compte.
- **user** - le nom d'utilisateur Oracle ; prend en charge l'ajout de l'une des options de connexion `as sysdba`, `as sysoper`, `as sysasm`, `as sysbackup`, `as sysdg`, `as syskm` ou `as sysrac` au format `user as sysdba` (une option de connexion est insensible à la casse et ne doit pas contenir d'espace final).
- **password** - le mot de passe Oracle ;
- **service** - le nom du service Oracle.

Exemples :

```
oracle.ping[tcp://127.0.0.1:1521,ZABBIX_MON,zabbix,xe]
oracle.ping[localhost,ZABBIX_MON,zabbix,xe]
oracle.ping[zbx_tns_example,ZABBIX_MON,zabbix,xe]
oracle.ping["(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=localhost)(PORT=1521))(CONNECT_DATA=(SERVICE_NAME=x
```

oracle.proc.stats[connString,<user>,<password>,<service>]

 Renvoie les statistiques des processus.
 Valeur de retour : *objet JSON*.

Paramètres :

- **connString** - peut être :
    - l'URI - si aucune session portant le nom indiqué n'est trouvée et que `Plugins.Oracle.ResolveTNS` est défini sur false. S'il contient un schéma (par ex. "tcp://"), un port (par ex. 1521) ou les deux, l'option ResolveTNS ne sera pas prise en compte et il sera de toute façon traité comme un URI ;
    - le nom de session - si un tel nom est trouvé dans le fichier de configuration du plugin ;
    - la clé TNS - si l'option ResolveTNS est définie sur true et qu'aucune des conditions ci-dessus ne s'applique ;
    - la valeur TNS - si elle commence par la parenthèse ouvrante "(" (espaces initiaux ignorés). Dans ce cas, l'option ResolveTNS n'est pas prise en compte.
- **user** - le nom d'utilisateur Oracle ; prend en charge l'ajout de l'une des options de connexion `as sysdba`, `as sysoper`, `as sysasm`, `as sysbackup`, `as sysdg`, `as syskm` ou `as sysrac` au format `user as sysdba` (une option de connexion n'est pas sensible à la casse et ne doit pas contenir d'espace final).
- **password** - le mot de passe Oracle ;
- **service** - le nom du service Oracle.

oracle.redolog.info[connString,<user>,<password>,<service>]

 Les informations du fichier journal provenant du fichier de contrôle.
 Valeur de retour : *objet JSON*.

Paramètres :

- **connString** - peut être :
    - l'URI - si aucune session portant le nom donné n'est trouvée et que `Plugins.Oracle.ResolveTNS` est défini sur false. S'il contient un schéma (par ex., "tcp://"), un port (par ex., 1521) ou les deux, l'option ResolveTNS ne sera pas prise en compte et il sera de toute façon traité comme un URI ;
    - le nom de session - si un tel nom est trouvé dans le fichier de configuration du plugin ;
    - la clé TNS - si l'option ResolveTNS est définie sur true et qu'aucune des conditions ci-dessus ne s'applique ;
    - la valeur TNS - si elle commence par la parenthèse ouvrante "(" (espaces initiaux ignorés). Dans ce cas, l'option ResolveTNS n'est pas prise en compte.
- **user** - le nom d'utilisateur Oracle ; prend en charge l'ajout de l'une des options de connexion `as sysdba`, `as sysoper`, `as sysasm`, `as sysbackup`, `as sysdg`, `as syskm` ou `as sysrac` au format `user as sysdba` (une option de connexion n'est pas sensible à la casse et ne doit pas contenir d'espace final).
- **password** - le mot de passe Oracle ;
- **service** - le nom du service Oracle.

oracle.sga.stats[connString,<user>,<password>,<service>]

 Renvoie les statistiques de la System Global Area (SGA).
 Valeur de retour : *objet JSON*.

Paramètres :

- **connString** - peut être :
    - l'URI - si aucune session portant le nom donné n'est trouvée et que `Plugins.Oracle.ResolveTNS` est défini sur false. S'il contient un schéma (par ex. "tcp://"), un port (par ex. 1521) ou les deux, l'option ResolveTNS ne sera pas prise en compte et il sera de toute façon traité comme un URI ;

    - le nom de session - si un tel nom est trouvé dans le fichier de configuration du plugin ;
    - la clé TNS - si l'option ResolveTNS est définie sur true et qu'aucune des conditions ci-dessus ne s'applique ;
    - la valeur TNS - si elle commence par la parenthèse ouvrante "(" (espaces initiaux ignorés). Dans ce cas, l'option ResolveTNS n'est pas prise en compte.
- **user** - le nom d'utilisateur Oracle ; prend en charge l'ajout de l'une des options de connexion `as sysdba`, `as sysoper`, `as sysasm`, `as sysbackup`, `as sysdg`, `as syskm` ou `as sysrac` au format `user as sysdba` (une option de connexion n'est pas sensible à la casse et ne doit pas contenir d'espace final).
- **password** - le mot de passe Oracle ;
- **service** - le nom du service Oracle.

oracle.sessions.stats[connString,<user>,<password>,<service>,<lockMaxTime>]

 Renvoie les statistiques des sessions.
 Valeur de retour : *objet JSON*.

Paramètres :

- **connString** - peut être :
    - l'URI - si aucune session portant le nom indiqué n'est trouvée et que `Plugins.Oracle.ResolveTNS` est défini sur false. S'il contient un schéma (par ex. "tcp://"), un port (par ex. 1521) ou les deux, l'option ResolveTNS ne sera pas prise en compte et il sera de toute façon traité comme un URI ;
    - nom de session - si un tel nom est trouvé dans le fichier de configuration du plugin ;
    - clé TNS - si l'option ResolveTNS est définie sur true et qu'aucune des conditions ci-dessus ne s'applique ;
    - valeur TNS - si elle commence par la parenthèse ouvrante "(" (espaces initiaux ignorés). Dans ce cas, l'option ResolveTNS n'est pas prise en compte.
- **user** - le nom d'utilisateur Oracle ; prend en charge l'ajout de l'une des options de connexion `as sysdba`, `as sysoper`, `as sysasm`, `as sysbackup`, `as sysdg`, `as syskm` ou `as sysrac` au format `user as sysdba` (une option de connexion est insensible à la casse et ne doit pas contenir d'espace final).
- **password** - le mot de passe Oracle ;
- **service** - le nom du service Oracle ;
- **lockMaxTime** - la durée maximale de verrouillage de session, en secondes, pour compter la session comme verrouillée de manière prolongée. Par défaut : 600 secondes.

oracle.sys.metrics[connString,<user>,<password>,<service>,<duration>]

 Renvoie un ensemble de valeurs de métriques système.
 Valeur renvoyée : *objet JSON*.

Paramètres :

- **connString** - peut être :
    - l'URI - si aucune session portant le nom indiqué n'est trouvée et que `Plugins.Oracle.ResolveTNS` est défini sur false. Si elle contient un schéma (par exemple, "tcp://"), un port (par exemple, 1521) ou les deux, l'option ResolveTNS ne sera pas prise en compte et elle sera tout de même traitée comme une URI ;
    - le nom de session - si ce nom est trouvé dans le fichier de configuration du plugin ;
    - la clé TNS - si l'option ResolveTNS est définie sur true et qu'aucune des conditions ci-dessus ne s'applique ;
    - la valeur TNS - si elle commence par la parenthèse ouvrante "(" (les espaces en début sont ignorés). Dans ce cas, l'option ResolveTNS n'est pas prise en compte.
- **user** - le nom d'utilisateur Oracle ; permet d'ajouter l'une des options de connexion `as sysdba`, `as sysoper`, `as sysasm`, `as sysbackup`, `as sysdg`, `as syskm` ou `as sysrac` au format `user as sysdba` (une option de connexion ne tient pas compte de la casse et ne doit pas contenir d'espace final).
- **password** - le mot de passe Oracle ;
- **service** - le nom du service Oracle ;
- **duration** - l'intervalle de collecte (en secondes) des valeurs de métriques système. Valeurs possibles : *60* - durée longue (par défaut), *15* - durée courte.

oracle.sys.params[connString,<user>,<password>,<service>]

 Renvoie un ensemble de valeurs de paramètres système.
 Valeur de retour : *objet JSON*.

Paramètres :

- **connString** - peut être :
    - l'URI - si aucune session portant le nom donné n'est trouvée et que `Plugins.Oracle.ResolveTNS` est défini sur false. S'il contient un schéma (par ex., "tcp://"), un port (par ex., 1521) ou les deux, l'option ResolveTNS ne sera pas prise en compte et il sera de toute façon traité comme un URI ;
    - le nom de session - si un tel nom est trouvé dans le fichier de configuration du plugin ;
    - la clé TNS - si l'option ResolveTNS est définie sur true et qu'aucune des conditions ci-dessus ne s'applique ;
    - la valeur TNS - si elle commence par la parenthèse ouvrante "(" (espaces initiaux ignorés). Dans ce cas, l'option ResolveTNS n'est pas prise en compte.

- **user** - le nom d'utilisateur Oracle ; prend en charge l'ajout de l'une des options de connexion `as sysdba`, `as sysoper`, `as sysasm`, `as sysbackup`, `as sysdg`, `as syskm` ou `as sysrac` au format `user as sysdba` (une option de connexion n'est pas sensible à la casse et ne doit pas contenir d'espace final).
- **password** - le mot de passe Oracle ;
- **service** - le nom du service Oracle.

oracle.ts.stats[connString,<user>,<password>,<service>,<tablespace>,<type>,<conname>]

 Renvoie les statistiques des tablespaces.
 Valeur de retour : *objet JSON*.

Paramètres :

- **connString** - peut être :
  - l'URI - si aucune session portant le nom donné n'est trouvée et que `Plugins.Oracle.ResolveTNS` est défini sur false. S'il contient un schéma (par ex. "tcp://"), un port (par ex. 1521) ou les deux, l'option ResolveTNS ne sera pas prise en compte et il sera de toute façon traité comme un URI ;
  - nom de session - si un tel nom est trouvé dans le fichier de configuration du plugin ;
  - clé TNS - si l'option ResolveTNS est définie sur true et qu'aucune des conditions ci-dessus ne s'applique ;
  - valeur TNS - si elle commence par la parenthèse ouvrante "(" (espaces initiaux ignorés). Dans ce cas, l'option ResolveTNS n'est pas prise en compte.
- **user** - le nom d'utilisateur Oracle ; prend en charge l'ajout de l'une des options de connexion `as sysdba`, `as sysoper`, `as sysasm`, `as sysbackup`, `as sysdg`, `as syskm` ou `as sysrac` au format `user as sysdba` (une option de connexion n'est pas sensible à la casse et ne doit pas contenir d'espace final).
- **password** - le mot de passe Oracle ;
- **service** - le nom du service Oracle ;
- **tablespace** - nom du tablespace à interroger. Par défaut (si laissé vide et que `type` est défini) :
  - "TEMP" (si `type` est défini sur "TEMPORARY") ;
  - "USERS" (si `type` est défini sur "PERMANENT").
- **type** - le type du tablespace à interroger. Par défaut (si `tablespace` est défini) : "PERMANENT".
- **conname** - nom du conteneur pour lequel les informations sont requises.

Si `tablespace`, `type` ou `conname` est omis, l'élément renverra les statistiques des tablespaces pour tous les conteneurs correspondants (y compris les PDB et le CDB).

oracle.ts.discovery[connString,<user>,<password>,<service>]

 Renvoie une liste de tablespaces. Utilisé pour la découverte de bas niveau.
 Valeur de retour : *objet JSON*.

Paramètres :

- **connString** - peut être :
  - l'URI - si aucune session portant le nom donné n'est trouvée et que `Plugins.Oracle.ResolveTNS` est défini sur false. S'il contient un schéma (par ex., "tcp://"), un port (par ex., 1521) ou les deux, l'option ResolveTNS ne sera pas prise en compte et il sera de toute façon traité comme un URI ;
  - nom de session - si un tel nom est trouvé dans le fichier de configuration du plugin ;
  - clé TNS - si l'option ResolveTNS est définie sur true et qu'aucune des conditions ci-dessus ne s'applique ;
  - valeur TNS - si elle commence par la parenthèse ouvrante "(" (espaces initiaux ignorés). Dans ce cas, l'option ResolveTNS n'est pas prise en compte.
- **user** - le nom d'utilisateur Oracle ; prend en charge l'ajout de l'une des options de connexion `as sysdba`, `as sysoper`, `as sysasm`, `as sysbackup`, `as sysdg`, `as syskm` ou `as sysrac` au format `user as sysdba` (une option de connexion n'est pas sensible à la casse et ne doit pas contenir d'espace final).
- **password** - le mot de passe Oracle ;
- **service** - le nom du service Oracle.

oracle.user.info[connString,<user>,<password>,<service>,<username>]

 Renvoie des informations sur l'utilisateur Oracle.
 Valeur de retour : *objet JSON*.

Paramètres :

- **connString** - peut être :
  - l'URI - si aucune session portant le nom indiqué n'est trouvée et que `Plugins.Oracle.ResolveTNS` est défini sur false. S'il contient un schéma (par ex., "tcp://"), un port (par ex., 1521) ou les deux, l'option ResolveTNS ne sera pas prise en compte et il sera de toute façon traité comme un URI ;
  - le nom de session - si un tel nom est trouvé dans le fichier de configuration du plugin ;
  - la clé TNS - si l'option ResolveTNS est définie sur true et qu'aucune des conditions ci-dessus ne s'applique ;
  - la valeur TNS - si elle commence par une parenthèse ouvrante "(" (espaces initiaux ignorés). Dans ce cas, l'option ResolveTNS n'est pas prise en compte.

- **user** - le nom d'utilisateur Oracle ; prend en charge l'ajout de l'une des options de connexion `as sysdba`, `as sysoper`, `as sysasm`, `as sysbackup`, `as sysdg`, `as syskm` ou `as sysrac` au format `user as sysdba` (une option de connexion n'est pas sensible à la casse et ne doit pas contenir d'espace final).
- **password** - le mot de passe Oracle ;
- **service** - le nom du service Oracle ;
- **username** - le nom d'utilisateur pour lequel les informations sont requises. Les noms d'utilisateur en minuscules ne sont pas pris en charge. Par défaut : utilisateur actuel.

oracle.version[connString,<user>,<password>,<service>]

 Renvoie la version du serveur de base de données.
 Valeur de retour : *String*.

Paramètres :

- **connString** - peut être :
  - l'URI - si aucune session portant le nom donné n'est trouvée et que `Plugins.Oracle.ResolveTNS` est défini sur false. S'il contient un schéma (par ex. "tcp://"), un port (par ex. 1521) ou les deux, l'option ResolveTNS ne sera pas prise en compte et il sera de toute façon traité comme un URI ;
  - le nom de session - si un tel nom est trouvé dans le fichier de configuration du plugin ;
  - la clé TNS - si l'option ResolveTNS est définie sur true et qu'aucune des conditions ci-dessus ne s'applique ;
  - la valeur TNS - si elle commence par la parenthèse ouvrante "(" (espaces initiaux ignorés). Dans ce cas, l'option ResolveTNS n'est pas prise en compte.
- **user** - le nom d'utilisateur Oracle ; prend en charge l'ajout de l'une des options de connexion `as sysdba`, `as sysoper`, `as sysasm`, `as sysbackup`, `as sysdg`, `as syskm` ou `as sysrac` au format `user as sysdba` (une option de connexion n'est pas sensible à la casse et ne doit pas contenir d'espace final).
- **password** - le mot de passe Oracle ;
- **service** - le nom du service Oracle.

pgsql.autovacuum.count[uri,<username>,<password>,<database name>]

 Le nombre de workers autovacuum.
 Valeur de retour: *Integer*.

Paramètres:

- **uri** - l'URI ou le nom de session;
- **username, password** - les identifiants PostgreSQL;
- **database name** - le nom de la base de données (si omis, la valeur de `Plugins.PostgreSQL.Default.Database` de `postgresql.conf` est utilisée).

pgsql.archive[uri,<username>,<password>,<database name>]

 Informations sur les fichiers archivés.
 Valeur de retour : *objet JSON*.

Paramètres :

- **uri** - l'URI ou le nom de session ;
- **username, password** - les identifiants PostgreSQL ;
- **database name** - le nom de la base de données (si omis, la valeur de `Plugins.PostgreSQL.Default.Database` dans `postgresql.conf` est utilisée).

pgsql.bgwriter[uri,<username>,<password>,<database name>]

 Le nombre total de points de contrôle pour le cluster de bases de données, ventilé par type de point de contrôle.
 Valeur de retour: *objet JSON*.

Paramètres:

- **uri** - l'URI ou le nom de session;
- **username, password** - les identifiants PostgreSQL;
- **database name** - le nom de la base de données (si omis, la valeur de `Plugins.PostgreSQL.Default.Database` dans `postgresql.conf` est utilisée).

pgsql.cache.hit[uri,<username>,<password>,<database name>]

 Le taux de réussite du cache tampon PostgreSQL.
 Valeur de retour : *Float*.

Paramètres :

- **uri** - l'URI ou le nom de session ;
- **username, password** - les identifiants PostgreSQL ;
- **database name** - le nom de la base de données (si omis, la valeur de `Plugins.PostgreSQL.Default.Database` définie dans `postgresql.conf` est utilisée).

pgsql.connections[uri,<username>,<password>,<database name>]

 Renvoie les connexions par type.
 Valeur de retour: *objet JSON*.

Paramètres:

- **uri** - l'URI ou le nom de session;
- **username, password** - les identifiants PostgreSQL;
- **database name** - le nom de la base de données (si omis, la valeur de `Plugins.PostgreSQL.Default.Database` de `postgresql.conf` est utilisée).

pgsql.custom.query[uri,<username>,<password>,<database name>,queryName,<args...>]

 Renvoie le résultat d'une requête personnalisée.
 Valeur de retour: *objet JSON*.

Paramètres:

- **uri** - l'URI ou le nom de session;
- **username, password** - les identifiants PostgreSQL;
- **database name** - le nom de la base de données (si omis, la valeur de Plugins.PostgreSQL.Default.Database de postgresql.conf est utilisée);
- **queryName** - le nom d'une requête personnalisée, doit correspondre au nom du fichier SQL sans extension;
- **args** - un ou plusieurs arguments séparés par des virgules à transmettre à une requête.

pgsql.db.age[uri,<username>,<password>,<database name>]

 L'âge du plus ancien FrozenXID de la base de données (nombre de transactions).
 Valeur de retour: *Integer*.

Paramètres:

- **uri** - l'URI ou le nom de session;
- **username, password** - les identifiants PostgreSQL;
- **database name** - le nom de la base de données (si omis, la valeur de `Plugins.PostgreSQL.Default.Database` de `postgresql.conf` est utilisée).

pgsql.db.bloating_tables[uri,<username>,<password>,<database name>]

 Le nombre de tables gonflées par base de données.
 Valeur de retour : *Integer*.

Paramètres :

- **uri** - l'URI ou le nom de session ;
- **username, password** - les identifiants PostgreSQL ;
- **database name** - le nom de la base de données (si omis, la valeur de `Plugins.PostgreSQL.Default.Database` de `postgresql.conf` est utilisée).

pgsql.db.discovery[uri,<username>,<password>,<database name>]

 La liste des bases de données PostgreSQL. Utilisé pour la découverte de bas niveau.
 Valeur de retour: *objet JSON*.

Paramètres:

- **uri** - l'URI ou le nom de session;
- **username, password** - les identifiants PostgreSQL;
- **database name** - le nom de la base de données (si omis, la valeur de `Plugins.PostgreSQL.Default.Database` dans `postgresql.conf` est utilisée).

pgsql.db.size[uri,<username>,<password>,<database name>]

 La taille de la base de données en octets.
 Valeur de retour: *Integer*.

Paramètres:

- **uri** - l'URI ou le nom de session;
- **username, password** - les identifiants PostgreSQL;
- **database name** - le nom de la base de données (si omis, la valeur de `Plugins.PostgreSQL.Default.Database` de `postgresql.conf` est utilisée).

pgsql.dbstat[uri,<username>,<password>,<database name>]

 Collecte les statistiques par base de données. Utilisé pour la découverte de bas niveau.
 Valeur de retour : *objet JSON*.

Paramètres :

- **uri** - l'URI ou le nom de session ;
- **username, password** - les identifiants PostgreSQL ;

- **database name** - le nom de la base de données (si omis, la valeur de `Plugins.PostgreSQL.Default.Database` de `postgresql.conf` est utilisée).

pgsql.dbstat.sum[uri,<username>,<password>,<database name>]

 Les données agrégées pour toutes les bases de données d'un cluster.
 Valeur de retour : *objet JSON*.

Paramètres :

- **uri** - l'URI ou le nom de session ;
- **username, password** - les identifiants PostgreSQL ;
- **database name** - le nom de la base de données (s'il est omis, la valeur de `Plugins.PostgreSQL.Default.Database` de `postgresql.conf` est utilisée).

pgsql.locks[uri,<username>,<password>,<database name>]

 Les informations sur les verrous accordés par base de données. Utilisé pour la découverte de bas niveau.
 Valeur de retour: *objet JSON*.

Paramètres:

- **uri** - l'URI ou le nom de session;
- **username, password** - les identifiants PostgreSQL;
- **database name** - le nom de la base de données (si omis, la valeur de `Plugins.PostgreSQL.Default.Database` de `postgresql.conf` est utilisée).

pgsql.oldest.xid[uri,<username>,<password>,<database name>]

 L'âge du plus ancien XID (nombre de transactions).
 Valeur de retour: *Integer*.

Paramètres:

- **uri** - l'URI ou le nom de session;
- **username, password** - les identifiants PostgreSQL;
- **database name** - le nom de la base de données (s'il est omis, la valeur de `Plugins.PostgreSQL.Default.Database` de `postgresql.conf` est utilisée).

pgsql.ping[uri,<username>,<password>,<database name>]

 Teste si une connexion est active ou non.
 Valeur de retour: *1* - la connexion est active; *0* - la connexion est rompue (en cas d'erreur, y compris les problèmes d'authentification et de configuration).

Paramètres:

- **uri** - l'URI ou le nom de session;
- **username, password** - les identifiants PostgreSQL;
- **database name** - le nom de la base de données (si omis, la valeur de `Plugins.PostgreSQL.Default.Database` dans `postgresql.conf` est utilisée).

pgsql.queries[uri,<username>,<password>,<database name>,time period]

 Interroge les métriques des requêtes selon le temps d'exécution.
 Valeur de retour: *objet JSON*.

Paramètres:

- **uri** - l'URI ou le nom de session;
- **username, password** - les identifiants PostgreSQL;
- **database name** - le nom de la base de données (si omis, la valeur de `Plugins.PostgreSQL.Default.Database` de `postgresql.conf` est utilisée);
- **time period** - la limite de temps d'exécution pour le nombre de requêtes lentes (doit être un entier positif).

pgsql.replication.count[uri,<username>,<password>]

 Le nombre de serveurs de secours.
 Valeur de retour : *Integer*.

Paramètres :

- **uri** - l'URI ou le nom de session ;
- **username, password** - les identifiants PostgreSQL.

pgsql.replication.process[uri,<username>,<password>]

 Le délai de flush, le délai d'écriture et le délai de relecture pour chaque processus expéditeur.
 Valeur de retour : *objet JSON*.

Paramètres :

- **uri** - l'URI ou le nom de session ;
- **username, password** - les identifiants PostgreSQL.

pgsql.replication.process.discovery[uri,<username>,<password>]

 La découverte du nom du processus de réplication.
 Valeur de retour : *objet JSON*.

Paramètres :

- **uri** - l'URI ou le nom de session ;
- **username, password** - les identifiants PostgreSQL.

pgsql.replication.recovery_role[uri,<username>,<password>]

 Le statut de récupération.
 Valeur de retour : *0* - mode maître ; *1* - la récupération est toujours en cours (mode veille).

Paramètres :

- **uri** - l'URI ou le nom de session ;
- **username, password** - les identifiants PostgreSQL.

pgsql.replication.status[uri,<username>,<password>]

 Le statut de la réplication.
 Valeur de retour : *0* - le streaming est arrêté ; *1* - le streaming est actif ; *2* - mode maître.

Paramètres :

- **uri** - l'URI ou le nom de session ;
- **username, password** - les identifiants PostgreSQL.

pgsql.replication_lag.b[uri,<username>,<password>]

 Le retard de réplication en octets.
 Valeur de retour : *Integer*.

Paramètres :

- **uri** - l'URI ou le nom de session ;
- **username, password** - les identifiants PostgreSQL.

pgsql.replication_lag.sec[uri,<username>,<password>]

 Le retard de réplication en secondes.
 Valeur de retour : *Integer*.

Paramètres :

- **uri** - l'URI ou le nom de session ;
- **username, password** - les identifiants PostgreSQL.

pgsql.uptime[uri,<username>,<password>,<database name>]

 Le temps de fonctionnement de PostgreSQL en millisecondes.
 Valeur de retour: *Float*.

Paramètres:

- **uri** - l'URI ou le nom de session;
- **username, password** - les identifiants PostgreSQL;
- **database name** - le nom de la base de données (si omis, la valeur de `Plugins.PostgreSQL.Default.Database` de `postgresql.conf` est utilisée).

pgsql.version[uri,<username>,<password>,<database name>]

 Renvoie la version de PostgreSQL.
 Valeur de retour: *String*.

Paramètres:

- **uri** - l'URI ou le nom de session;
- **username, password** - les identifiants PostgreSQL;
- **database name** - le nom de la base de données (si omis, la valeur de `Plugins.PostgreSQL.Default.Database` de `postgresql.conf` est utilisée).

pgsql.wal.stat[uri,<username>,<password>,<database name>]

 Les statistiques WAL.
 Valeur de retour: *objet JSON*.

Paramètres:

- **uri** - l'URI ou le nom de session;
- **username, password** - les identifiants PostgreSQL;

- **database name** - le nom de la base de données (s'il est omis, la valeur de `Plugins.PostgreSQL.Default.Database` de `postgresql.conf` est utilisée).

redis.config[connString,<password>,<pattern>]

 Obtient les paramètres de configuration d'une instance Redis qui correspondent au motif.
 Valeur de retour : *JSON* - si un motif de type glob a été utilisé ; valeur unique - si le motif ne contenait aucun caractère générique.

Paramètres :

- **connString** - l'URI ou le nom de session ;
- **password** - le mot de passe Redis ;
- **pattern** - un motif de type glob (* par défaut).

Commentaire :

- Avertissement de sécurité : `redis.config` exécute la commande Redis `CONFIG GET` et peut renvoyer des paramètres de configuration sensibles (par exemple `requirepass`, `masterauth`, les paramètres liés à TLS et d'autres secrets). L'utilisation du motif par défaut (*) ou de motifs larges peut exposer ces secrets à toute personne autorisée à consulter les valeurs des éléments dans Zabbix.

redis.info[connString,<password>,<section>]

 Obtient la sortie de la commande INFO.
 Valeur de retour : *JSON* - la sortie est sérialisée au format JSON.

Paramètres :

- **connString** - l'URI ou le nom de session ;
- **password** - le mot de passe Redis ;
- **section** - la section d'informations (*default* par défaut).

redis.ping[connString,<password>]

 Teste si une connexion est active ou non.
 Valeur de retour : *1* - la connexion est active ; *0* - la connexion est interrompue (si une erreur est présente, y compris des problèmes d'AUTH et de configuration).

Paramètres :

- **connString** - l'URI ou le nom de session ;
- **password** - le mot de passe Redis.

redis.slowlog.count[connString,<password>]

 Le nombre d'entrées du journal lent depuis le démarrage de Redis.
 Valeur de retour : *Integer*.

Paramètres :

- **connString** - l'URI ou le nom de session ;
- **password** - le mot de passe Redis.

smart.attribute.discovery

 Renvoie une liste des attributs S.M.A.R.T. du périphérique.
 Valeur de retour: *objet JSON*.

Commentaires:

- Les macros suivantes et leurs valeurs sont renvoyées: `{#NAME}`, `{#DISKTYPE}`, `{#ID}`, `{#ATTRNAME}`, `{#THRESH}`;
- Les types de disques HDD, SSD et NVME sont pris en charge. Les disques peuvent être seuls ou combinés dans un RAID. `{#NAME}` comportera un suffixe en cas de RAID, par exemple: `{"{#NAME}": "/dev/sda cciss,2"}`.

smart.disk.discovery[<type>]

 Renvoie une liste de périphériques S.M.A.R.T.
 Valeur de retour: *objet JSON*.

Paramètre:

- **type** - spécifie une valeur à rechercher pour les disques. Valeurs possibles: *id*, *name* (par défaut). *id* n'est pas pris en charge sous Windows et renverra une erreur s'il est sélectionné

Commentaires:

- Les macros suivantes et leurs valeurs sont renvoyées: `{#NAME}`, `{#DISKTYPE}`, `{#MODEL}`, `{#SN}`, `{#PATH}`, `{#ATTRIBUTES}`, `{#RAIDTYPE}`;
- Les types de lecteurs HDD, SSD et NVME sont pris en charge. Si un lecteur n'appartient pas à un RAID, `{#RAIDTYPE}` sera vide. `{#NAME}` comportera un ajout en cas de RAID, par exemple: `{"{#NAME}": "/dev/sda cciss,2"}`.

smart.disk.get[<path>,<raid type>]

 Renvoie toutes les propriétés disponibles des périphériques S.M.A.R.T.
 Valeur de retour: *objet JSON*.

Paramètres:

- **path** - le chemin du disque, la macro `{#PATH}` peut être utilisée comme valeur;
- **raid_type** - le type de RAID, la macro `{#RAID}` peut être utilisée comme valeur

Commentaires:

- Les types de disques HDD, SSD et NVME sont pris en charge. Les disques peuvent être seuls ou combinés dans un RAID;
- Les données incluent la version de smartctl et les arguments d'appel, ainsi que des champs supplémentaires:
disk_name - contient le nom avec les ajouts requis pour la découverte RAID, par exemple: `{"disk_name": "/dev/sda cciss,2"}`
disk_type - contient le type de disque HDD, SSD ou NVME, par exemple: `{"disk_type": "ssd"}`;
- Si aucun paramètre n'est spécifié, l'élément renverra des informations sur tous les disques.

systemd.unit.get[unit name,<interface>]

 Renvoie toutes les propriétés d'une unité systemd.
 Valeur de retour: *objet JSON*.

Paramètres:

- **unit name** - le nom de l'unité (vous pouvez utiliser la macro `{#UNIT.NAME}` dans le prototype d'élément pour découvrir le nom);
- **interface** - le type d'interface de l'unité, valeurs possibles: *Unit* (par défaut), *Service*, *Socket*, *Device*, *Mount*, *Automount*, *Swap*, *Target*, *Path*.

Commentaires:

- Cet élément est pris en charge uniquement sur la plateforme Linux;
- LoadState, ActiveState et UnitFileState pour l'interface Unit sont renvoyés sous forme de texte et d'entier: `"ActiveState":{"state"`

systemd.unit.info[unit name,<property>,<interface>]

 Informations sur une unité systemd.
 Valeur retournée : *String*.

Paramètres :

- **unit name** - nom de l'unité (vous pouvez utiliser la macro `{#UNIT.NAME}` dans le prototype d'élément pour découvrir le nom) ;
- **property** - propriété de l'unité (par exemple, ActiveState (par défaut), LoadState, Description) ;
- **interface** - type d'interface de l'unité (par exemple, Unit (par défaut), Socket, Service).

Commentaires :

- Cet élément est pris en charge uniquement sur la plateforme Linux ;
- Cet élément permet de récupérer une propriété spécifique d'un type d'interface spécifique, comme décrit dans l'API dbus.

Exemples :

```
systemd.unit.info["{#UNIT.NAME}"] #collecter les informations sur l'état actif (active, reloading, inactiv
systemd.unit.info["{#UNIT.NAME}",LoadState] #collecter les informations sur l'état du chargement des unité
systemd.unit.info[mysqld.service,Id] #récupérer le nom technique du service (mysqld.service)
systemd.unit.info[mysqld.service,Description] #récupérer la description du service (MySQL Server)
systemd.unit.info[mysqld.service,ActiveEnterTimestamp] #récupérer la dernière date à laquelle le service e
systemd.unit.info[dbus.socket,NConnections,Socket] #collecter le nombre de connexions de cette unité socke
```

systemd.unit.discovery[<type>]

 Liste des unités systemd et de leurs détails. Utilisé pour la découverte de bas niveau.
 Valeur de retour : *objet JSON*.

Paramètre :

- **type** - valeurs possibles : *all*, *automount*, *device*, *mount*, *path*, *service* (par défaut), *socket*, *swap*, *target*.

Cet élément est pris en charge uniquement sur la plateforme Linux.

web.certificate.get[hostname,<port>,<address>]

 Valide les certificats et renvoie les détails du certificat.
 Valeur de retour: *objet JSON*.

Paramètres:

- **hostname** - peut être une adresse IP ou DNS.
Peut contenir le schéma d'URL (*https* uniquement), le chemin (il sera ignoré) et le port.
Si un port est fourni à la fois dans le premier et le deuxième paramètre, leurs valeurs doivent correspondre.
Si address (le 3e paramètre) est spécifié, le hostname est utilisé uniquement pour SNI et la vérification du nom d'hôte;
- **port** - le numéro de port (la valeur par défaut est 443 pour HTTPS);
- **address** - peut être une adresse IP ou DNS. Si elle est spécifiée, elle sera utilisée pour la connexion, et hostname (le 1er paramètre) sera utilisé pour SNI et la vérification de l'hôte. Dans le cas où le 1er paramètre est une IP et le 3e paramètre est un DNS, le 1er paramètre sera utilisé pour la connexion, et le 3e paramètre sera utilisé pour SNI et la vérification de l'hôte.

Commentaires:

- Cet élément devient non pris en charge si la destination spécifiée dans la configuration de l'hôte n'existe pas, est indisponible, ou si la négociation TLS échoue avec toute erreur autre qu'un certificat invalide;
- Actuellement, l'extension X.509 AIA (Authority Information Access), les CRL et l'OCSP (y compris l'OCSP stapling), ainsi que Certificate Transparency ne sont pas pris en charge;
- Champs de la réponse JSON:
  - *x509*: contient les détails du certificat X.509.
    - *version*: la version X.509 (par ex. 3).
    - *serial_number*: le numéro de série du certificat.
    - *signature_algorithm*: l'algorithme utilisé pour signer le certificat (par ex. `SHA256-RSA`).
    - *issuer*: l'émetteur du certificat.
    - *not_before*: la date de début de validité du certificat.
    - *not_after*: la date d'expiration du certificat.
    - *subject*: le sujet du certificat.
    - *public_key_algorithm*: l'algorithme utilisé pour la clé publique (par ex. `RSA`).
    - *alternative_names*: noms alternatifs du sujet (s'ils sont présents), sinon `null`.
  - result: contient le résultat de la validation.
    - *value*: l'état de validation (voir les valeurs possibles ci-dessous).
    - *message*: message détaillé de validation (par ex. `"certificate verified successfully"`).
  - *sha1_fingerprint*: l'empreinte SHA-1 du certificat.
  - *sha256_fingerprint*: l'empreinte SHA-256 du certificat.
- Le champ `$.result.value` indique le résultat de la validation du certificat. Les valeurs possibles incluent:
  - *valid* - le certificat est valide et approuvé.
  - *valid-but-self-signed* - le certificat est valide mais auto-signé, ce qui signifie que son sujet correspond à son émetteur.
  - *invalid* - le certificat est invalide en raison d'un problème tel qu'une expiration, un nom d'hôte incorrect ou une autorité de signature inconnue.

Exemple:

```
web.certificate.get[example.com,443]
```

Réponse JSON:


```
{
  "x509": {
    "version": 3,
    "serial_number": "0ad893bafa68b0b7fb7a404f06ecaf9a",
    "signature_algorithm": "ECDSA-SHA384",
    "issuer": "CN=DigiCert Global G3 TLS ECC SHA384 2020 CA1,O=DigiCert Inc,C=US",
    "not_before": {
      "value": "Jan 15 00:00:00 2025 GMT",
      "timestamp": 1736899200
    },
    "not_after": {
      "value": "Jan 15 23:59:59 2026 GMT",
      "timestamp": 1768521599
    },
    "subject": "CN=*.example.com,O=Internet Corporation for Assigned Names and Numbers,L=Los Angeles,ST=Ca
    "public_key_algorithm": "ECDSA",
    "alternative_names": [
      "*.example.com",
      "example.com"
    ]
  },
  "result": {
    "value": "valid",
```

```
    "message": "certificate verified successfully"
  },
  "sha1_fingerprint": "310db7af4b2bc9040c8344701aca08d0c69381e3",
  "sha256_fingerprint": "455943cf819425761d1f950263ebf54755d8d684c25535943976f488bc79d23b"
}
```


Aperçu

Les éléments de l'agent Zabbix Windows sont présentés dans deux listes :

- Éléments partagés - les clés d'élément qui sont partagées avec l'agent Zabbix UNIX ;
- Éléments spécifiques à Windows - les clés d'élément qui sont prises en charge **uniquement** sous Windows.

Notez que toutes les clés d'élément prises en charge par l'agent Zabbix sous Windows sont également prises en charge par l'agent Zabbix 2 de nouvelle génération. Consultez les clés d'élément supplémentaires que vous pouvez utiliser uniquement avec l'agent 2.

Voir aussi : Autorisations minimales pour les éléments Windows

Éléments partagés

Le tableau ci-dessous répertorie les éléments de Zabbix agent pris en charge sous Windows et partagés avec le Zabbix agent UNIX :

- La clé de l'élément est un lien vers les détails complets de l'élément Zabbix agent UNIX
- Les commentaires d'élément pertinents pour Windows sont inclus

Item key	Description	Item group
log	La surveillance d'un fichier journal. Cet élément n'est pas pris en charge pour le journal des événements Windows. Le paramètre `persistent_dir` n'est pas pris en charge sous Windows.	Surveillance des journaux
log.count	Le nombre de lignes correspondantes dans un fichier journal surveillé. Cet élément n'est pas pris en charge pour le journal des événements Windows. Le paramètre `persistent_dir` n'est pas pris en charge sous Windows.	
logrt	La surveillance d'un fichier journal rotatif. Cet élément n'est pas pris en charge pour le journal des événements Windows. Le paramètre `persistent_dir` n'est pas pris en charge sous Windows.	
logrt.count	Le nombre de lignes correspondantes dans un fichier journal rotatif surveillé. Cet élément n'est pas pris en charge pour le journal des événements Windows. Le paramètre `persistent_dir` n'est pas pris en charge sous Windows.	
modbus.get	Lit des données Modbus.	Modbus
net.dns	Vérifie si le service DNS est actif. Les paramètres `ip`, `timeout` et `count` sont ignorés sous Windows, sauf si vous utilisez Zabbix agent 2.	Réseau
net.dns.perf	Vérifie les performances d'un service DNS. Les paramètres `ip`, `timeout` et `count` sont ignorés sous Windows, sauf si vous utilisez Zabbix agent 2.	
net.dns.record	Effectue une requête DNS. Les paramètres `ip`, `timeout` et `count` sont ignorés sous Windows, sauf si vous utilisez Zabbix agent 2.	
net.if.discovery	La liste des interfaces réseau. Certaines versions de Windows (par exemple Server 2008) peuvent nécessiter l'installation des dernières mises à jour pour prendre en charge les caractères non ASCII dans les noms d'interface.	
net.if.in	Les statistiques du trafic entrant sur une interface réseau. Sous Windows, l'élément obtient les valeurs à partir de compteurs 64 bits s'ils sont disponibles. Les compteurs de statistiques d'interface 64 bits ont été introduits dans Windows Vista et Windows Server 2008. Si les compteurs 64 bits ne sont pas disponibles, l'agent utilise des compteurs 32 bits. Les noms d'interface multioctets sont pris en charge sous Windows. Vous pouvez obtenir les descriptions des interfaces réseau sous Windows avec les éléments net.if.discovery ou net.if.list.	

Item key	Description	Item group
net.if.out	Les statistiques du trafic sortant sur une interface réseau. Sous Windows, l'élément obtient les valeurs à partir de compteurs 64 bits s'ils sont disponibles. Les compteurs de statistiques d'interface 64 bits ont été introduits dans Windows Vista et Windows Server 2008. Si les compteurs 64 bits ne sont pas disponibles, l'agent utilise des compteurs 32 bits. Les noms d'interface multioctets sont pris en charge sous Windows. Vous pouvez obtenir les descriptions des interfaces réseau sous Windows avec les éléments net.if.discovery ou net.if.list.	
net.if.total	La somme des statistiques du trafic entrant et sortant sur une interface réseau. Sous Windows, l'élément obtient les valeurs à partir de compteurs 64 bits s'ils sont disponibles. Les compteurs de statistiques d'interface 64 bits ont été introduits dans Windows Vista et Windows Server 2008. Si les compteurs 64 bits ne sont pas disponibles, l'agent utilise des compteurs 32 bits. Les noms d'interface multioctets sont pris en charge sous Windows. Vous pouvez obtenir les descriptions des interfaces réseau sous Windows avec les éléments net.if.discovery ou net.if.list.	
net.tcp.listen	Vérifie si ce port TCP est à l'état LISTEN.	
net.tcp.port	Vérifie s'il est possible d'établir une connexion TCP vers le port spécifié.	
net.tcp.service	Vérifie si un service est en cours d'exécution et accepte des connexions TCP. La vérification de LDAP et HTTPS sous Windows n'est prise en charge que par Zabbix agent 2.	
net.tcp.service.perf	Vérifie les performances d'un service TCP. La vérification de LDAP et HTTPS sous Windows n'est prise en charge que par Zabbix agent 2.	
net.tcp.socket.count	Renvoie le nombre de sockets TCP correspondant aux paramètres. Cet élément est pris en charge sous Linux par Zabbix agent, mais sous Windows il n'est pris en charge que par Zabbix agent 2 sur Windows 64 bits.	
net.udp.service	Vérifie si un service est en cours d'exécution et répond aux requêtes UDP.	
net.udp.service.perf	Vérifie les performances d'un service UDP.	
net.udp.socket.count	Renvoie le nombre de sockets UDP correspondant aux paramètres. Cet élément est pris en charge sous Linux par Zabbix agent, mais sous Windows il n'est pris en charge que par Zabbix agent 2 sur Windows 64 bits.	
proc.get	La liste des processus du système d'exploitation et de leurs paramètres. Le paramètre `cmdline` n'est pas pris en charge sous Windows.	Processus
proc.num	Le nombre de processus. Sous Windows, seuls les paramètres `name` et `user` sont pris en charge.	
system.cpu.discovery	La liste des CPU/cœurs CPU détectés.	Système
system.cpu.load	La charge CPU. Lorsqu'un processus de collecte est démarré sur Zabbix agent, les compteurs de performance suivants sont initialisés puis utilisés pour cet élément : `\System\Processor Queue Length`	
system.cpu.num	Le nombre de CPU.	
system.cpu.util	Le pourcentage d'utilisation du CPU. La valeur est obtenue à l'aide du compteur de performance *Processor Time*. Notez que depuis Windows 8, le Gestionnaire des tâches affiche l'utilisation du CPU sur la base du compteur de performance *Processor Utility*, alors que dans les versions précédentes il s'agissait du compteur *Processor Time* (voir plus de détails). *system* est le seul paramètre `type` pris en charge sous Windows.	
system.hostname	Le nom d'hôte du système. La valeur est obtenue via les fonctions `GetComputerName()` (pour **netbios**), `GetComputerNameExA()` (pour **fqdn**) ou `gethostname()` (pour **host**) sous Windows. Voir aussi une description plus détaillée.	
system.localtime	L'heure système.	
system.run	Exécute la commande spécifiée sur l'hôte.	
system.sw.arch	Les informations sur l'architecture logicielle.	
system.sw.os	Les informations sur le système d'exploitation.	
system.sw.os.get	Informations détaillées sur le système d'exploitation (version, type, nom de la distribution, version mineure et majeure, etc.).	

Item key	Description	Item group
system.swap.size	La taille de l'espace d'échange en octets ou en pourcentage du total. Le paramètre de type pused est pris en charge sous Linux par Zabbix agent, mais sous Windows il n'est pris en charge que par Zabbix agent 2. Notez que cette clé peut signaler une taille/un pourcentage d'espace d'échange incorrect sur les plateformes Windows virtualisées (VMware ESXi, VirtualBox). Dans ce cas, vous pouvez utiliser la clé `perf_counter[\700(_Total)\702]` pour obtenir le pourcentage d'espace d'échange correct.	
system.uname	Identification du système. Sous Windows, la valeur de cet élément est obtenue à partir des classes WMI `Win32_OperatingSystem` et `Win32_Processor`. Le nom du système d'exploitation (y compris l'édition) peut être traduit dans la langue d'affichage de l'utilisateur. Sur certaines versions de Windows, il contient des symboles de marque déposée et des espaces supplémentaires.	
system.uptime	Le temps de fonctionnement du système en secondes.	
vfs.dir.count	Le nombre d'entrées de répertoire. Sous Windows, les liens symboliques de répertoire sont ignorés et les liens physiques ne sont comptés qu'une seule fois.	Systèmes de fichiers virtuels
vfs.dir.get	La liste des entrées de répertoire. Sous Windows, les liens symboliques de répertoire sont ignorés et les liens physiques ne sont comptés qu'une seule fois.	
vfs.dir.size	La taille du répertoire. Sous Windows, tout lien symbolique est ignoré et les liens physiques ne sont pris en compte qu'une seule fois.	
vfs.file.cksum	La somme de contrôle du fichier, calculée par l'algorithme UNIX cksum.	
vfs.file.contents	Récupération du contenu d'un fichier.	
vfs.file.exists	Vérifie si le fichier existe. Sous Windows, les guillemets doubles doivent être échappés avec une barre oblique inverse '\' et toute la clé de l'élément doit être placée entre guillemets doubles lors de l'utilisation de l'utilitaire en ligne de commande pour appeler `zabbix_get.exe` ou `agent2`. Notez que l'élément peut devenir non pris en charge sous Windows si un répertoire est recherché dans un répertoire inexistant, par exemple `vfs.file.exists[C:\no\dir,dir]` (où 'no' n'existe pas).	
vfs.file.get	Renvoie des informations sur un fichier. Types de fichiers pris en charge sous Windows : fichier ordinaire, répertoire, lien symbolique	
vfs.file.md5sum	La somme de contrôle MD5 du fichier.	
vfs.file.owner	Récupère le propriétaire d'un fichier.	
vfs.file.regexp	Récupère une chaîne dans le fichier.	
vfs.file.regmatch	Recherche une chaîne dans le fichier.	
vfs.file.size	La taille du fichier.	
vfs.file.time	Les informations temporelles du fichier. Sous Windows XP, `vfs.file.time[file,change]` peut être égal à `vfs.file.time[file,access]`.	
vfs.fs.discovery	La liste des systèmes de fichiers montés avec leur type et leurs options de montage. La macro `{#FSLABEL}` est prise en charge sous Windows.	
vfs.fs.get	La liste des systèmes de fichiers montés avec leur type, l'espace disque disponible, les statistiques d'inodes et les options de montage. La macro {#FSLABEL} est prise en charge sous Windows.	
vfs.fs.size	L'espace disque en octets ou en pourcentage du total.	
vm.memory.size	La taille de la mémoire en octets ou en pourcentage du total.	Mémoire virtuelle
web.page.get	Récupère le contenu d'une page web.	Surveillance Web
web.page.perf	Le temps de chargement d'une page web complète.	
web.page.regexp	Recherche une chaîne sur la page web.	
agent.hostmetadata	Les métadonnées d'hôte de l'agent.	Zabbix
agent.hostname	Le nom d'hôte de l'agent.	
agent.ping	La vérification de disponibilité de l'agent.	
agent.variant	La variante de Zabbix agent (Zabbix agent ou Zabbix agent 2).	

Item key	Description	Item group
agent.version	La version de Zabbix agent.	
zabbix.stats	Renvoie à distance un ensemble de métriques internes du serveur ou du proxy Zabbix.	
zabbix.stats (queue)	Renvoie à distance le nombre d'éléments surveillés dans la file d'attente qui sont retardés sur le serveur ou le proxy Zabbix.	

Éléments spécifiques à Windows

Le tableau fournit des détails sur les clés d'élément prises en charge **uniquement** par l'agent Zabbix Windows.

Les éléments spécifiques à Windows sont parfois une contrepartie approximative d'un élément d'agent similaire ; par exemple, `proc_info`, pris en charge sous Windows, correspond approximativement à l'élément `proc.mem`, qui n'est pas pris en charge sous Windows.

La clé d'élément est un lien vers les détails complets de la clé d'élément.

Clé d'élément	Description	Groupe d'éléments
eventlog	La surveillance du journal des événements Windows.	Surveillance des journaux
eventlog.count	Le nombre de lignes dans le journal des événements Windows.	
net.if.list	La liste des interfaces réseau (inclut le type d'interface, l'état, l'adresse IPv4, la description).	Réseau
perf_counter	La valeur de n'importe quel compteur de performance Windows.	Compteurs de performance
perf_counter_en	La valeur de n'importe quel compteur de performance Windows en anglais.	
perf_instance.discovery	La liste des instances d'objet des compteurs de performance Windows.	
perf_instance_en.discovery	La liste des instances d'objet des compteurs de performance Windows, découvertes à l'aide des noms d'objet en anglais.	
proc_info	Diverses informations sur un ou plusieurs processus spécifiques.	Processus
registry.data	Renvoie les données pour le nom de valeur spécifié dans la clé du Registre Windows.	Registre
registry.get	La liste des valeurs ou des clés du Registre Windows situées à la clé donnée.	
service.discovery	La liste des services Windows.	Services
service.info	Informations sur un service.	
services	La liste des services.	
vm.vmemory.size	La taille de la mémoire virtuelle en octets ou en pourcentage du total.	Mémoire virtuelle
wmi.get	Exécute une requête WMI et renvoie le premier objet sélectionné.	WMI
wmi.getall	Exécute une requête WMI et renvoie la réponse complète.	

Détails de la clé d'élément

Les paramètres sans crochets angulaires sont obligatoires. Les paramètres marqués par des crochets angulaires **< >** sont facultatifs.

eventlog[name,<regexp>,<severity>,<source>,<eventid>,<maxlines>,<mode>]

 La surveillance du journal des événements.
 Valeur de retour: *Log*.

Paramètres:

- **name** - le nom du canal du journal des événements (*Log Name* dans l'interface graphique de l'Observateur d'événements);
- **regexp** - une expression régulière décrivant le motif requis (sensible à la casse);
- **severity** - une expression régulière décrivant la gravité (insensible à la casse). Ce paramètre accepte une expression régulière basée sur les valeurs suivantes: "Information", "Warning", "Error", "Critical", "Verbose" (sous Windows Vista ou version ultérieure).
- **source** - une expression régulière décrivant l'identifiant de la source (insensible à la casse);
- **eventid** - une expression régulière décrivant l'identifiant ou les identifiants de l'événement (sensible à la casse);
- **maxlines** - le nombre maximal de nouvelles lignes par seconde que l'agent enverra au serveur Zabbix ou au proxy. Ce paramètre remplace la valeur de `MaxLinesPerSecond` dans `zabbix_agentd.conf`.

- **mode** - valeurs possibles:
  - *all* (par défaut);
  - *skip* - ignore le traitement des données plus anciennes (n'affecte que les éléments nouvellement créés).

Commentaires:

- L'élément doit être configuré comme une vérification active;
- L'agent ne peut pas envoyer d'événements provenant du journal "Forwarded events";
- Windows Eventing 6.0 est pris en charge;
- Le choix d'un type d'information autre que Log pour cet élément entraînera la perte de l'horodatage local, ainsi que des informations de gravité et de source du journal;
- Voir aussi des informations supplémentaires sur la surveillance des journaux.

Exemples:

```
eventlog[Application]
eventlog[Microsoft-Windows-Application-Experience/Program-Compatibility-Assistant]
eventlog[Security,,"Failure Audit",,^(529|680)$]
eventlog[System,,"Warning|Error"]
eventlog[System,,,,^1$]
eventlog[Windows PowerShell,,,,,,skip]
eventlog[System,,,,@TWOSHORT] #here a custom regular expression named `TWOSHORT` is referenced (defined as
```

eventlog.count[name,<regexp>,<severity>,<source>,<eventid>,<maxproclines>,<mode>]

 Nombre de lignes dans le journal des événements Windows.
 Valeur de retour: *Integer*.

Paramètres:

- **name** - nom du canal du journal des événements (*Log Name* dans l'interface graphique Event Viewer);
- **regexp** - une expression régulière décrivant le motif requis (sensible à la casse);
- **severity** - une expression régulière décrivant la gravité (insensible à la casse). Ce paramètre accepte une expression régulière basée sur les valeurs suivantes: "Information", "Warning", "Error", "Critical", "Verbose" (sous Windows Vista ou version ultérieure).
- **source** - une expression régulière décrivant l'identifiant de la source (insensible à la casse);
- **eventid** - une expression régulière décrivant le ou les identifiants d'événement (sensible à la casse);
- **maxproclines** - nombre maximal de nouvelles lignes par seconde que l'agent analysera (ne peut pas dépasser 10000). La valeur par défaut est 10*`MaxLinesPerSecond` dans `zabbix_agentd.conf`.
- **mode** - valeurs possibles:
  - *all* (par défaut);
  - *skip* - ignorer le traitement des données plus anciennes (n'affecte que les éléments nouvellement créés).

Commentaires:

- L'élément doit être configuré comme une vérification active;
- L'agent ne peut pas envoyer d'événements depuis le journal "Forwarded events";
- Windows Eventing 6.0 est pris en charge;
- Le fait de sélectionner pour cet élément un type d'information autre que Log entraînera la perte de l'horodatage local, ainsi que des informations de gravité et de source du journal;
- Voir aussi des informations supplémentaires sur la surveillance des journaux.

Exemples:

```
eventlog.count[System,,"Warning|Error"]
eventlog.count[Windows PowerShell,,,,,,skip]
```

net.if.list

 La liste des interfaces réseau (inclut le type d'interface, le statut, l'adresse IPv4, la description).
 Valeur de retour : *Texte*.

Commentaires :

- Les noms d'interface multi-octets sont pris en charge ;
- Les interfaces désactivées ne sont pas répertoriées ;
- L'activation/la désactivation de certains composants peut modifier leur ordre dans le nom d'interface Windows ;
- Certaines versions de Windows (par exemple, Server 2008) peuvent nécessiter l'installation des dernières mises à jour pour prendre en charge les caractères non ASCII dans les noms d'interface.

perf_counter[counter,<interval>]

 La valeur de n'importe quel compteur de performance Windows.
 Valeur de retour : *Integer*, *float*, *string* ou *text* (selon la requête).

Paramètres :

- **counter** - le chemin vers le compteur;
- **interval** - les N dernières secondes pour stocker la valeur moyenne. Le `interval` doit être compris entre 1 et 900 secondes (inclus) et la valeur par défaut est 1.

Commentaires :

- Cet élément suit les chemins UNC, ce qui peut constituer un risque de sécurité ;
- `interval` est utilisé pour les compteurs qui nécessitent plus d'un échantillon (comme l'utilisation du CPU), de sorte que la vérification renvoie à chaque fois une valeur moyenne pour les dernières secondes "interval" ;
- Performance Monitor peut être utilisé pour obtenir la liste des compteurs disponibles.
- Voir aussi : compteurs de performance Windows.

perf_counter_en[counter,<interval>]

 La valeur de n'importe quel compteur de performance Windows en anglais.
 Valeur de retour: *Integer*, *float*, *string* ou *text* (selon la requête).

Paramètres:

- **counter** - le chemin vers le compteur en anglais;
- **interval** - les N dernières secondes pour le stockage de la valeur moyenne. Le `interval` doit être compris entre 1 et 900 secondes (inclus) et la valeur par défaut est 1.

Commentaires:

- Cet élément suit les chemins UNC, ce qui peut constituer un risque de sécurité;
- `interval` est utilisé pour les compteurs qui nécessitent plus d'un échantillon (comme l'utilisation du CPU), de sorte que la vérification renvoie à chaque fois une valeur moyenne pour les dernières secondes "interval";
- Cet élément n'est pris en charge que sur **Windows Server 2008/Vista** et versions ultérieures;
- Vous pouvez trouver la liste des chaînes en anglais en affichant la clé de registre suivante: `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Perflib\009`.

perf_instance.discovery[object]

 La liste des instances d'objet des compteurs de performance Windows. Utilisée pour la découverte de bas niveau.
 Valeur de retour : *objet JSON*.

Paramètre :

- **object** - le nom de l'objet (localisé).

perf_instance_en.discovery[object]

 La liste des instances d'objet des compteurs de performance Windows, découvertes à l'aide des noms d'objet en anglais. Utilisé pour la découverte de bas niveau.
 Valeur de retour : *objet JSON*.

Paramètre :

- **object** - le nom de l'objet (en anglais).

proc_info[process,<attribute>,<type>]

 Diverses informations sur un ou plusieurs processus spécifiques.
 Valeur de retour: *Float*.

Paramètres:

- **process** - le nom du processus;
- **attribute** - l'attribut du processus demandé;
- **type** - le type de représentation (pertinent lorsqu'il existe plus d'un processus portant le même nom)

Commentaires:

- Les `attributes` suivants sont pris en charge:
  - *vmsize* (par défaut) - taille de la mémoire virtuelle du processus en Kbytes;
  - *wkset* - taille du working set du processus (quantité de mémoire physique utilisée par le processus) en Kbytes;
  - *pf* - nombre de défauts de page;
  - *ktime* - temps noyau du processus en millisecondes;
  - *utime* - temps utilisateur du processus en millisecondes;
  - *io_read_b* - nombre d'octets lus par le processus lors des opérations d'E/S;
  - *io_read_op* - nombre d'opérations de lecture effectuées par le processus;

    - *io_write_b* - nombre d'octets écrits par le processus lors des opérations d'E/S;
    - *io_write_op* - nombre d'opérations d'écriture effectuées par le processus;
    - *io_other_b* - nombre d'octets transférés par le processus lors d'opérations autres que les opérations de lecture et d'écriture;
    - *io_other_op* - nombre d'opérations d'E/S effectuées par le processus, autres que les opérations de lecture et d'écriture;
    - *gdiobj* - nombre d'objets GDI utilisés par le processus;
    - *userobj* - nombre d'objets USER utilisés par le processus.
- Les types valides sont:
    - *avg* (par défaut) - valeur moyenne pour tous les processus nommés <process>;
    - *min* - valeur minimale parmi tous les processus nommés <process>;
    - *max* - valeur maximale parmi tous les processus nommés <process>;
    - *sum* - somme des valeurs pour tous les processus nommés <process>.
- Sur un système 64 bits, un agent Zabbix 64 bits est requis pour que cet élément fonctionne correctement.

Exemples:

```
proc_info[iexplore.exe,wkset,sum] #récupérer la quantité de mémoire physique utilisée par tous les process
proc_info[iexplore.exe,pf,avg] #récupérer le nombre moyen de défauts de page pour les processus Internet E
```

registry.data[key,<value name>]

 Renvoie les données pour le nom de valeur spécifié dans la clé du Registre Windows.
 Valeur de retour: *Integer*, *string* ou *text* (selon le type de valeur)

Paramètres:

- **key** - la clé du Registre, y compris la clé racine; les abréviations de clé racine (par ex. HKLM) sont autorisées;
- **value name** - le nom de la valeur du Registre dans la clé (chaîne vide "" par défaut). La valeur par défaut est renvoyée si le nom de valeur n'est pas fourni.

Commentaires:

- Abréviations de clé racine prises en charge:
    - HKCR - HKEY_CLASSES_ROOT
    - HKCC - HKEY_CURRENT_CONFIG
    - HKCU - HKEY_CURRENT_USER
    - HKCULS - HKEY_CURRENT_USER_LOCAL_SETTINGS
    - HKLM - HKEY_LOCAL_MACHINE
    - HKPD - HKEY_PERFORMANCE_DATA
    - HKPN - HKEY_PERFORMANCE_NLSTEXT
    - HKPT - HKEY_PERFORMANCE_TEXT
    - HKU - HKEY_USERS
- Les clés contenant des espaces doivent être placées entre guillemets doubles.

Exemples:

```
registry.data["HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\Windows Error Reporting"] #renvoie les donnée
registry.data["HKLM\SOFTWARE\Microsoft\Windows\Windows Error Reporting","EnableZip"] #renvoie les données
```

registry.get[key,<mode>,<name regexp>]

 La liste des valeurs ou des clés du Registre Windows situées à la clé donnée.
 Valeur de retour: *objet JSON*.

Paramètres:

- **key** - la clé du Registre, y compris la clé racine; les abréviations de clé racine (par exemple HKLM) sont autorisées (voir les commentaires de `registry.data[]` pour la liste complète des abréviations);
- **mode** - valeurs possibles:
values (par défaut) ou *keys*;
- **name regexp** - ne découvrir que les valeurs dont les noms correspondent à l'expression régulière (par défaut - découvrir toutes les valeurs). Autorisé uniquement avec *values* comme *mode*.

Les clés contenant des espaces doivent être placées entre guillemets doubles.

Exemples:

```
registry.get[HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall,values,"^DisplayName|DisplayVersion$
Le JSON inclura les détails de la clé, de la dernière sous-clé, du nom de la valeur, du type de valeur et
registry.get[HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall,values] #retourne les données de tou
Le JSON inclura les détails de la clé, de la dernière sous-clé, du nom de la valeur, du type de valeur et
registry.get[HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall,keys] #retourne toutes les sous-clés
Le JSON inclura les détails de la clé et de la dernière sous-clé.
```

service.discovery

 La liste des services Windows. Utilisé pour la découverte de bas niveau.
 Valeur de retour : *objet JSON*.

service.info[service,<param>]

 Informations sur un service.
 Valeur de retour : - *Integer* - avec `param` comme *state*, *startup* ;

 Plus précisément pour *state* : - 0 - en cours d'exécution - 1 - en pause - 2 - démarrage en attente - 3 - pause en attente - 4 - reprise en attente - 5 - arrêt en attente - 6 - arrêté - 7 - inconnu - 255 - aucun service de ce type

 Plus précisément pour *startup* : - 0 - automatique - 1 - automatique différé - 2 - manuel - 3 - désactivé - 4 - inconnu - 5 - démarrage automatique par déclencheur - 6 - démarrage automatique différé par déclencheur - 7 - démarrage manuel par déclencheur - *String* - avec `param` comme *displayname*, *path*, *user* ; - *Text* - avec `param` comme *description*.

Paramètres :

- **service** - un nom de service réel ou son nom d'affichage tel qu'il apparaît dans le composant logiciel enfichable Services de la MMC ;
- **param** - *state* (par défaut), *displayname*, *path*, *user*, *startup*, ou *description*.

Commentaires :

- Les éléments comme `service.info[service,state]` et `service.info[service]` renverront la même information ;
- Uniquement avec `param` comme *state*, cet élément renvoie une valeur pour les services inexistants (255).

Exemples :

```
service.info[SNMPTRAP] - état du service SNMPTRAP ;
service.info[SNMP Trap] - état du même service, mais avec le nom d'affichage spécifié ;
service.info[EventLog,startup] - type de démarrage du service EventLog
```

services[<type>,<state>,<exclude>]

 La liste des services.
 Valeur de retour : *0* - si vide ; *Texte* - la liste des services séparés par un saut de ligne.

Paramètres :

- **type** - *all* (par défaut), *automatic*, *manual* ou *disabled* ;
- **state** - *all* (par défaut), *stopped*, *started*, *start_pending*, *stop_pending*, *running*, *continue_pending*, *pause_pending* ou *paused* ;
- **exclude** - les services à exclure du résultat. Les services exclus doivent être indiqués entre guillemets doubles, séparés par des virgules, sans espaces.

Exemples :

```
services[,started] #renvoie la liste des services démarrés ;
services[automatic, stopped] #renvoie la liste des services arrêtés qui devraient être en cours d'exécution
services[automatic, stopped, "service1,service2,service3"] #renvoie la liste des services arrêtés qui devr
```

vm.vmemory.size[<type>]

 La taille de la mémoire virtuelle en octets ou en pourcentage du total.
 Valeur de retour : *Integer* - pour les octets ; *float* - pour le pourcentage.

Paramètre :

- **type** - valeurs possibles : *available* (mémoire virtuelle disponible), *pavailable* (mémoire virtuelle disponible, en pourcentage), *pused* (mémoire virtuelle utilisée, en pourcentage), *total* (mémoire virtuelle totale, par défaut) ou *used* (mémoire virtuelle utilisée)

Commentaires :

- La surveillance des statistiques de mémoire virtuelle est basée sur :
  - La mémoire virtuelle totale sous Windows (mémoire physique totale + taille du fichier d'échange) ;
  - La quantité maximale de mémoire que l'agent Zabbix peut valider ;
  - La limite actuelle de mémoire validée pour le système ou l'agent Zabbix, selon la plus petite des deux.

Exemple :

```
vm.vmemory.size[pavailable] #retourne la mémoire virtuelle disponible, en pourcentage
```

wmi.get[<namespace>,<query>]

 Exécute une requête WMI et renvoie le premier objet sélectionné.
 Valeur de retour : *Integer*, *float*, *string* ou *text* (selon la requête).

Paramètres :

- **namespace** - l'espace de noms WMI ;
- **query** - la requête WMI renvoyant un seul objet.

Les requêtes WMI sont effectuées avec WQL.

Exemple :

```
wmi.get[root\cimv2,select status from Win32_DiskDrive where Name like '%PHYSICALDRIVE0%'] #renvoie le stat
```

wmi.getall[<namespace>,<query>]

 Exécute une requête WMI et renvoie la réponse complète. Peut être utilisé pour la découverte de bas niveau.
 Valeur de retour : *objet JSON*

Paramètres :

- **namespace** - l'espace de noms WMI ;
- **query** - la requête WMI.

Commentaires :

- Les requêtes WMI sont effectuées avec WQL.
- Le prétraitement JSONPath peut être utilisé pour cibler des valeurs plus spécifiques dans le JSON renvoyé.

Exemple :

```
wmi.getall[root\cimv2,select * from Win32_DiskDrive where Name like '%PHYSICALDRIVE%'] #renvoie des inform
```

Surveillance des services Windows

Ce tutoriel fournit des instructions étape par étape pour configurer la surveillance des services Windows. Il est supposé que le serveur Zabbix et l'agent sont configurés et opérationnels.

Étape 1

Obtenez le nom du service.

Vous pouvez obtenir le nom du service en ouvrant le composant logiciel enfichable Services de la MMC et en affichant les propriétés du service. Dans l'onglet *Général*, vous devriez voir un champ appelé *Nom du service*. La valeur qui suit est le nom que vous utiliserez lors de la configuration d'un élément à surveiller. Par exemple, si vous souhaitiez surveiller le service "workstation", votre service pourrait être : **lanmanworkstation**.

Étape 2

Configurez un élément pour surveiller le service.

L'élément `service.info[service,<param>]` récupère des informations sur un service particulier. Selon les informations dont vous avez besoin, spécifiez l'option `param`, qui accepte les valeurs suivantes : *displayname*, *state*, *path*, *user*, *startup* ou *description*. La valeur par défaut est *state* si `param` n'est pas spécifié (`service.info[service]`).

Le type de valeur renvoyée dépend du `param` choisi : entier pour *state* et *startup* ; chaîne de caractères pour *displayname*, *path* et *user* ; texte pour *description*.

Exemple :

- *Clé :* `service.info[lanmanworkstation]`
- *Type d'information :* Numérique (non signé)

L'élément `service.info[lanmanworkstation]` récupérera des informations sur l'état du service sous forme de valeur numérique. Pour associer une valeur numérique à une représentation textuelle dans l'interface web (« 0 » comme « En cours d'exécution », « 1 » comme « En pause », etc.), vous pouvez configurer un mapping de valeurs sur l'hôte sur lequel l'élément est configuré. Pour ce faire, liez soit le modèle *Windows services by Zabbix agent* ou *Windows services by Zabbix agent active* à l'hôte, soit configurez sur l'hôte un nouveau mapping de valeurs basé sur le mapping de valeurs *Windows service state* configuré sur les modèles mentionnés.

Notez que les deux modèles mentionnés disposent d'une règle de découverte configurée qui découvrira automatiquement les services. Si vous ne le souhaitez pas, vous pouvez désactiver la règle de découverte au niveau de l'hôte une fois le modèle lié à l'hôte.

Découverte des services Windows

La découverte de bas niveau permet de créer automatiquement des éléments, des déclencheurs et des graphiques pour différentes entités sur un ordinateur. Zabbix peut commencer automatiquement à surveiller les services Windows sur votre machine, sans

qu'il soit nécessaire de connaître le nom exact d'un service ni de créer manuellement des éléments pour chaque service. Un filtre peut être utilisé pour générer des éléments, des déclencheurs et des graphiques réels uniquement pour les services pertinents.

Surveillance des fichiers journaux

Vue d'ensemble

Zabbix peut être utilisé pour la surveillance centralisée et l'analyse de fichiers journaux avec ou sans prise en charge de la rotation des journaux.

Les notifications peuvent être utilisées pour avertir les utilisateurs lorsqu'un fichier journal contient certaines chaînes ou certains motifs de chaînes.

Pour surveiller un fichier journal, vous devez disposer de :

- Un agent Zabbix en cours d'exécution sur l'hôte.
- Un élément de surveillance des journaux configuré.

Attention:
La limite de taille d'un fichier journal surveillé dépend de la prise en charge des gros fichiers.

Configuration

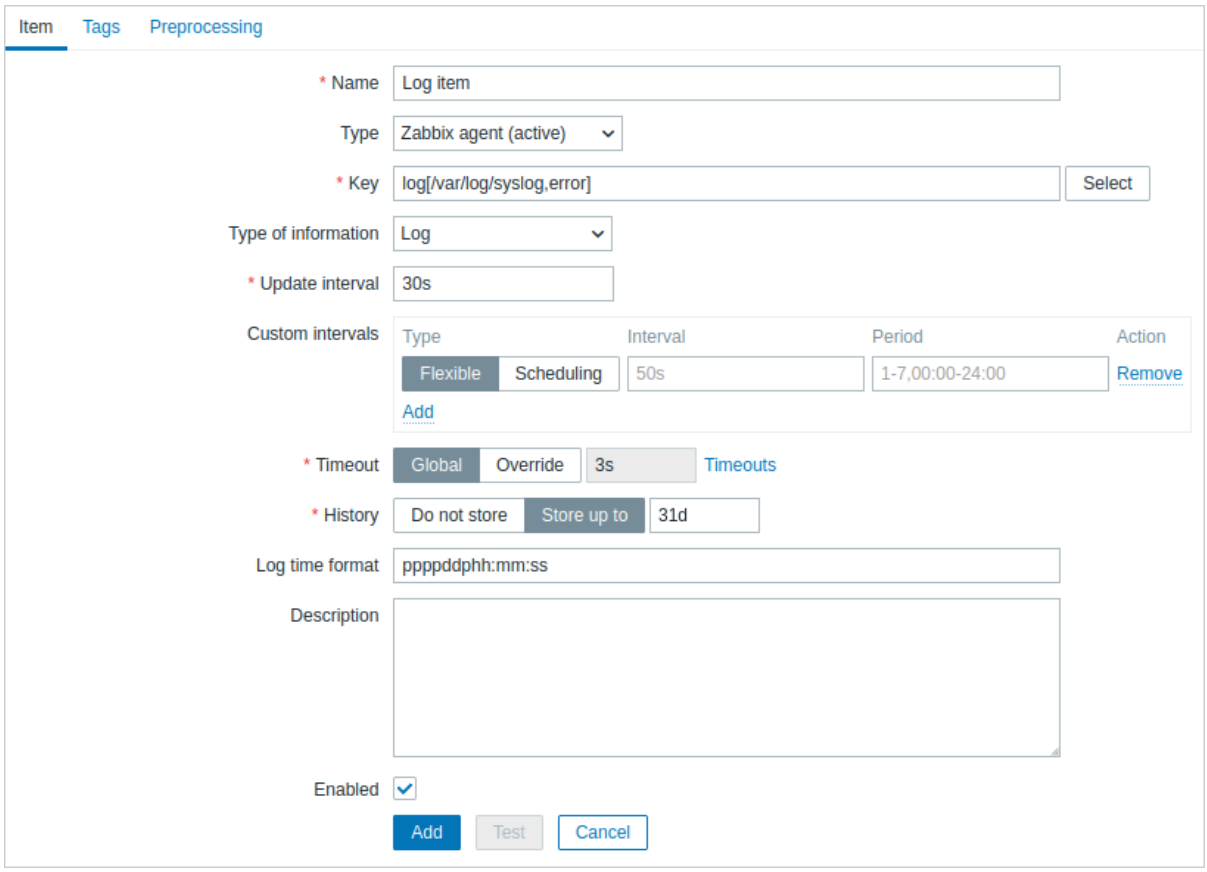
Vérifier les paramètres de l'agent

Assurez-vous que dans le fichier de configuration de l'agent :

- Le paramètre `Hostname` correspond au nom de l'hôte dans l'interface.
- Les serveurs dans le paramètre `ServerActive` sont spécifiés pour le traitement des vérifications actives.

Configuration de l'élément

Configurez un élément de surveillance des journaux.



Tous les champs obligatoires sont marqués d'un astérisque rouge.

Pour les éléments de surveillance des journaux, saisissez notamment :

Type	Sélectionnez **Zabbix agent (active)** ici.
Key	Utilisez l'une des clés d'élément suivantes : `log[]` ou `logrt[]` : Ces deux clés d'élément vous permettent de surveiller les journaux et de filtrer les entrées de journal selon l'expression régulière du contenu, si elle est présente. Par exemple : `log[/var/log/syslog,error]`. Assurez-vous que le fichier dispose des autorisations de lecture pour l'utilisateur « zabbix » ; sinon, l'état de l'élément sera défini sur « unsupported ». **log.count[]** ou **logrt.count[]** : Ces deux clés d'élément vous permettent de renvoyer uniquement le nombre de lignes correspondantes. Consultez la section consacrée aux clés d'élément de Zabbix agent prises en charge pour plus d'informations sur l'utilisation de ces clés d'élément et de leurs paramètres.
Type of information	Prérempli automatiquement : Pour les éléments `log[]` ou `logrt[]` : *Log* ; Pour les éléments `log.count[]` ou `logrt.count[]` : *Numeric (unsigned)*. Si vous utilisez éventuellement le paramètre `output`, vous pouvez sélectionner manuellement un type d'information approprié autre que *Log*. Notez que le choix d'un type d'information autre que *Log* entraînera la perte de l'horodatage local.
Update interval (in sec)	Ce paramètre définit la fréquence à laquelle Zabbix agent vérifie les modifications dans le fichier journal. En le définissant sur 1 seconde, vous vous assurez de recevoir les nouveaux enregistrements dès que possible.
Log time format	Dans ce champ, vous pouvez éventuellement spécifier le modèle d'analyse de l'horodatage de la ligne de journal. Espaces réservés pris en charge : * **y** : *Année (1970-2038)* * **M** : *Mois (01-12)* * **d** : *Jour (01-31)* * **h** : *Heure (00-23)* * **m** : *Minute (00-59)* * **s** : *Seconde (00-59)* Si ce champ est laissé vide, l'horodatage sera défini sur 0 au format Unix, ce qui représente le `1er janvier 1970`. Par exemple, considérons la ligne suivante provenant du fichier journal de Zabbix agent : `23480:20100328:154718.045 Zabbix agent started. Zabbix 1.8.2 (revision 11211).` Elle commence par six positions de caractères correspondant au PID, suivies de la date, de l'heure et du reste du message. Le format de l'heure du journal pour cette ligne serait « ppppp:yyyyMMdd:hhmmss ». Notez que les caractères « p » et « : » sont des espaces réservés et peuvent être remplacés par n'importe quel caractère, à l'exception de « yMdhms ».

Notes importantes

- Le serveur et l'agent conservent la trace de la taille d'un journal surveillé et de sa dernière heure de modification (pour `logrt`) dans deux compteurs. De plus :
  - L'agent utilise également en interne des numéros d'inode (sur UNIX/GNU/Linux), des index de fichiers (sur Microsoft Windows) et des sommes MD5 des 512 premiers octets du fichier journal afin d'améliorer les décisions lorsque les fichiers journaux sont tronqués et pivotés.
  - Sur les systèmes UNIX/GNU/Linux, il est supposé que les systèmes de fichiers où sont stockés les fichiers journaux renvoient des numéros d'inode, qui peuvent être utilisés pour suivre les fichiers.
  - Sur Microsoft Windows, l'agent Zabbix détermine le type de système de fichiers sur lequel se trouvent les fichiers journaux et utilise :
    * Sur les systèmes de fichiers NTFS, des index de fichiers 64 bits.
    * Sur les systèmes de fichiers ReFS (uniquement à partir de Microsoft Windows Server 2012), des ID de fichiers 128 bits.
    * Sur les systèmes de fichiers où les index de fichiers changent (par exemple FAT32, exFAT), un algorithme de repli est utilisé pour adopter une approche raisonnable dans des conditions incertaines lorsque la rotation des fichiers journaux entraîne plusieurs fichiers journaux ayant la même heure de dernière modification.
  - Les numéros d'inode, les index de fichiers et les sommes MD5 sont collectés en interne par l'agent Zabbix. Ils ne sont pas transmis au serveur Zabbix et sont perdus lorsque l'agent Zabbix est arrêté.
  - Ne modifiez pas l'heure de dernière modification d'un fichier journal (par exemple avec `touch`), et ne remplacez pas

un fichier journal surveillé en copiant un fichier sous son nom d'origine (cela crée un nouvel inode). Dans l'un ou l'autre cas, Zabbix peut considérer le fichier comme un fichier différent et le relire depuis le début, ce qui peut produire des alertes en double.

- S'il existe plusieurs fichiers journaux correspondant à `logrt[]` et que l'agent Zabbix suit le plus récent d'entre eux, puis que ce fichier journal le plus récent est supprimé, un message d'avertissement `"there are no files matching "<regexp mask>" in "<directory>"` est consigné. L'agent Zabbix ignore les fichiers journaux dont l'heure de modification est antérieure à la plus récente heure de modification observée par l'agent pour l'élément `logrt[]` en cours de vérification.

- L'agent commence à lire le fichier journal à partir du point où il s'était arrêté la fois précédente.
- Le nombre d'octets déjà analysés (le compteur de taille) et la dernière heure de modification (le compteur de temps) sont stockés dans la base de données Zabbix et sont envoyés à l'agent afin de s'assurer que l'agent commence à lire le fichier journal à partir de ce point lorsque l'agent vient d'être démarré ou a reçu des éléments qui étaient auparavant désactivés ou non pris en charge. Cependant, si l'agent a reçu du serveur un compteur de taille non nul, mais que l'élément `logrt[]` ou `logrt.count[]` ne parvient pas à trouver de fichiers correspondants, le compteur de taille est réinitialisé à 0 afin d'analyser depuis le début si les fichiers apparaissent plus tard.
- Chaque fois que le fichier journal devient plus petit que le compteur de taille du journal connu de l'agent, le compteur est réinitialisé à zéro et l'agent commence à lire le fichier journal depuis le début en tenant compte du compteur de temps.
- S'il existe plusieurs fichiers correspondants ayant la même heure de dernière modification dans le répertoire, l'agent tente alors d'analyser correctement tous les fichiers journaux ayant la même heure de modification et d'éviter de sauter des données ou d'analyser deux fois les mêmes données, bien que cela ne puisse pas être garanti dans toutes les situations. L'agent ne suppose aucun schéma particulier de rotation des fichiers journaux et n'en détermine aucun. Lorsqu'il est présenté avec plusieurs fichiers journaux ayant la même heure de dernière modification, l'agent les traite dans un ordre lexicographique décroissant. Ainsi, pour certains schémas de rotation, les fichiers journaux seront analysés et signalés dans leur ordre d'origine. Pour d'autres schémas de rotation, l'ordre d'origine des fichiers journaux ne sera pas respecté, ce qui peut entraîner la remontée des enregistrements de fichiers journaux correspondants dans un ordre modifié (le problème ne se produit pas si les fichiers journaux ont des heures de dernière modification différentes).
- L'agent Zabbix traite les nouveaux enregistrements d'un fichier journal une fois toutes les *Update interval* secondes.
- L'agent Zabbix n'envoie pas plus de `maxlines` lignes d'un fichier journal par seconde. Cette limite évite de surcharger les ressources réseau et CPU et remplace la valeur par défaut fournie par le paramètre `MaxLinesPerSecond` dans le fichier de configuration de l'agent.
- Pour trouver la chaîne requise, Zabbix traitera 10 fois plus de nouvelles lignes que la valeur définie dans `MaxLinesPerSecond`. Ainsi, par exemple, si un élément `log[]` ou `logrt[]` a un *Update interval* de 1 seconde, par défaut l'agent analysera au plus 200 enregistrements de fichier journal et enverra au plus 20 enregistrements correspondants au serveur Zabbix en une seule vérification. En augmentant `MaxLinesPerSecond` dans le fichier de configuration de l'agent ou en définissant le paramètre `maxlines` dans la clé de l'élément, la limite peut être augmentée jusqu'à 10000 enregistrements de fichier journal analysés et 1000 enregistrements correspondants envoyés au serveur Zabbix en une seule vérification. Si le *Update interval* est défini à 2 secondes, les limites pour une vérification seraient fixées à un niveau 2 fois plus élevé qu'avec un *Update interval* de 1 seconde.
- De plus, les valeurs `log` et `log.count` sont toujours limitées à 50 % de la taille du tampon d'envoi de l'agent, même s'il n'y a aucune valeur non liée aux journaux dans celui-ci. Ainsi, pour que les valeurs `maxlines` soient envoyées en une seule connexion (et non en plusieurs), le paramètre `BufferSize` de l'agent doit être au moins égal à `maxlines` x 2. L'agent Zabbix peut téléverser des données pendant la collecte des journaux et libérer ainsi le tampon, tandis que l'agent Zabbix 2 interrompra la collecte des journaux jusqu'à ce que les données soient téléversées et que le tampon soit libéré, ce qui est effectué de manière asynchrone.
- En l'absence d'éléments de journal, toute la taille du tampon de l'agent est utilisée pour les valeurs non liées aux journaux. Lorsque des valeurs de journal arrivent, elles remplacent au besoin les anciennes valeurs non liées aux journaux, jusqu'à la limite de 50 % définie.
- Pour les enregistrements de fichier journal de plus de 256 kB, seuls les 256 kB initiaux sont comparés à l'expression régulière et le reste de l'enregistrement est ignoré. Cependant, si l'agent Zabbix est arrêté alors qu'il traite un enregistrement long, l'état interne de l'agent est perdu et l'enregistrement long peut être analysé à nouveau et différemment après le redémarrage de l'agent.
- Remarque particulière pour les séparateurs de chemin `\` : si `file_format` est `file\.log`, alors il ne doit pas y avoir de répertoire `file`, car il n'est pas possible de définir sans ambiguïté si `.` est échappé ou s'il s'agit du premier symbole du nom de fichier.
- Les expressions régulières pour `logrt` sont prises en charge uniquement dans le nom de fichier ; la correspondance par expression régulière sur le répertoire n'est pas prise en charge.
- Sur les plateformes UNIX, un élément `logrt[]` devient NOTSUPPORTED si un répertoire dans lequel les fichiers journaux sont censés se trouver n'existe pas.
- Sur Microsoft Windows, si un répertoire n'existe pas, l'élément ne deviendra pas NOTSUPPORTED (par exemple, si le répertoire est mal orthographié dans la clé de l'élément).
- L'absence de fichiers journaux pour un élément `logrt[]` ne le rend pas NOTSUPPORTED. Les erreurs de lecture des fichiers journaux pour un élément `logrt[]` sont consignées comme des avertissements dans le fichier journal de l'agent Zabbix, mais ne rendent pas l'élément NOTSUPPORTED.

- Le fichier journal de l'agent Zabbix peut être utile pour déterminer pourquoi un élément `log[]` ou `logrt[]` est devenu NOTSUPPORTED. Zabbix peut surveiller le fichier journal de son agent, sauf avec `DebugLevel=4` ou `DebugLevel=5`.
- La recherche d'un point d'interrogation à l'aide d'une expression régulière, par exemple `\?`, peut entraîner des faux positifs si le fichier texte contient des symboles NUL, car ceux-ci sont remplacés par ? par Zabbix afin de poursuivre le traitement de la ligne jusqu'au caractère de nouvelle ligne.

Extraction de la partie correspondante d'une expression régulière

Il peut parfois être souhaitable d'extraire uniquement la valeur intéressante d'un fichier cible au lieu de renvoyer la ligne entière lorsqu'une correspondance avec une expression régulière est trouvée.

Les éléments de journal disposent de la capacité d'extraire les valeurs souhaitées des lignes correspondantes. Cela est possible grâce au paramètre supplémentaire **output** des éléments `log` et `logrt`.

L'utilisation du paramètre « output » permet d'indiquer le « groupe capturant » de la correspondance qui nous intéresse.

Par exemple :

```
log[/path/to/the/file,"large result buffer allocation.*Entries: ([0-9]+)",,,,\1]
```

devrait permettre de renvoyer le nombre d'entrées trouvé dans le contenu suivant :

```
Fr Feb 07 2014 11:07:36.6690 */ Thread Id 1400 (GLEWF) large result
buffer allocation - /Length: 437136/Entries: 5948/Client Ver: >=10/RPC
ID: 41726453/User: AUser/Form: CFG:ServiceLevelAgreement
```

Seul le nombre sera renvoyé, car `\1` fait référence au premier et unique groupe capturant : `([0-9]+)`.

Grâce à la possibilité d'extraire et de renvoyer un nombre, cette valeur peut être utilisée pour définir des déclencheurs.

Utilisation du paramètre maxdelay

Le paramètre `maxdelay` dans les éléments de journal permet d'ignorer certaines lignes plus anciennes des fichiers journaux afin que les lignes les plus récentes soient analysées dans les `maxdelay` secondes.

> **Warning:**
> Spécifier `maxdelay` > 0 peut entraîner l'**ignorance d'enregistrements importants du fichier journal et des alertes manquées**. Utilisez-le avec prudence, à vos risques et périls, uniquement lorsque cela est nécessaire.

Par défaut, les éléments de surveillance des journaux suivent toutes les nouvelles lignes apparaissant dans les fichiers journaux. Cependant, certaines applications commencent, dans certaines situations, à écrire un nombre énorme de messages dans leurs fichiers journaux. Par exemple, si une base de données ou un serveur DNS est indisponible, ces applications inondent les fichiers journaux de milliers de messages d'erreur presque identiques jusqu'à ce que le fonctionnement normal soit rétabli. Par défaut, tous ces messages seront analysés consciencieusement et les lignes correspondantes seront envoyées au serveur comme configuré dans les éléments `log` et `logrt`.

La protection intégrée contre la surcharge consiste en un paramètre configurable `maxlines` (qui protège le serveur contre un trop grand nombre de lignes de journal correspondantes entrantes) et une limite de 10*`maxlines` (qui protège le CPU et les E/S de l'hôte contre une surcharge par l'agent lors d'une seule vérification). Néanmoins, il existe 2 problèmes avec cette protection intégrée. Premièrement, un grand nombre de messages potentiellement peu informatifs sont signalés au serveur et occupent de l'espace dans la base de données. Deuxièmement, en raison du nombre limité de lignes analysées par seconde, l'agent peut prendre du retard sur les enregistrements de journal les plus récents pendant des heures. Il est fort probable que vous préfériez être informé plus rapidement de la situation actuelle dans les fichiers journaux plutôt que de parcourir d'anciens enregistrements pendant des heures.

La solution à ces deux problèmes consiste à utiliser le paramètre `maxdelay`. Si `maxdelay` > 0 est spécifié, lors de chaque vérification, le nombre d'octets traités, le nombre d'octets restants et le temps de traitement sont mesurés. À partir de ces valeurs, l'agent calcule un délai estimé - le nombre de secondes qu'il faudrait pour analyser tous les enregistrements restants dans un fichier journal.

Si le délai ne dépasse pas `maxdelay`, l'agent poursuit l'analyse du fichier journal comme d'habitude.

Si le délai est supérieur à `maxdelay`, l'agent **ignore un bloc du fichier journal en le "sautant"** vers une nouvelle position estimée afin que les lignes restantes puissent être analysées dans les `maxdelay` secondes.

Notez que l'agent ne lit même pas les lignes ignorées dans le tampon, mais calcule une position approximative vers laquelle sauter dans un fichier.

Le fait de sauter des lignes du fichier journal est consigné dans le fichier journal de l'agent comme ceci :

```
14287:20160602:174344.206 item:"logrt["/home/zabbix32/test[0-9].log",ERROR,,1000,,,120.0]"
logfile:"/home/zabbix32/test1.log" skipping 679858 bytes
(from byte 75653115 to byte 76332973) to meet maxdelay
```

La valeur ”to byte” est approximative, car après le ”saut” l'agent ajuste la position dans le fichier au début d'une ligne de journal, qui peut se trouver plus loin dans le fichier ou plus tôt.

Selon la vitesse de croissance par rapport à la vitesse d'analyse du fichier journal, vous pouvez ne voir aucun ”saut”, des ”sauts” rares ou fréquents, des ”sauts” grands ou petits, ou même un petit ”saut” à chaque vérification. Les fluctuations de la charge système et la latence réseau affectent également le calcul du délai et donc le fait de ”sauter” en avant pour suivre le paramètre `maxdelay`.

Il n'est pas recommandé de définir `maxdelay` < `update interval` (cela peut entraîner des ”sauts” fréquents et de petite taille).

Notes sur la gestion de la rotation des fichiers journaux avec `copytruncate`

`logrt` avec l'option `copytruncate` suppose que différents fichiers journaux contiennent des enregistrements différents (au moins que leurs horodatages soient différents), par conséquent les sommes MD5 des blocs initiaux (jusqu'aux 512 premiers octets) seront différentes. Deux fichiers ayant les mêmes sommes MD5 de leurs blocs initiaux signifient que l'un d'eux est l'original, l'autre une copie.

`logrt` avec l'option `copytruncate` s'efforce de traiter correctement les copies de fichiers journaux sans signaler de doublons. Cependant, des situations telles que la création de plusieurs copies d'un fichier journal avec le même horodatage, une rotation du fichier journal plus fréquente que l'intervalle de mise à jour de l'élément `logrt[]`, ou des redémarrages fréquents de l'agent ne sont pas recommandées. L'agent tente de gérer raisonnablement bien toutes ces situations, mais de bons résultats ne peuvent pas être garantis dans tous les cas.

Notes sur les fichiers persistants pour les éléments log*[]

Objectif des fichiers persistants

Lorsque l'agent Zabbix est démarré, il reçoit une liste de vérifications actives depuis le serveur Zabbix ou le proxy. Pour les métriques `log*[]`, il reçoit la taille du journal traité et l'heure de modification afin de déterminer à partir de quel point commencer la surveillance du fichier journal. Selon la taille réelle du fichier journal et l'heure de modification signalées par le système de fichiers, l'agent décide soit de poursuivre la surveillance du fichier journal à partir de la taille du journal traité, soit de réanalyser le fichier journal depuis le début.

Un agent en cours d'exécution conserve un ensemble plus large d'attributs pour suivre tous les fichiers journaux surveillés entre les vérifications. Cet état en mémoire est perdu lorsque l'agent est arrêté.

Le nouveau paramètre facultatif `persistent_dir` spécifie un répertoire destiné à stocker cet état de l'élément `log[]`, `log.count[]`, `logrt[]` ou `logrt.count[]` dans un fichier. L'état de l'élément de journal est restauré à partir du fichier persistant après le redémarrage de l'agent Zabbix.

Le principal cas d'utilisation est la surveillance d'un fichier journal situé sur un système de fichiers en miroir. Jusqu'à un certain moment, le fichier journal est écrit sur les deux miroirs. Ensuite, les miroirs sont séparés. Sur la copie active, le fichier journal continue de croître et reçoit de nouveaux enregistrements. L'agent Zabbix l'analyse et envoie la taille du journal traité ainsi que l'heure de modification au serveur. Sur la copie passive, le fichier journal reste inchangé, bien en retard par rapport à la copie active. Plus tard, le système d'exploitation et l'agent Zabbix sont redémarrés à partir de la copie passive. La taille du journal traité et l'heure de modification que l'agent Zabbix reçoit du serveur peuvent ne pas être valides pour la situation sur la copie passive. Pour reprendre la surveillance du fichier journal à l'endroit où l'agent s'était arrêté au moment de la séparation du miroir du système de fichiers, l'agent restaure son état à partir du fichier persistant.

Fonctionnement de l'agent avec un fichier persistant

Au démarrage, l'agent Zabbix ne connaît rien des fichiers persistants. Ce n'est qu'après avoir reçu une liste de vérifications actives du serveur Zabbix (proxy) que l'agent constate que certains éléments de journal doivent être pris en charge par des fichiers persistants dans des répertoires spécifiés.

Pendant le fonctionnement de l'agent, les fichiers persistants sont ouverts en écriture (avec `fopen(filename, "w")`) et écrasés avec les données les plus récentes. Le risque de perdre des données du fichier persistant si l'écrasement et la séparation du miroir du système de fichiers se produisent en même temps est très faible ; aucun traitement spécial n'est prévu à cet effet. L'écriture dans le fichier persistant n'est PAS suivie d'une synchronisation forcée vers le support de stockage (`fsync()` n'est pas appelée).

L'écrasement avec les données les plus récentes est effectué après le signalement réussi à Zabbix server de l'enregistrement correspondant du fichier journal ou des métadonnées (taille du journal traitée et heure de modification). Cela peut se produire aussi souvent qu'à chaque vérification de l'élément si le fichier journal continue de changer.

Aucune action spéciale n'est effectuée lors de l'arrêt de l'agent.

Après avoir reçu une liste de vérifications actives, l'agent marque les fichiers persistants obsolètes pour suppression. Un fichier persistant devient obsolète si :

1. L'élément de journal correspondant n'est plus surveillé.
2. Un élément de journal est reconfiguré avec un emplacement `persistent_dir` différent de celui d'avant.

La suppression est effectuée avec un délai de 24 heures, car les fichiers journaux à l'état NOTSUPPORTED ne sont pas inclus dans la liste des vérifications actives, mais ils peuvent devenir SUPPORTED plus tard et leurs fichiers persistants seront utiles.

Si l'agent est arrêté avant l'expiration des 24 heures, les fichiers obsolètes ne seront pas supprimés, car l'agent Zabbix ne reçoit plus d'informations sur leur emplacement depuis le serveur Zabbix.

Warning:
Reconfigurer le `persistent_dir` d'un élément de journal vers l'ancien emplacement `persistent_dir` alors que l'agent est arrêté, sans supprimer l'ancien fichier persistant par l'utilisateur, entraînera la restauration de l'état de l'agent à partir de l'ancien fichier persistant, ce qui provoquera des messages manqués ou de fausses alertes.

Nommage et emplacement des fichiers persistants

L'agent Zabbix distingue les contrôles actifs par leurs clés. Par exemple, `logrt[/home/zabbix/test.log]` et `logrt[/home/zabbix/tes` sont des éléments différents. La modification de l'élément `logrt[/home/zabbix/test.log,,,10]` dans l'interface en `logrt[/home/zabbix/test.log,,,20]` entraîne la suppression de l'élément `logrt[/home/zabbix/test.log,,,10]` de la liste des contrôles actifs de l'agent et la création de l'élément `logrt[/home/zabbix/test.log,,,20]` (certains attributs sont conservés lors de la modification dans l'interface/serveur, mais pas dans l'agent).

Le nom du fichier est composé de la somme MD5 de la clé de l'élément, à laquelle est ajoutée la longueur de la clé afin de réduire le risque de collisions. Par exemple, l'état de l'élément `logrt[/home/zabbix50/test.log,,,,,,,,/home/zabbix50/agent_private]` sera conservé dans le fichier persistant c963ade4008054813bbc0a650bb8e09266.

Plusieurs éléments de journal peuvent utiliser la même valeur de `persistent_dir`.

`persistent_dir` est spécifié en tenant compte de la configuration spécifique des systèmes de fichiers, des points de montage et des options de montage, ainsi que de la configuration de la mise en miroir du stockage - le fichier persistant doit se trouver sur le même système de fichiers mis en miroir que le fichier journal surveillé.

Si le répertoire `persistent_dir` ne peut pas être créé ou n'existe pas, ou si l'agent Zabbix ne dispose pas des autorisations suffisantes pour créer, écrire, lire ou supprimer des fichiers, l'élément de journal devient `NOTSUPPORTED`.

Si les droits d'accès aux fichiers de stockage persistant sont supprimés pendant le fonctionnement de l'agent ou si d'autres erreurs surviennent (par exemple, disque plein), les erreurs sont consignées dans le fichier journal de l'agent, mais l'élément de journal ne devient pas `NOTSUPPORTED`.

Charge sur les E/S

Le fichier persistant de l'élément est mis à jour après l'envoi réussi de chaque lot de données (contenant les données de l'élément) au serveur. Par exemple, la valeur par défaut de `BufferSize` est 100. Si un élément de journal a trouvé 70 enregistrements correspondants, alors les 50 premiers enregistrements seront envoyés dans un premier lot, le fichier persistant sera mis à jour, puis les 20 enregistrements restants seront envoyés (éventuellement avec un certain délai lorsque davantage de données sont accumulées) dans le deuxième lot, et le fichier persistant sera à nouveau mis à jour.

Actions si la communication échoue entre l'agent et le serveur

Chaque ligne correspondante des éléments `log[]` et `logrt[]`, ainsi que le résultat de chaque vérification des éléments `log.count[]` et `logrt.count[]`, nécessite un emplacement libre dans la zone dédiée de 50 % du tampon d'envoi de l'agent. Les éléments du tampon sont régulièrement envoyés au serveur (ou au proxy) et les emplacements du tampon sont alors de nouveau libres.

Tant qu'il reste des emplacements libres dans la zone dédiée aux journaux du tampon d'envoi de l'agent et que la communication échoue entre l'agent et le serveur (ou le proxy), les résultats de surveillance des journaux sont accumulés dans le tampon d'envoi. Cela permet d'atténuer les brèves interruptions de communication.

Lors de pannes de communication plus longues, tous les emplacements de journaux sont occupés et les actions suivantes sont effectuées :

- Les vérifications des éléments `log[]` et `logrt[]` sont arrêtées. Lorsque la communication est rétablie et que des emplacements libres sont disponibles dans le tampon, les vérifications reprennent à partir de la position précédente. Aucune ligne correspondante n'est perdue, elle est simplement signalée plus tard.
- Les vérifications `log.count[]` et `logrt.count[]` sont arrêtées si `maxdelay = 0` (par défaut). Le comportement est similaire à celui des éléments `log[]` et `logrt[]` décrits ci-dessus. Notez que cela peut affecter les résultats de `log.count[]` et `logrt.count[]` : par exemple, une vérification compte 100 lignes correspondantes dans un fichier journal, mais comme aucun emplacement n'est libre dans le tampon, la vérification est arrêtée. Lorsque la communication est rétablie, l'agent

compte les mêmes 100 lignes correspondantes ainsi que 70 nouvelles lignes correspondantes. L'agent envoie alors count = 170 comme si elles avaient été trouvées lors d'une seule vérification.

- Les vérifications `log.count[]` et `logrt.count[]` avec `maxdelay` > 0 : s'il n'y a pas eu de "saut" pendant la vérification, le comportement est similaire à celui décrit ci-dessus. Si un "saut" au-dessus de lignes du fichier journal a eu lieu, alors la position après le "saut" est conservée et le résultat compté est ignoré. Ainsi, l'agent essaie de suivre un fichier journal en croissance même en cas de panne de communication.

Gestion des erreurs de compilation et d'exécution des expressions régulières

Si une expression régulière utilisée dans un élément `log[]`, `logrt[]`, `log.count[]` ou `logrt.count[]` ne peut pas être compilée par la bibliothèque PCRE ou PCRE2, l'élément passe à l'état NON PRIS EN CHARGE avec un message d'erreur. Pour continuer à surveiller l'élément de journal, l'expression régulière doit être corrigée.

Si l'expression régulière est compilée avec succès, mais échoue lors de l'exécution (pour certains ou pour tous les enregistrements du journal), l'élément de journal reste pris en charge et la surveillance continue. L'erreur d'exécution est enregistrée dans le fichier journal de l'agent Zabbix (sans l'enregistrement du fichier journal).

La fréquence d'enregistrement est limitée à une erreur d'exécution par vérification afin de permettre à l'agent Zabbix de surveiller son propre fichier journal. Par exemple, si 10 enregistrements sont analysés et que 3 enregistrements échouent en raison d'une erreur d'exécution d'une expression régulière, un seul enregistrement est généré dans le journal de l'agent.

Exception : si `MaxLinesPerSecond=1` et `update interval=1` (ce qui ne permet d'analyser qu'un seul enregistrement par vérification), les erreurs d'exécution des expressions régulières ne sont pas enregistrées.

zabbix_agentd consigne la clé de l'élément en cas d'erreur d'exécution, tandis que zabbix_agent2 consigne l'ID de l'élément afin d'aider à identifier l'élément de journal présentant des erreurs d'exécution. Il est recommandé de repenser l'expression régulière en cas d'erreurs d'exécution.

Vérification simple

Aperçu

Les vérifications simples sont normalement utilisées pour des vérifications distantes sans agent de services.

Notez que Zabbix agent n'est pas nécessaire pour les vérifications simples. Le serveur/proxy Zabbix est responsable du traitement des vérifications simples (établissement de connexions externes, etc.).

Exemples d'utilisation des vérifications simples :

```
net.tcp.service[ftp,,155]
net.tcp.service[http]
net.tcp.service.perf[http,,8080]
net.udp.service.perf[ntp]
```

> **Note:**
> Les champs *Nom d'utilisateur* et *Mot de passe* (limités à 255 caractères) dans la configuration d'un élément de vérification simple sont utilisés pour les éléments de supervision VMware ; ils sont ignorés dans les autres cas.

Vérifications prises en charge

Les clés d'élément sont listées sans paramètres facultatifs ni informations supplémentaires. Cliquez sur la clé d'élément pour voir les détails complets.

Voir aussi clés d'élément de surveillance VMware.

Item key	Description
icmpping	L'accessibilité de l'hôte par ping ICMP.
icmppingloss	Le pourcentage de paquets perdus.
icmppingretry	L'accessibilité de l'hôte par ping ICMP avec tentatives de reprise.
icmppingsec	Le temps de réponse du ping ICMP.
net.tcp.service	Vérifie si un service est en cours d'exécution et accepte les connexions TCP.
net.tcp.service.perf	Vérifie les performances d'un service TCP.
net.udp.service	Vérifie si un service est en cours d'exécution et répond aux requêtes UDP.
net.udp.service.perf	Vérifie les performances d'un service UDP.

Détails de la clé d'élément

Les paramètres sans crochets angulaires sont obligatoires. Les paramètres marqués par des crochets angulaires **< >** sont facultatifs.

icmpping[<target>,<packets>,<interval>,<size>,<timeout>,<options>]

 L'accessibilité de l'hôte par ping ICMP.
 Valeur de retour: *0* - le ping ICMP échoue; *1* - le ping ICMP réussit.

Paramètres:

- **target** - l'adresse IP ou le nom DNS de l'hôte;
- **packets** - le nombre de paquets;
- **interval** - le temps entre les paquets successifs en millisecondes;
- **size** - la taille du paquet en octets;
- **timeout** - le délai d'attente en millisecondes;
- **options** - utilisé pour autoriser la redirection: si vide (valeur par défaut), les réponses redirigées sont considérées comme indiquant que l'hôte cible est hors ligne; si défini sur *allow_redirect*, les réponses redirigées sont considérées comme indiquant que l'hôte cible est en ligne.

Voir aussi le tableau des valeurs par défaut.

Commentaire:

- Si au moins un des paquets définis par le paramètre `packets` est renvoyé, l'élément renvoie 1. Si aucun des paquets n'est renvoyé, l'élément renvoie 0.

Exemple:

```
icmpping[,4] #Si au moins un des quatre paquets est renvoyé, l'élément renvoie 1.
```

icmppingloss[<target>,<packets>,<interval>,<size>,<timeout>,<options>]

 Le pourcentage de paquets perdus.
 Valeur de retour : *Float*.

Paramètres :

- **target** - l'adresse IP ou le nom DNS de l'hôte ;
- **packets** - le nombre de paquets ;
- **interval** - le temps entre les paquets successifs, en millisecondes ;
- **size** - la taille du paquet en octets ;
- **timeout** - le délai d'expiration en millisecondes ;
- **options** - utilisé pour autoriser la redirection : si vide (valeur par défaut), les réponses redirigées sont traitées comme si l'hôte cible était indisponible ; si défini sur *allow_redirect*, les réponses redirigées sont traitées comme si l'hôte cible était disponible.

Voir aussi le tableau des valeurs par défaut.

icmppingretry[<target>,<retries>,<backoff>,<size>,<timeout>,<options>]

 L'accessibilité de l'hôte par ping ICMP avec nouvelles tentatives. Si le premier paquet réussit, l'opération s'arrête ; si le paquet échoue, une nouvelle tentative est effectuée jusqu'à ce que le nombre maximal de tentatives défini dans le paramètre `retries` soit atteint. Cet élément est utile pour réduire le nombre de paquets envoyés sur le réseau.
 Valeur de retour : *0* - échec du ping ICMP ; *1* - ping ICMP réussi.

Paramètres :

- **target** - l'adresse IP ou le nom DNS de l'hôte ;
- **retries** - le nombre de tentatives d'envoi d'un ping vers une cible, sans compter le premier essai (0 ou plus ; valeur par défaut : 1) ;
- **backoff** - le nombre par lequel le temps d'attente est multiplié à chaque requête successive (plage de 1.0 à 5.0 ; valeur par défaut : 1.0) ;
- **size** - la taille du paquet en octets ;
- **timeout** - le délai d'expiration en millisecondes ;
- **options** - utilisé pour autoriser la redirection : si vide (valeur par défaut), les réponses redirigées sont traitées comme si l'hôte cible était indisponible ; si défini sur *allow_redirect*, les réponses redirigées sont traitées comme si l'hôte cible était disponible.

Voir aussi le tableau des valeurs par défaut.

icmppingsec[<target>,<packets>,<interval>,<size>,<timeout>,<mode>,<options>]

 Le temps de réponse du ping ICMP (en secondes).
 Valeur de retour : *Float*.

Paramètres :

- **target** - l'adresse IP ou le nom DNS de l'hôte ;
- **packets** - le nombre de paquets ;
- **interval** - le temps entre les paquets successifs, en millisecondes ;
- **size** - la taille du paquet en octets ;
- **timeout** - le délai d'expiration en millisecondes ;
- **mode** - valeurs possibles : *min*, *max* ou *avg* (par défaut) ;
- **options** - utilisé pour autoriser la redirection : si vide (valeur par défaut), les réponses redirigées sont traitées comme si l'hôte cible était indisponible ; si défini sur *allow_redirect*, les réponses redirigées sont traitées comme si l'hôte cible était disponible.

Commentaires :

- Les paquets perdus ou expirés ne sont pas utilisés dans le calcul ;
- Si l'hôte n'est pas disponible (délai d'expiration atteint), l'élément renverra 0 ;
- Si la valeur de retour est inférieure à 0.0001 seconde, la valeur sera définie sur 0.0001 seconde ;
- Voir également le tableau des valeurs par défaut.

net.tcp.service[service,<ip>,<port>]

 Vérifie si un service est en cours d'exécution et accepte les connexions TCP.
 Valeur de retour : *0* - le service est indisponible ; *1* - le service est en cours d'exécution.

Paramètres :

- **service** - valeurs possibles : *ssh*, *ldap*, *smtp*, *ftp*, *http*, *pop*, *nntp*, *imap*, *tcp*, *https*, *telnet* (voir les détails) ;
- **ip** - l'adresse IP ou le nom DNS (par défaut, l'IP/DNS de l'hôte est utilisé) ;
- **port** - le numéro de port (par défaut, le numéro de port standard du service est utilisé).

Commentaires :

- Notez qu'avec le service *tcp*, l'indication du port est obligatoire ;
- Ces vérifications peuvent entraîner des messages supplémentaires dans les fichiers journaux des démons système (les sessions SMTP et SSH étant généralement journalisées) ;
- La vérification des protocoles chiffrés (comme IMAP sur le port 993 ou POP sur le port 995) n'est actuellement pas prise en charge. Comme solution de contournement, veuillez utiliser `net.tcp.service[tcp,<ip>,port]` pour ce type de vérifications.

Exemple :

```
net.tcp.service[ftp,,45] #Cet élément peut être utilisé pour tester la disponibilité du serveur FTP sur le
```

Attention:
Si SELinux s'exécute en mode enforced, les contrôles simples TCP/UDP personnalisés peuvent être bloqués par la politique. Pour vérifier et autoriser la nouvelle connexion sortante, examinez les refus d'audit : avec `grep denied /var/log/audit/audit.log`

net.tcp.service.perf[service,<ip>,<port>]

 Vérifie les performances d'un service TCP.
 Valeur de retour : *Float* : *0.000000* - le service est indisponible ; *secondes* - le nombre de secondes passées lors de la connexion au service.

Paramètres :

- **service** - valeurs possibles : *ssh*, *ldap*, *smtp*, *ftp*, *http*, *pop*, *nntp*, *imap*, *tcp*, *https*, *telnet* (voir les détails) ;
- **ip** - l'adresse IP ou le nom DNS (par défaut, l'IP/DNS de l'hôte est utilisé) ;
- **port** - le numéro de port (par défaut, le numéro de port standard du service est utilisé).

Commentaires :

- Notez qu'avec le service *tcp*, l'indication du port est obligatoire ;
- La vérification des protocoles chiffrés (comme IMAP sur le port 993 ou POP sur le port 995) n'est actuellement pas prise en charge. Comme solution de contournement, veuillez utiliser `net.tcp.service[tcp,<ip>,port]` pour ce type de vérifications.

Exemple :

```
net.tcp.service.perf[ssh] #Cet élément peut être utilisé pour tester la vitesse de la réponse initiale du
```

net.udp.service[service,<ip>,<port>]

 Vérifie si un service est en cours d'exécution et répond aux requêtes UDP.
 Valeur de retour : *0* - le service est indisponible ; *1* - le service est en cours d'exécution.

Paramètres :

- **service** - valeurs possibles : *ntp* (voir détails) ;
- **ip** - l'adresse IP ou le nom DNS (par défaut, l'IP/DNS de l'hôte est utilisé) ;
- **port** - le numéro de port (par défaut, le numéro de port standard du service est utilisé).

Exemple :

```
net.udp.service[ntp,,45] #Cet élément peut être utilisé pour tester la disponibilité du service NTP sur le
```

net.udp.service.perf[service,<ip>,<port>]

 Vérifie les performances d'un service UDP.
 Valeur de retour : *Float* : *0.000000* - le service est indisponible ; *seconds* - le nombre de secondes passées à attendre une réponse du service.

Paramètres :

- **service** - valeurs possibles : *ntp* (voir les détails) ;
- **ip** - l'adresse IP ou le nom DNS (par défaut, l'IP/DNS de l'hôte est utilisé) ;
- **port** - le numéro de port (par défaut, le numéro de port standard du service est utilisé).

Exemple :

```
net.udp.service.perf[ntp] #Cet élément peut être utilisé pour tester le temps de réponse du service NTP.
```

Attention:
Pour la prise en charge de SourceIP dans les vérifications LDAP simples (par exemple, `net.tcp.service[ldap]`), la version 2.6.1 ou ultérieure d'OpenLDAP est requise.

Traitement du délai d'expiration

Les délais d'expiration flexibles des éléments, bien qu'ils soient pris en charge pour les vérifications simples, ne s'appliquent pas aux éléments `icmpping*` et VMware. Voir Prise en charge des délais d'expiration flexibles.

Pings ICMP

Zabbix utilise un utilitaire externe **fping** pour traiter les pings ICMP (**icmpping**, **icmppingloss**, **icmppingretry**, **icmppingsec**).

Installation

fping n'est pas inclus avec Zabbix et doit être installé séparément :

- Diverses plateformes basées sur Unix proposent le paquet fping dans leurs dépôts par défaut, mais il n'est pas préinstallé. Dans ce cas, vous pouvez utiliser le gestionnaire de paquets pour installer fping.
- Zabbix fournit des paquets fping pour RHEL et ses dérivés. Veuillez noter que ces paquets sont fournis sans support officiel.
- fping peut également être compilé à partir des sources.

Configuration

Spécifiez l'emplacement de fping dans le paramètre *FpingLocation* du fichier de configuration du serveur/proxy Zabbix (ou le paramètre *Fping6Location* pour utiliser des adresses IPv6).

fping doit être exécutable par l'utilisateur sous lequel le serveur/proxy Zabbix s'exécute, et cet utilisateur doit disposer de droits suffisants.

Voir aussi : Problèmes connus pour le traitement des vérifications simples avec les versions de fping antérieures à 3.10.

Valeurs par défaut

Valeurs par défaut, limites et description des valeurs des paramètres de vérification ICMP :

Paramètre	Unité	Description	Indicateur de fping	Valeurs par défaut définies par		Limites autorisées par Zabbix	
				fping	**Zabbix**	**min**	**max**
packets	nombre	Nombre de paquets de requête envoyés à une cible	-C		3	1	10000
interval	millisecondes	Temps d'attente entre les paquets successifs envoyés à une cible individuelle	-p	1000		20	illimité

Paramètre	Unité	Description	Indicateur de fping	Valeurs par défaut définies par		Limites autorisées par Zabbix	
size	octets	Taille du paquet en octets 56 octets sur x86, 68 octets sur x86_64	-b	56 ou 68		24	65507
timeout	millisecondes	**fping v3.x** - délai d'attente après l'envoi du dernier paquet, affecté par l'indicateur `-C` **fping v4.x** - délai d'attente individuel pour chaque paquet	-t	**fping v3.x** - 500 **fping v4.x** et versions ultérieures - hérité de l'indicateur `-p`, mais ne dépassant pas 2000		50	illimité
retries	nombre	Nombre de tentatives d'envoi d'une requête ping à une cible, sans compter la première tentative	-r	3	1	0	illimité
backoff factor	nombre	Facteur par lequel le temps d'attente est multiplié à chaque requête successive	-B	1.5	1.0	1.0	5.0

Les valeurs par défaut peuvent différer légèrement selon la plateforme et la version.

En outre, Zabbix utilise les options fping `-i interval ms` (à ne pas confondre avec le paramètre d'élément `interval` mentionné dans le tableau ci-dessus, qui correspond à l'option fping `-p`) et `-S source IP address` (ou `-I` dans les anciennes versions de fping). Ces options sont détectées automatiquement en exécutant des vérifications avec différentes combinaisons d'options. Zabbix tente de détecter la valeur minimale en millisecondes que fping autorise pour l'option `-i` en essayant trois valeurs : 0, 1 et 10. La première valeur qui fonctionne est ensuite utilisée pour les vérifications ICMP suivantes. Ce processus est effectué individuellement par chaque processus ICMP pinger.

Les options fping détectées automatiquement sont invalidées toutes les heures et détectées à nouveau lors de la tentative suivante d'effectuer une vérification ICMP. Définissez DebugLevel>=4 afin d'afficher les détails de ce processus dans le fichier journal du serveur ou du proxy.

Zabbix écrit les adresses IP à vérifier avec l'une des clés `icmpping*` dans un fichier temporaire, qui est ensuite transmis à fping. Si les éléments possèdent des paramètres de clé différents, seuls ceux ayant des paramètres de clé identiques sont écrits dans un même fichier. Toutes les adresses IP écrites dans le même fichier sont vérifiées en parallèle par fping ; le processus ICMP pinger de Zabbix consacre donc une durée fixe à cette opération, quel que soit le nombre d'adresses IP contenues dans le fichier.

Clés d'élément de surveillance VMware

La liste des clés d'élément de la surveillance VMware a été déplacée vers la section Surveillance VMware.

agent SNMP

Vue d'ensemble

Vous pouvez souhaiter utiliser la supervision SNMP sur des périphériques tels que des imprimantes, des commutateurs réseau, des routeurs ou des onduleurs, qui sont généralement compatibles SNMP et sur lesquels il serait peu pratique de tenter d'installer des systèmes d'exploitation complets et des agents Zabbix.

Pour pouvoir récupérer les données fournies par les agents SNMP sur ces périphériques, le serveur Zabbix doit être configuré initialement avec la prise en charge SNMP en spécifiant l'option `--with-net-snmp`. Il est également recommandé d'installer les

fichiers MIB afin de garantir que les valeurs des éléments s'affichent dans le bon format. Sans les fichiers MIB, des problèmes de formatage, tels que l'affichage des valeurs en HEX au lieu de UTF-8 ou inversement, peuvent se produire.

Les vérifications SNMP sont effectuées uniquement via le protocole UDP.

Les démons du serveur Zabbix et du proxy consignent des lignes similaires à la suivante s'ils reçoivent une réponse SNMP incorrecte :

```
SNMP response from host "gateway" does not contain all of the requested variable bindings
```

Bien qu'elles ne couvrent pas tous les cas problématiques, elles sont utiles pour identifier les périphériques SNMP individuels pour lesquels les requêtes combinées doivent être désactivées.

Le serveur/proxy Zabbix réessaiera jusqu'à 5 fois pour les éléments SNMP `walk` et `get`. Le mécanisme de nouvelle tentative ne s'applique pas aux échecs de résolution DNS.

Pour les vérifications SNMP héritées (numéro ou chaîne OID unique), le serveur/proxy Zabbix réessaiera au moins une fois après une tentative de requête infructueuse : soit via le mécanisme de nouvelle tentative de la bibliothèque SNMP, soit via le mécanisme interne de traitement combiné.

Warning:
Si vous surveillez des périphériques SNMPv3, assurez-vous que msgAuthoritativeEngineID (également appelé snmpEngineID ou "Engine ID") n'est jamais partagé par deux périphériques. Selon la RFC 2571 (section 3.1.1.1), il doit être unique pour chaque périphérique.

Warning:
La RFC3414 exige que les périphériques SNMPv3 conservent leurs engineBoots de manière persistante. Certains périphériques ne le font pas, ce qui entraîne le rejet de leurs messages SNMP comme obsolètes après un redémarrage. Dans une telle situation, le cache SNMP doit être vidé manuellement sur un serveur/proxy (en utilisant -R snmp_cache_reload) ou le serveur/proxy doit être redémarré.

Note:
Zabbix met en cache les correspondances SNMPv3 EngineID → IP et réutilise les EngineID mis en cache pour les vérifications suivantes au lieu d'envoyer une sonde à chaque fois, ce qui réduit le trafic réseau. Si un EngineID ne peut pas être réutilisé, une nouvelle tentative avec une sonde sera effectuée pour découvrir le nouvel EngineID.

Configuration de la supervision SNMP

Pour commencer à superviser un appareil via SNMP, les étapes suivantes doivent être effectuées :

Étape 1

Déterminez la chaîne SNMP (ou l'OID) de l'élément que vous souhaitez surveiller.

Pour obtenir une liste de chaînes SNMP, utilisez la commande `snmpwalk` (faisant partie du logiciel net-snmp que vous devriez avoir installé dans le cadre de l'installation de Zabbix) ou un outil équivalent :

```
snmpwalk -v 2c -c public <host IP> .
```

Comme 2c correspond ici à la version SNMP, vous pouvez également le remplacer par 1 pour indiquer la version 1 de SNMP sur le périphérique.

Cela devrait vous fournir une liste de chaînes SNMP et leur dernière valeur. Si ce n'est pas le cas, il est possible que la "community" SNMP soit différente de la valeur standard `public`; dans ce cas, vous devrez la déterminer.

Vous pouvez ensuite parcourir la liste jusqu'à trouver la chaîne que vous souhaitez surveiller, par exemple : si vous vouliez surveiller les octets entrants sur votre commutateur sur le port 3, vous utiliseriez la chaîne `IF-MIB::ifHCInOctets.3` à partir de cette ligne :

```
IF-MIB::ifHCInOctets.3 = Counter64: 3409739121
```

Vous pouvez maintenant utiliser la commande `snmpget` pour déterminer l'OID numérique de `IF-MIB::ifHCInOctets.3` :

```
snmpget -v 2c -c public -On <host IP> IF-MIB::ifHCInOctets.3
```

Notez que le dernier nombre dans la chaîne correspond au numéro de port que vous cherchez à surveiller. Voir aussi : Index dynamiques.

Cela devrait vous donner quelque chose comme ce qui suit :

```
.1.3.6.1.2.1.31.1.1.1.6.3 = Counter64: 3472126941
```

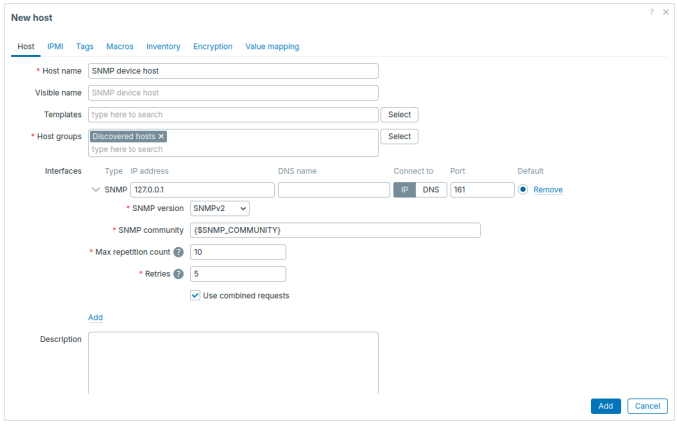
Là encore, le dernier nombre dans l'OID correspond au numéro de port.

Note:
Certains des OID SNMP les plus utilisés sont traduits automatiquement en représentation numérique par Zabbix.

Dans l'exemple précédent, le type de valeur est "Counter64", ce qui correspond en interne au type ASN_COUNTER64. La liste complète des types pris en charge est ASN_COUNTER, ASN_COUNTER64, ASN_UINTEGER, ASN_UNSIGNED64, ASN_INTEGER, ASN_INTEGER64, ASN_FLOAT, ASN_DOUBLE, ASN_TIMETICKS, ASN_GAUGE, ASN_IPADDRESS, ASN_OCTET_STR et ASN_OBJECT_ID. Ces types correspondent approximativement à "Counter32", "Counter64", "UInteger32", "INTEGER", "Float", "Double", "Timeticks", "Gauge32", "IpAddress", "OCTET STRING", "OBJECT IDENTIFIER" dans la sortie de **snmpget**, mais peuvent aussi être affichés comme "STRING", "Hex-STRING", "OID" et autres, selon la présence d'un indice d'affichage.

Étape 2

Créez un hôte correspondant à un périphérique.



Ajoutez une interface SNMP pour l'hôte :

1. Saisissez l'adresse IP/le nom DNS et le numéro de port.
2. Sélectionnez la *version SNMP* dans la liste déroulante.
3. Ajoutez les identifiants de l'interface en fonction de la version SNMP sélectionnée :
   - SNMPv1 et v2 nécessitent uniquement la communauté (généralement 'public').
   - SNMPv3 nécessite des options plus spécifiques ; voir SNMPv3.
4. Indiquez la valeur maximale de répétition (par défaut : 10) pour les requêtes SNMP bulk natives (GetBulkRequest-PDU) ; uniquement pour les éléments `discovery[]` et `walk[]` en SNMPv2 et v3. Notez qu'une valeur trop élevée peut entraîner un délai d'attente de la vérification de l'agent SNMP.
5. Cochez la case *Utiliser des requêtes combinées* pour autoriser le traitement combiné des requêtes SNMP (sans lien avec les requêtes SNMP bulk natives "walk" et "get").

Vous pouvez utiliser l'un des modèles SNMP fournis, qui ajoutera automatiquement un ensemble d'éléments. Avant d'utiliser un modèle, vérifiez qu'il est compatible avec l'hôte.

Cliquez sur *Ajouter* pour enregistrer l'hôte.

SNMPv3

Les paramètres suivants sont requis pour SNMPv3 :

- *Nom de contexte* - Saisissez le nom de contexte pour identifier l'élément sur le sous-réseau SNMP.
Les macros utilisateur sont résolues dans ce champ.
- *Nom de sécurité* - Saisissez le nom de sécurité.
Les macros utilisateur sont résolues dans ce champ.
- *Niveau de sécurité* - Sélectionnez le niveau de sécurité :
  - *noAuthNoPriv* - aucun protocole d'authentification ni de confidentialité n'est utilisé ;
  - *AuthNoPriv* - le protocole d'authentification est utilisé, le protocole de confidentialité ne l'est pas ;
  - *AuthPriv* - les protocoles d'authentification et de confidentialité sont tous deux utilisés.
- *Protocole d'authentification* - Sélectionnez le protocole d'authentification : *MD5*, *SHA1* ; avec net-snmp 5.8 et versions ultérieures *SHA224*, *SHA256*, *SHA384* ou *SHA512*.
- *Phrase secrète d'authentification* - Saisissez la phrase secrète d'authentification.
Les macros utilisateur sont résolues dans ce champ.
- *Protocole de confidentialité* - Sélectionnez le protocole de confidentialité : *DES*, *AES128*, *AES192*, *AES256*, *AES192C* (Cisco) ou *AES256C* (Cisco).
Consultez les notes sur la prise en charge du protocole de confidentialité.

- *Phrase secrète de confidentialité* - Saisissez la phrase secrète de confidentialité.
Les macros utilisateur sont résolues dans ce champ.

En cas d'identifiants SNMPv3 incorrects (nom de sécurité, protocole/phrase secrète d'authentification, protocole de confidentialité) :

- Zabbix reçoit une erreur ERROR de net-snmp, sauf en cas de *Phrase secrète de confidentialité* incorrecte, auquel cas Zabbix reçoit une erreur TIMEOUT de net-snmp.
- La disponibilité de l'interface SNMP passera au rouge (indisponible).

Note:
Les modifications apportées au *Protocole d'authentification*, à la *Phrase secrète d'authentification*, au *Protocole de confidentialité* ou à la *Phrase secrète de confidentialité*, sans modification du *Nom de sécurité*, sont normalement appliquées automatiquement lorsque l'interface SNMPv3 correspondante est mise à jour dans Zabbix. Si le *Nom de sécurité* est également modifié, tous les paramètres seront mis à jour immédiatement.

Prise en charge du protocole de confidentialité

Selon votre système d'exploitation et la configuration de net-snmp, certains protocoles de confidentialité peuvent ne pas être disponibles :

- Sur certains systèmes d'exploitation plus récents (par exemple, RHEL9), la prise en charge de DES est supprimée pour le package `net-snmp`.
- Les protocoles de chiffrement AES192 et plus puissants ne sont pas pris en charge nativement sur les systèmes d'exploitation antérieurs à RHEL 8, CentOS 8, Oracle Linux 8, Debian 12, Ubuntu LTS 22.04, openSUSE Leap 15.5.

Pour vérifier si la bibliothèque net-snmp prend en charge AES192+, utilisez l'une des options suivantes :

1. `net-snmp-config` :

```
net-snmp-config --configure-options
```

Si la sortie contient `--enable-blumenthal-aes`, AES192+ est pris en charge.

Notez que `net-snmp-config` fait partie du package de développement pour SNMP (`libsnmp-dev` pour Debian/Ubuntu, `net-snmp-devel` pour CentOS/RHEL/OL/SUSE) et peut ne pas être installé par défaut.

2. `snmpget` :

```
snmpget -v 3 -x AES-256
```

Si la sortie contient `Invalid privacy protocol specified after -3x flag: AES-256`, AES192+ n'est pas pris en charge. Si la sortie contient `No hostname specified.`, AES192+ n'est pas pris en charge.

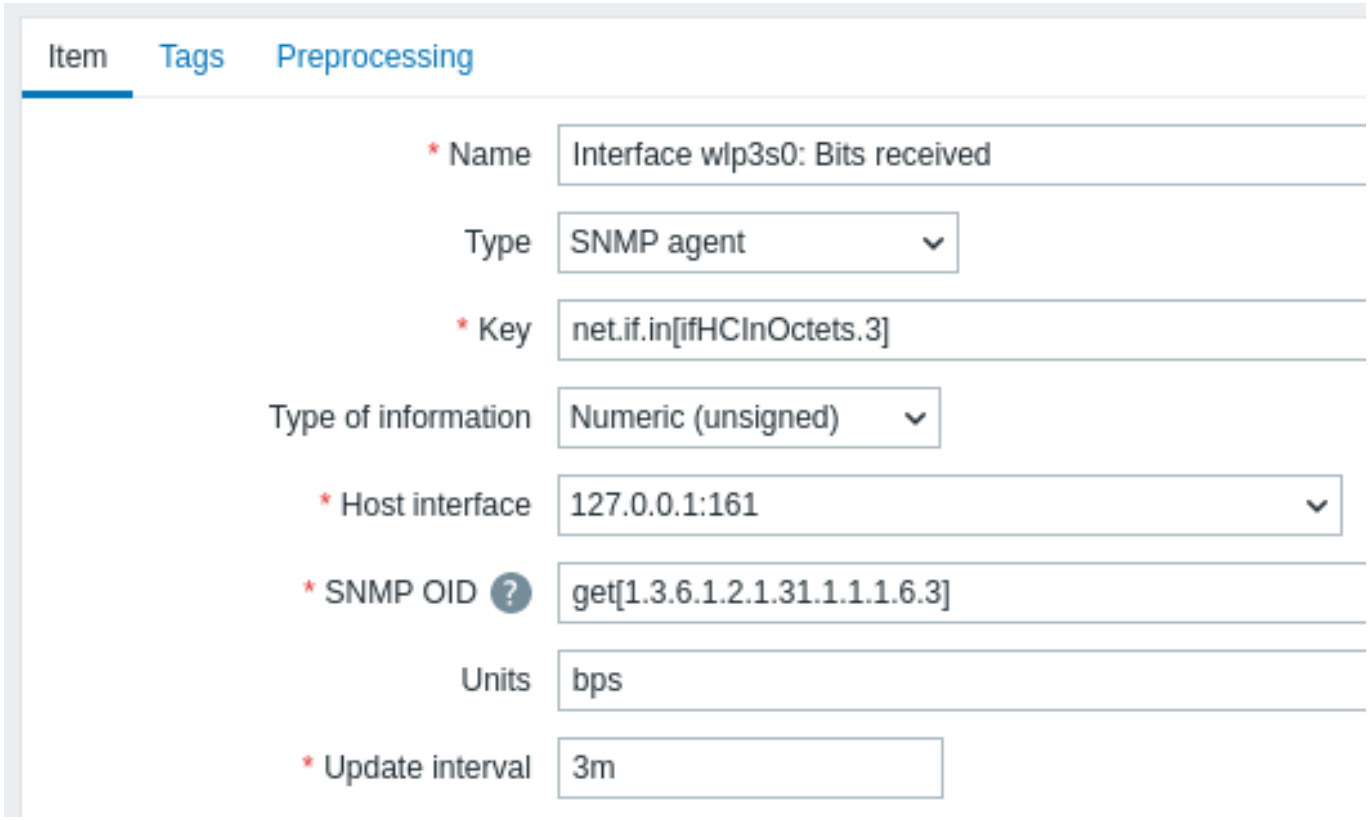
Si votre bibliothèque net-snmp ne prend pas en charge les protocoles AES192 et supérieurs, recompilez net-snmp avec l'option `--enable-blumenthal-aes`, puis recompilez le serveur Zabbix en spécifiant l'option `--with-net-snmp=/home/user/yourcustomnets`

Étape 3

Créez un élément pour la supervision.

Retournez maintenant dans Zabbix et cliquez sur *Éléments* pour l'hôte SNMP créé précédemment. Selon que vous avez utilisé un modèle ou non lors de la création de votre hôte, vous disposerez soit d'une liste d'éléments SNMP associés à votre hôte, soit d'une liste vide. Nous partirons du principe que vous allez créer vous-même l'élément à l'aide des informations que vous venez de recueillir avec snmpwalk et snmpget. Cliquez donc sur *Créer un élément*.

Renseignez les paramètres obligatoires dans le formulaire du nouvel élément :



Paramètre	Description
Nom	Saisissez le nom de l'élément.
Type	Sélectionnez **Agent SNMP** ici.
Clé	Saisissez une clé explicite.
Interface de l'hôte	Veillez à sélectionner l'interface SNMP, par exemple celle de votre commutateur/routeur.

Paramètre	Description
OID SNMP	Utilisez l'un des formats pris en charge pour saisir la ou les valeurs OID : **walk[OID1,OID2,...]** - récupère un sous-arbre de valeurs. Par exemple : `walk[1.3.6.1.2.1.2.2.1.2,1.3.6.1.2.1.2.2.1.3]`. Cette option utilise les requêtes SNMP bulk natives (GetBulkRequest-PDU) de manière asynchrone. Les paramètres de délai d'expiration de cet élément peuvent être définis dans le formulaire de configuration de l'élément. Il est recommandé de définir une valeur de délai d'expiration faible afin d'éviter de longs délais si l'équipement est inaccessible, car jusqu'à 5 nouvelles tentatives seront effectuées si les précédentes expirent ou échouent (par exemple, un délai d'expiration de 3 secondes peut entraîner une attente de 15 secondes). Vous pouvez utiliser cet élément comme élément maître, avec des éléments dépendants qui extraient les données de l'élément maître à l'aide du prétraitement. Il est possible de spécifier plusieurs OID dans un seul parcours snmp, par exemple `walk[OID1,OID2,...]`, afin de traiter les OID un par un de manière asynchrone. Si la requête bulk ne renvoie aucun résultat, une tentative de récupération d'un enregistrement unique sans requête bulk est effectuée. Les noms MIB sont pris en charge comme paramètres ; ainsi, `walk[1.3.6.1.2.1.2.2.1.2]` et `walk[ifDescr]` renverront le même résultat. Si plusieurs OID/MIB sont spécifiés, par exemple `walk[ifDescr,ifType,ifPhysAddress]`, le résultat est une liste concaténée. Les requêtes GetBulk sont utilisées avec les interfaces SNMPv2 et v3, et GetNext avec les interfaces SNMPv1 ; le nombre maximal de répétitions pour les requêtes bulk est configuré au niveau de l'interface. Le paramètre de répétitions maximales agit sur les requêtes bulk en déterminant le nombre maximal d'OID renvoyés dans une seule réponse bulk. Une valeur plus élevée produit des réponses bulk plus volumineuses et réduit le nombre de transmissions nécessaires. Cependant, certains équipements peuvent ne pas prendre en charge des valeurs très élevées, ce qui pourrait entraîner des problèmes. Cet élément renvoie la sortie de l'utilitaire snmpwalk avec les paramètres -Oe -Ot -On. Vous pouvez utiliser cet élément comme élément maître dans la découverte SNMP. **get[OID]** - récupère une *valeur unique* de manière asynchrone. Par exemple : `get[1.3.6.1.2.1.31.1.1.1.6.3]` Les paramètres de délai d'expiration de cet élément peuvent être définis dans le formulaire de configuration de l'élément. Il est recommandé de définir une valeur de délai d'expiration faible afin d'éviter de longs délais si l'équipement est inaccessible, car jusqu'à 5 nouvelles tentatives seront effectuées si les précédentes expirent ou échouent (par exemple, un délai d'expiration de 3 secondes peut entraîner une attente de 15 secondes). **OID** - (ancien format) saisissez un OID textuel ou numérique unique pour récupérer une valeur unique de manière synchrone, éventuellement combiné à d'autres valeurs. Par exemple : `1.3.6.1.2.1.31.1.1.1.6.3`. Avec cette option, le délai d'expiration de la vérification de l'élément sera égal à la valeur définie dans le fichier de configuration du serveur. Il est **recommandé** d'utiliser les éléments `walk[OID]` et `get[OID]` pour de meilleures performances. Tous les éléments `walk[OID]` et `get[OID]` sont exécutés de manière asynchrone : il n'est pas nécessaire de recevoir la réponse à une requête avant de démarrer d'autres vérifications. La résolution DNS est également asynchrone. La simultanéité maximale des vérifications asynchrones est de 1000 (définie par MaxConcurrentChecksPerPoller). Le nombre de pollers SNMP asynchrones est défini par le paramètre StartSNMPPollers. Notez que pour les statistiques de trafic réseau renvoyées par l'une des méthodes, une étape *Variation par seconde* doit être ajoutée dans l'onglet *Prétraitement* ; sinon, vous obtiendrez la valeur cumulée de l'équipement SNMP au lieu de la variation la plus récente.

Tous les champs obligatoires sont signalés par un astérisque rouge.

Enregistrez maintenant l'élément, puis accédez à *Supervision* > *Dernières données* pour consulter vos données SNMP.

Exemple 1

Exemple général :

Paramètre	Description
OID	1.2.3.45.6.7.8.0 (ou .1.2.3.45.6.7.8.0)
Clé	<Chaîne unique à utiliser comme référence pour les déclencheurs> Par exemple, ”my_param”.

Notez que l'OID peut être fourni sous forme numérique ou textuelle. Cependant, dans certains cas, l'OID textuel doit être converti en représentation numérique. L'utilitaire snmpget peut être utilisé à cette fin :

```
snmpget -On localhost public enterprises.ucdavis.memory.memTotalSwap.0
```

Exemple 2

Surveillance du temps de fonctionnement :

Paramètre	Description
OID	MIB::sysUpTime.0
Clé	router.uptime
Type de valeur	Float
Unités	uptime
Étape de prétraitement : Multiplicateur personnalisé	0.01

Requêtes SNMP bulk natives

L'élément **walk[OID1,OID2,...]** permet d'utiliser la fonctionnalité SNMP native pour les requêtes bulk (PDU GetBulkRequest), disponible dans les versions 2 et 3 de SNMP.

Une requête GetBulk dans SNMP exécute plusieurs requêtes GetNext et renvoie le résultat dans une seule réponse. Cela peut être utilisé aussi bien pour les éléments SNMP classiques que pour la découverte SNMP, afin de réduire le nombre d'allers-retours réseau.

L'élément SNMP **walk[OID1,OID2,...]** peut être utilisé comme élément maître pour collecter les données en une seule requête, avec des éléments dépendants qui analysent la réponse selon les besoins à l'aide du prétraitement.

Notez que l'utilisation de requêtes SNMP bulk natives n'est pas liée à l'option de combinaison des requêtes SNMP, qui correspond à la méthode propre à Zabbix pour combiner plusieurs requêtes SNMP (voir la section suivante).

Jusqu'à cinq nouvelles tentatives seront effectuées pour les éléments SNMP bulk afin d'éviter un échec en cas de perte de l'un des paquets. Le délai d'expiration des éléments SNMP utilisant `get` et `walk` (défini dans le formulaire de configuration de l'élément) s'applique à l'ensemble de la session. Le délai d'expiration est appliqué indépendamment du fait que les données soient récupérées complètement ou non ; si les données sont reçues partiellement (par exemple, si les données ne sont collectées avec succès que pour un seul des plusieurs OID), l'élément devient non pris en charge avec le message « Only partial data received ». Si le délai d'expiration est atteint, une nouvelle tentative est effectuée, le délai d'expiration est réinitialisé et la dernière requête est renvoyée, ce qui permet à la session de continuer à partir de la dernière requête si un seul paquet est perdu ou arrive trop tard. Envisagez de définir une valeur de délai d'expiration faible afin d'éviter de longs délais lorsque le périphérique est inaccessible, car jusqu'à 5 nouvelles tentatives seront effectuées si les précédentes expirent ou échouent (par exemple, un délai d'expiration de 3 secondes peut entraîner un temps d'attente de 15 secondes).

Fonctionnement interne du traitement combiné

Le serveur Zabbix et le proxy peuvent interroger les périphériques SNMP pour obtenir plusieurs valeurs dans une seule requête. Cela concerne plusieurs types d'éléments SNMP :

- éléments SNMP standard
- éléments SNMP avec des index dynamiques
- règles de découverte de bas niveau SNMP

Tous les éléments SNMP d'une même interface et ayant des paramètres identiques sont planifiés pour être interrogés au même moment. Les deux premiers types d'éléments sont pris en charge par les pollers par lots de 128 éléments au maximum, tandis que les règles de découverte de bas niveau sont traitées individuellement, comme auparavant.

À un niveau inférieur, deux types d'opérations sont effectués pour interroger les valeurs : l'obtention de plusieurs objets spécifiés et le parcours d'une arborescence d'OID.

Pour l'« obtention », un GetRequest-PDU est utilisé avec 128 liaisons de variables au maximum. Pour le « parcours », un GetNextRequest-PDU est utilisé avec SNMPv1, et un GetBulkRequest avec un champ « max-repetitions » de 128 au maximum est utilisé avec SNMPv2 et SNMPv3.

Ainsi, les avantages du traitement combiné pour chaque type d'élément SNMP sont les suivants :

- les éléments SNMP standard bénéficient des améliorations liées à l'« obtention » ;
- les éléments SNMP avec des index dynamiques bénéficient des améliorations liées à l'« obtention » et au « parcours » : l'« obtention » est utilisée pour vérifier les index et le « parcours » pour créer le cache ;
- les règles de découverte de bas niveau SNMP bénéficient des améliorations liées au « parcours ».

Cependant, un problème technique se pose : tous les périphériques ne sont pas capables de renvoyer 128 valeurs par requête. Certains renvoient toujours une réponse correcte, tandis que d'autres renvoient soit une erreur « tooBig(1) », soit aucune réponse lorsque la réponse potentielle dépasse une certaine limite.

Afin de déterminer le nombre optimal d'objets à interroger pour un périphérique donné, Zabbix utilise la stratégie suivante. Il commence prudemment par interroger 1 valeur par requête. Si l'opération réussit, il interroge 2 valeurs par requête. Si elle réussit à nouveau, il interroge 3 valeurs par requête et continue de la même manière en multipliant le nombre d'objets interrogés par 1,5, ce qui donne la séquence de tailles de requête suivante : 1, 2, 3, 4, 6, 9, 13, 19, 28, 42, 63, 94, 128.

Cependant, lorsqu'un périphérique refuse de renvoyer une réponse correcte (par exemple, pour 42 variables), Zabbix effectue deux opérations.

Tout d'abord, pour le lot d'éléments actuel, il divise par deux le nombre d'objets dans une seule requête et interroge 21 variables. Si le périphérique est disponible, la requête devrait fonctionner dans la grande majorité des cas, car il a été établi que 28 variables fonctionnaient et que 21 est nettement inférieur à cette valeur. Cependant, si cela échoue également, Zabbix revient à l'interrogation des valeurs une par une. Si cela échoue encore à ce stade, le périphérique ne répond définitivement pas et la taille de la requête n'est pas en cause.

La deuxième opération effectuée par Zabbix pour les lots d'éléments suivants consiste à commencer par le dernier nombre de variables ayant réussi (28 dans notre exemple), puis à augmenter la taille des requêtes de 1 jusqu'à atteindre la limite. Par exemple, en supposant que la taille de réponse maximale soit de 32 variables, les requêtes suivantes auront les tailles 29, 30, 31, 32 et 33. La dernière requête échouera et Zabbix n'enverra plus jamais de requête de taille 33. À partir de ce moment, Zabbix interrogera au maximum 32 variables pour ce périphérique.

Si les requêtes volumineuses échouent avec ce nombre de variables, cela peut avoir deux causes. Les critères exacts utilisés par un périphérique pour limiter la taille des réponses ne peuvent pas être connus, mais nous essayons de les approximer à l'aide du nombre de variables. La première possibilité est que ce nombre de variables soit proche de la limite réelle de taille de réponse du périphérique dans le cas général : parfois, la réponse est inférieure à la limite et parfois, elle lui est supérieure. La deuxième possibilité est qu'un paquet UDP ait simplement été perdu dans l'une ou l'autre direction. Pour ces raisons, si Zabbix reçoit une réponse indiquant l'échec d'une requête, il réduit le nombre maximal de variables à essayer afin de se rapprocher davantage de la plage confortable du périphérique, mais seulement deux fois au maximum.

Dans l'exemple ci-dessus, si une requête contenant 32 variables échoue, Zabbix réduit le nombre à 31. Si cette requête échoue également, Zabbix réduit le nombre à 30. Cependant, Zabbix ne réduira pas le nombre en dessous de 30, car il considérera que les échecs suivants sont dus à la perte de paquets UDP plutôt qu'à la limite du périphérique.

Si, toutefois, un périphérique ne peut pas gérer correctement les requêtes combinées pour d'autres raisons et que l'heuristique décrite ci-dessus ne fonctionne pas, un paramètre *Use combined requests* est disponible pour chaque interface et permet de désactiver les requêtes combinées pour ce périphérique.

Si les requêtes combinées provoquent des réponses partielles ou malformées entraînant des calculs incorrects par seconde (delta) (par exemple, des pics apparents dans les compteurs d'interface), désactivez *Use combined requests* pour l'interface concernée afin de forcer des requêtes distinctes pour chaque élément ; cela permet souvent d'éviter les faux pics. Vous pouvez également envisager d'utiliser des éléments `get[]` ou `walk[]` asynchrones, qui sont exécutés de manière asynchrone et ne sont pas soumis au regroupement *Use combined requests* par interface ; ils peuvent être utilisés à la place des vérifications OID synchrones héritées afin d'éviter les problèmes liés aux requêtes combinées. Recherchez dans les journaux du serveur/proxy des entrées similaires à celle présentée dans la section Vue d'ensemble afin d'identifier les périphériques concernés.

De plus, si l'interface devient fréquemment indisponible, il peut être nécessaire d'augmenter le paramètre `UnavailableDelay` dans les fichiers de configuration du serveur Zabbix ou du proxy Zabbix afin de réduire la fréquence des requêtes. Les éléments peuvent devenir non pris en charge si des données partielles sont reçues pendant la découverte ou les parcours d'OID.

Indexes dynamiques

Aperçu

Bien que vous puissiez trouver le numéro d'index requis (par exemple, celui d'une interface réseau) parmi les OID SNMP, il arrive que vous ne puissiez pas vous fier entièrement au fait que ce numéro d'index reste toujours identique.

Les numéros d'index peuvent être dynamiques : ils peuvent changer au fil du temps et, par conséquent, votre élément peut cesser de fonctionner.

Pour éviter ce scénario, il est possible de définir un OID qui prend en compte la possibilité qu'un numéro d'index change.

Par exemple, si vous devez récupérer la valeur d'index à ajouter à **ifInOctets** qui correspond à l'interface **GigabitEthernet0/1** sur un appareil Cisco, utilisez l'OID suivant :

```
ifInOctets["index","ifDescr","GigabitEthernet0/1"]
```

La syntaxe

Une syntaxe spéciale est utilisée pour l'OID :

<OID des données>["index","<OID de base de l'index>","<chaîne à rechercher>"]

Paramètre	Description
OID des données	OID principal à utiliser pour la récupération des données dans l'élément.
index	Méthode de traitement. Actuellement, une seule méthode est prise en charge : **index** - rechercher l'index et l'ajouter à l'OID des données
OID de base de l'index	Cet OID sera recherché pour obtenir la valeur d'index correspondant à la chaîne.
chaîne à rechercher	La chaîne à utiliser pour une correspondance exacte avec une valeur lors de la recherche. Sensible à la casse.

Exemple

Obtention de l'utilisation mémoire du processus *apache*.

Si vous utilisez cette syntaxe OID :

```
HOST-RESOURCES-MIB::hrSWRunPerfMem["index","HOST-RESOURCES-MIB::hrSWRunPath", "/usr/sbin/apache2"]
```

le numéro d'index sera recherché ici :

```
...
HOST-RESOURCES-MIB::hrSWRunPath.5376 = STRING: "/sbin/getty"
HOST-RESOURCES-MIB::hrSWRunPath.5377 = STRING: "/sbin/getty"
HOST-RESOURCES-MIB::hrSWRunPath.5388 = STRING: "/usr/sbin/apache2"
HOST-RESOURCES-MIB::hrSWRunPath.5389 = STRING: "/sbin/sshd"
...
```

Nous avons maintenant l'index, 5388. L'index sera ajouté à l'OID de données afin de recevoir la valeur qui nous intéresse :

```
HOST-RESOURCES-MIB::hrSWRunPerfMem.5388 = INTEGER: 31468 KBytes
```

Mise en cache de la recherche d'index

Lorsqu'un élément d'index dynamique est demandé, Zabbix récupère et met en cache l'ensemble de la table SNMP sous l'OID de base pour l'index, même si une correspondance pouvait être trouvée plus tôt. Cela est fait au cas où un autre élément ferait référence plus tard au même OID de base - Zabbix rechercherait alors l'index dans le cache, au lieu d'interroger à nouveau l'hôte supervisé. Notez que chaque processus poller utilise un cache distinct.

Dans toutes les opérations ultérieures de récupération de valeur, seul l'index trouvé est vérifié. S'il n'a pas changé, la valeur est demandée. S'il a changé, le cache est reconstruit - chaque poller qui rencontre un index modifié parcourt à nouveau la table SNMP d'index.

OID spéciaux

Certains des OID SNMP les plus utilisés sont automatiquement traduits en une représentation numérique par Zabbix. Par exemple, **ifIndex** est traduit en **1.3.6.1.2.1.2.2.1.1**, **ifIndex.0** est traduit en **1.3.6.1.2.1.2.2.1.1.0**.

Le tableau contient la liste des OID spéciaux.

OID spécial	Identifiant	Description
ifIndex	1.3.6.1.2.1.2.2.1.1	Une valeur unique pour chaque interface.

OID spécial	Identifiant	Description
ifDescr	1.3.6.1.2.1.2.2.1.2	Une chaîne de texte contenant des informations sur l'interface. Cette chaîne doit inclure le nom du fabricant, le nom du produit et la version de l'interface matérielle.
ifType	1.3.6.1.2.1.2.2.1.3	Le type d'interface, distingué selon le ou les protocoles physiques/de liaison immédiatement "en dessous" de la couche réseau dans la pile de protocoles.
ifMtu	1.3.6.1.2.1.2.2.1.4	La taille du plus grand datagramme pouvant être envoyé/reçu sur l'interface, exprimée en octets.
ifSpeed	1.3.6.1.2.1.2.2.1.5	Une estimation de la bande passante actuelle de l'interface en bits par seconde.
ifPhysAddress	1.3.6.1.2.1.2.2.1.6	L'adresse de l'interface au niveau de la couche de protocole immédiatement 'en dessous' de la couche réseau dans la pile de protocoles.
ifAdminStatus	1.3.6.1.2.1.2.2.1.7	L'état administratif actuel de l'interface.
ifOperStatus	1.3.6.1.2.1.2.2.1.8	L'état opérationnel actuel de l'interface.
ifInOctets	1.3.6.1.2.1.2.2.1.10	Le nombre total d'octets reçus sur l'interface, y compris les caractères de tramage.
ifInUcastPkts	1.3.6.1.2.1.2.2.1.11	Le nombre de paquets unicast de sous-réseau livrés à un protocole de couche supérieure.
ifInNUcastPkts	1.3.6.1.2.1.2.2.1.12	Le nombre de paquets non unicast (c'est-à-dire broadcast ou multicast de sous-réseau) livrés à un protocole de couche supérieure.
ifInDiscards	1.3.6.1.2.1.2.2.1.13	Le nombre de paquets entrants qui ont été choisis pour être abandonnés, bien qu'aucune erreur n'ait été détectée, afin d'empêcher leur livraison à un protocole de couche supérieure. Une raison possible pour abandonner un tel paquet pourrait être de libérer de l'espace tampon.
ifInErrors	1.3.6.1.2.1.2.2.1.14	Le nombre de paquets entrants contenant des erreurs empêchant leur livraison à un protocole de couche supérieure.
ifInUnknownProtos	1.3.6.1.2.1.2.2.1.15	Le nombre de paquets reçus via l'interface qui ont été abandonnés en raison d'un protocole inconnu ou non pris en charge.
ifOutOctets	1.3.6.1.2.1.2.2.1.16	Le nombre total d'octets transmis hors de l'interface, y compris les caractères de tramage.
ifOutUcastPkts	1.3.6.1.2.1.2.2.1.17	Le nombre total de paquets que les protocoles de niveau supérieur ont demandé de transmettre, et qui n'étaient pas adressés à une adresse multicast ou broadcast à ce sous-niveau, y compris ceux qui ont été abandonnés ou non envoyés.
ifOutNUcastPkts	1.3.6.1.2.1.2.2.1.18	Le nombre total de paquets que les protocoles de niveau supérieur ont demandé de transmettre, et qui étaient adressés à une adresse multicast ou broadcast à ce sous-niveau, y compris ceux qui ont été abandonnés ou non envoyés.
ifOutDiscards	1.3.6.1.2.1.2.2.1.19	Le nombre de paquets sortants qui ont été choisis pour être abandonnés, bien qu'aucune erreur n'ait été détectée, afin d'empêcher leur transmission. Une raison possible pour abandonner un tel paquet pourrait être de libérer de l'espace tampon.
ifOutErrors	1.3.6.1.2.1.2.2.1.20	Le nombre de paquets sortants qui n'ont pas pu être transmis en raison d'erreurs.
ifOutQLen	1.3.6.1.2.1.2.2.1.21	La longueur de la file d'attente des paquets de sortie (en paquets).

Fichiers MIB

Introduction

MIB signifie Management Information Base (base d'informations de gestion). Les fichiers MIB permettent d'utiliser des représentations textuelles des OID (identifiants d'objet). Il est possible d'utiliser des OID bruts lors de la supervision de périphériques SNMP avec Zabbix, mais si vous êtes plus à l'aise avec une représentation textuelle, vous devez installer les fichiers MIB.

Par exemple,

`ifHCOutOctets`

est la représentation textuelle de l'OID

```
1.3.6.1.2.1.31.1.1.1.10
```

Installation des fichiers MIB

Sur les systèmes basés sur Debian :

```
apt install snmp-mibs-downloader
download-mibs
```

Sur les systèmes basés sur RedHat :

```
dnf install net-snmp-libs
```

Activation des fichiers MIB

Sur les systèmes basés sur RedHat, les fichiers MIB devraient être activés par défaut. Sur les systèmes basés sur Debian, vous devez modifier le fichier `/etc/snmp/snmp.conf` et mettre en commentaire la ligne `mibs :`

```
# Comme les paquets snmp sont fournis sans fichiers MIB pour des raisons de licence, le chargement
# des MIB est désactivé par défaut. Si vous avez ajouté les MIB, vous pouvez réactiver
# leur chargement en mettant en commentaire la ligne suivante.
mibs :
```

Test des fichiers MIB

Le test des MIB SNMP peut être effectué à l'aide de l'utilitaire `snmpwalk`. Si vous ne l'avez pas installé, utilisez les instructions suivantes.

Sur les systèmes basés sur Debian :

```
apt install snmp
```

Sur les systèmes basés sur RedHat :

```
dnf install net-snmp-utils
```

Après cela, la commande suivante ne doit pas renvoyer d'erreur lorsque vous interrogez un périphérique réseau :

```
$ snmpwalk -v 2c -c public <NETWORK DEVICE IP> ifInOctets
IF-MIB::ifInOctets.1 = Counter32: 176137634
IF-MIB::ifInOctets.2 = Counter32: 0
IF-MIB::ifInOctets.3 = Counter32: 240375057
IF-MIB::ifInOctets.4 = Counter32: 220893420
[...]
```

Utilisation des MIB dans Zabbix

Le point le plus important à garder à l'esprit est que les processus Zabbix ne sont pas informés des modifications apportées aux fichiers MIB. Ainsi, après chaque modification, vous devez redémarrer le serveur ou le proxy Zabbix, par exemple :

```
systemctl restart zabbix-server
```

Après cela, les modifications apportées aux fichiers MIB prennent effet.

Utilisation de fichiers MIB personnalisés

Il existe des fichiers MIB standard fournis avec chaque distribution GNU/Linux. Mais certains fabricants d'équipements fournissent les leurs.

Supposons que vous souhaitiez utiliser le fichier MIB CISCO-SMI. Les instructions suivantes le téléchargeront et l'installeront :

```
wget ftp://ftp.cisco.com/pub/mibs/v2/CISCO-SMI.my -P /tmp
mkdir -p /usr/local/share/snmp/mibs
grep -q '^mibdirs +/usr/local/share/snmp/mibs' /etc/snmp/snmp.conf 2>/dev/null || echo "mibdirs +/usr/loca
cp /tmp/CISCO-SMI.my /usr/local/share/snmp/mibs
```

Vous devriez maintenant pouvoir l'utiliser. Essayez de traduire le nom de l'objet *ciscoProducts* du fichier MIB en OID :

```
snmptranslate -IR -On CISCO-SMI::ciscoProducts
.1.3.6.1.4.1.9.1
```

Si vous recevez des erreurs au lieu de l'OID, assurez-vous qu'aucune des commandes précédentes n'a renvoyé d'erreur.

La traduction du nom de l'objet a fonctionné, vous êtes prêt à utiliser un fichier MIB personnalisé. Notez le préfixe du nom MIB (*CISCO-SMI::*) utilisé dans la requête. Vous en aurez besoin lors de l'utilisation des outils en ligne de commande ainsi que de Zabbix.

N'oubliez pas de redémarrer le serveur/proxy Zabbix avant d'utiliser ce fichier MIB dans Zabbix.

Attention:
Gardez à l'esprit que les fichiers MIB peuvent avoir des dépendances. Autrement dit, un MIB peut en nécessiter un autre. Afin de satisfaire ces dépendances, vous devez installer tous les fichiers MIB concernés.

SNMP trap

Vue d'ensemble

La réception des traps SNMP est l'inverse de l'interrogation des appareils compatibles SNMP.

Dans ce cas, les informations sont envoyées depuis un appareil compatible SNMP vers `snmptrapd` et sont collectées ou "trapées" par le serveur Zabbix ou le proxy Zabbix à partir d'un fichier.

En général, les traps sont envoyés lorsqu'un certain état change, et l'agent se connecte au serveur sur le port 162 (par opposition au port 161 côté agent, qui est utilisé pour les requêtes). L'utilisation des traps peut permettre de détecter certains problèmes brefs qui surviennent entre deux intervalles de requête et qui peuvent être manqués par les données de requête.

La réception des traps SNMP dans Zabbix est conçue pour fonctionner avec `snmptrapd` et l'un des mécanismes de transmission des traps vers Zabbix - soit un script Bash ou Perl, soit SNMPTT.

Note:
La manière la plus simple de configurer la surveillance des traps après avoir configuré Zabbix consiste à utiliser la solution par script Bash, car Perl et SNMPTT sont souvent absents des distributions modernes et nécessitent une configuration plus complexe. Cependant, cette solution utilise un script configuré comme `traphandle`. Pour de meilleures performances sur les systèmes de production, utilisez la solution Perl intégrée (soit un script avec l'option `do perl`, soit SNMPTT).

Le flux de traitement de la réception d'un trap :

1. `snmptrapd` reçoit un trap.
2. `snmptrapd` transmet le trap au script récepteur (Bash, Perl) ou à SNMPTT.
3. Le récepteur analyse, met en forme et écrit le trap dans un fichier.
4. Le collecteur de traps SNMP de Zabbix lit et analyse le fichier de traps.
5. Pour chaque trap, Zabbix recherche tous les éléments de collecteur de traps SNMP dont les interfaces d'hôte correspondent à l'adresse du trap reçu. Notez que seul le *IP* ou le *DNS* sélectionné dans l'interface d'hôte est utilisé lors de la correspondance.
6. Pour chaque élément trouvé, le trap est comparé à l'expression régulière dans `snmptrap[regexp]`. Le trap est défini comme valeur de **tous** les éléments correspondants. Si aucun élément correspondant n'est trouvé et qu'il existe un élément `snmptrap.fallback`, le trap est défini comme sa valeur.
7. Si le trap n'a été défini comme valeur d'aucun élément, Zabbix consigne par défaut le trap non apparié. (Cela est configuré par *Consigner les traps SNMP non appariés* dans *Administration > Général > Autre*.)

Remarques sur le basculement HA

Lors du basculement d'un nœud de haute disponibilité (HA), Zabbix continuera le traitement après le dernier enregistrement dans le dernier horodatage ISO 8601 ; si le même enregistrement n'est pas trouvé, seul l'horodatage sera utilisé pour identifier la dernière position.

Configuration des traps SNMP

Ce type d'élément nécessite la configuration d'interface suivante :

Étape 1: Créer une interface SNMP pour l'hôte

1. Dans *Collecte de données > Hôtes*, créez un hôte ou modifiez celui qui existe déjà.
2. Dans le champ *Interfaces*, sélectionnez le type d'interface *SNMP*.
3. Saisissez l'adresse IP/le nom DNS et le numéro de port.
4. Sélectionnez la *version SNMP* dans la liste déroulante : (*SNMPv1*, *SNMPv2*, ou *SNMPv3*) et ajoutez les informations d'identification de l'interface en fonction de la version SNMP sélectionnée.

L'adresse de chaque trap reçu sera comparée aux adresses IP et DNS de toutes les interfaces SNMP afin de trouver les hôtes correspondants.

Étape 2: Configurer l'élément

1. Pour l'hôte, créez un élément ou modifiez celui qui existe déjà.
2. Dans le champ *Clé*, utilisez l'une des clés de trap SNMP :

- `snmptrap[regexp]` - capture tous les traps SNMP qui correspondent à l'expression régulière spécifiée dans le paramètre `regexp` ; si `regexp` n'est pas spécifié, capture n'importe quel trap.
 Les macros utilisateur et les expressions régulières globales sont prises en charge dans le paramètre.
 Valeur de retour : trap SNMP.
 Cet élément peut être défini uniquement pour les interfaces SNMP.
- `snmptrap.fallback` - capture tous les traps SNMP qui n'ont été capturés par aucun des éléments `snmptrap[]` pour cette interface.
 Valeur de retour : trap SNMP.
 Cet élément peut être défini uniquement pour les interfaces SNMP.

Note:
La correspondance d'expressions régulières multilignes n'est pas prise en charge pour le moment.

3. Définissez *Type d'information* sur *Log* afin que les horodatages soient analysés. D'autres formats, tels que *Numérique*, sont également acceptables, mais peuvent nécessiter un gestionnaire de trap personnalisé.

1 Configuration de la supervision des traps SNMP

Configuration du serveur/proxy Zabbix

Pour lire les traps, le serveur ou le proxy Zabbix doit être configuré pour démarrer le processus SNMP trapper et pointer vers le fichier de traps qui est écrit par SNMPTT ou par un récepteur de traps Bash/Perl. Pour ce faire, modifiez le fichier de configuration (`zabbix_server.conf` ou `zabbix_proxy.conf`) :

```
StartSNMPTrapper=1
SNMPTrapperFile=[TRAP FILE]
```

Warning:
Si le paramètre systemd `PrivateTmp` est utilisé, il est peu probable que ce fichier fonctionne dans `/tmp`.

Configuration du récepteur de traps Bash

Exigences : uniquement snmptrapd.

Un script de récepteur de traps Bash peut être utilisé pour transmettre des traps au serveur Zabbix depuis snmptrapd à l'aide du fichier trapper. Pour le configurer, ajoutez l'option `traphandle` au fichier de configuration de snmptrapd (`snmptrapd.conf`), voir l'exemple.

Note:
Il peut être nécessaire de redémarrer snmptrapd pour prendre en compte les modifications de sa configuration.

Configuration du récepteur de traps Perl

Prérequis: Perl, Net-SNMP compilé avec `--enable-embedded-perl` (activé par défaut depuis Net-SNMP 5.4)

Un récepteur de traps Perl (voir `misc/snmptrap/zabbix_trap_receiver.pl`) peut être utilisé pour transmettre les traps au serveur Zabbix directement depuis snmptrapd. Pour le configurer:

- Ajoutez le script Perl au fichier de configuration de snmptrapd (`snmptrapd.conf`), par exemple:

```
perl do "[FULL PATH TO PERL RECEIVER SCRIPT]";
```

- Configurez le récepteur, par exemple:

```
$SNMPTrapperFile = '[TRAP FILE]';
$DateTimeFormat = '[DATE TIME FORMAT]';
```

Note:
snmptrapd peut devoir être redémarré pour prendre en compte les modifications de sa configuration.

Note:
Si le nom du script n'est pas entre guillemets, snmptrapd refusera de démarrer avec des messages similaires à ceux-ci:


```
Regexp modifiers "/l" and "/a" are mutually exclusive at (eval 2) line 1, at end of line
Regexp modifier "/l" may not appear twice at (eval 2) line 1, at end of line
```

Configuration de SNMPTT

Au départ, `snmptrapd` doit être configuré pour utiliser SNMPTT.

Note:
Pour de meilleures performances, SNMPTT doit être configuré comme un démon en utilisant `snmptthandler-embedded` pour lui transmettre les traps. Voir les instructions pour configurer SNMPTT.

Lorsque SNMPTT est configuré pour recevoir les traps, configurez `snmptt.ini` :

1. Activez l'utilisation du module Perl du package `NET-SNMP` :

```
net_snmp_perl_enable = 1
```

2. Enregistrez les traps dans le fichier de traps qui sera lu par Zabbix :

```
log_enable = 1
log_file = [TRAP FILE]
```

3. Définissez le format de date et d'heure :

```
date_time_format = %Y-%m-%dT%H:%M:%S%z
```

Warning:
Le package `net-snmp-perl` a été supprimé dans RHEL 8.0-8.2 ; il a été réintégré dans RHEL 8.3. Pour plus d'informations, consultez les problèmes connus.

Formatez maintenant les traps pour que Zabbix puisse les reconnaître (modifiez `snmptt.conf`) :

1. Chaque instruction `FORMAT` doit commencer par `ZBXTRAP [address]`, où `[address]` sera comparée aux adresses IP et DNS des interfaces SNMP sur Zabbix. Par exemple :

```
EVENT coldStart .1.3.6.1.6.3.1.1.5.1 "Status Events" Normal
FORMAT ZBXTRAP $aA Device reinitialized (coldStart)
```

2. Voir plus bas les détails sur le format des traps SNMP.

Attention:
N'utilisez pas de traps inconnus - Zabbix ne pourra pas les reconnaître. Les traps inconnus peuvent être gérés en définissant un événement général dans `snmptt.conf` :


```
EVENT general .* "General event" Normal
```

Format du trap SNMP

Tous les récepteurs de traps Perl personnalisés et la configuration SNMPTT doivent formater le trap de la manière suivante :

```
[timestamp] [the trap, part 1] ZBXTRAP [address] [the trap, part 2]
```

où

- `[timestamp]` - l'horodatage au format `%Y-%m-%dT%H:%M:%S%z`.
- `ZBXTRAP` - en-tête indiquant qu'un nouveau trap commence sur cette ligne.
- `[address]` - adresse IP utilisée pour trouver l'hôte de ce trap.

Notez que `ZBXTRAP` et `[address]` seront supprimés du message pendant le traitement. Si le trap est formaté autrement, Zabbix peut analyser les traps de manière inattendue.

Exemple de trap :

```
2024-01-11T15:28:47+0200 .1.3.6.1.6.3.1.1.5.3 Normal "Status Events" localhost - ZBXTRAP 192.168.1.1 Link
```

Cela donnera le trap suivant pour l'interface SNMP avec IP=`192.168.1.1` :

```
2024-01-11T15:28:47+0200 .1.3.6.1.6.3.1.1.5.3 Normal "Status Events"
localhost - Link down on interface 2. Admin state: 1. Operational state: 2
```

Configuration système requise

Note:
Il est recommandé d'installer les fichiers MIB afin de garantir que les valeurs des éléments soient affichées dans le format correct. Sans les fichiers MIB, des problèmes de formatage peuvent survenir, comme l'affichage des valeurs en HEX au lieu de l'UTF-8 ou inversement.

Prise en charge des gros fichiers

Zabbix prend en charge les gros fichiers pour les fichiers du collecteur SNMP trapper. La taille maximale de fichier que Zabbix peut lire est de 2^63 (8 EiB). Notez que le système de fichiers peut imposer une limite inférieure à la taille du fichier.

Rotation des journaux

Zabbix ne fournit aucun système de rotation des journaux - cela doit être géré par l'utilisateur. La rotation des journaux doit d'abord renommer l'ancien fichier, puis seulement ensuite le supprimer, afin qu'aucun trap ne soit perdu :

1. Zabbix ouvre le fichier de trap à la dernière position connue et passe à l'étape 3.
2. Zabbix vérifie si le fichier actuellement ouvert a été soumis à une rotation en comparant le numéro d'inode au numéro d'inode du fichier de trap défini. S'il n'y a aucun fichier ouvert, Zabbix réinitialise la dernière position et revient à l'étape 1.
3. Zabbix lit les données du fichier actuellement ouvert et définit la nouvelle position.
4. Les nouvelles données sont analysées. S'il s'agissait du fichier rotaté, le fichier est fermé et le processus revient à l'étape 2.
5. S'il n'y a pas de nouvelles données, Zabbix dort pendant 1 seconde et revient à l'étape 2.

Système de fichiers

En raison de l'implémentation du fichier de trap, Zabbix a besoin que le système de fichiers prenne en charge les inodes afin de différencier les fichiers (l'information est obtenue par un appel `stat()`).

Exemples de configuration utilisant différentes versions du protocole SNMP

Cet exemple utilise snmptrapd et un script récepteur Bash pour transmettre les traps au serveur Zabbix.

Configuration :

1. Configurez Zabbix pour démarrer le trapper SNMP et définissez le fichier de traps. Ajoutez les lignes suivantes à `zabbix_server.conf` :

```
StartSNMPTrapper=1
SNMPTrapperFile=/var/lib/zabbix/snmptraps/snmptraps.log
```

2. Téléchargez le script Bash vers `/usr/sbin/zabbix_trap_handler.sh` :

```
curl -o /usr/sbin/zabbix_trap_handler.sh https://raw.githubusercontent.com/zabbix/zabbix-docker/trunk/temp
```

Si nécessaire, ajustez la variable `ZABBIX_TRAPS_FILE` dans le script. Pour utiliser la valeur par défaut, créez d'abord le répertoire parent :

```
mkdir -p /var/lib/zabbix/snmptraps
```

3. Ajoutez les lignes suivantes à `snmtrapd.conf` (consultez l'exemple fonctionnel) :

```
traphandle default /bin/bash /usr/sbin/zabbix_trap_handler.sh
```

> **Note:**
> Il peut être nécessaire de redémarrer snmptrapd pour qu'il prenne en compte les modifications apportées à sa configuration.

4. Créez un élément SNMP TEST (tenez compte des conditions de configuration initiales) :

   Type : *SNMP trap*
 Type d'information : *Log*
 Interface de l'hôte : *SNMP 127.0.0.1*
 Clé : `snmptrap["linkUp"]`
 Format de l'heure du journal : `yyyyMMdd.hhmmss`

Notez que le format de date et d'heure ISO 8601 est utilisé.

5. Nous allons ensuite configurer `snmptrapd` pour la version du protocole SNMP choisie et envoyer des traps de test à l'aide de l'utilitaire `snmptrap`.

SNMPv1, SNMPv2

Les protocoles SNMPv1 et SNMPv2 utilisent l'authentification par « chaîne de communauté ». Dans l'exemple ci-dessous, nous utiliserons « secret » comme chaîne de communauté. Cette valeur doit être définie sur les émetteurs de traps SNMP.

Veuillez noter que, bien qu'il soit encore largement utilisé dans les environnements de production, SNMPv2 n'offre aucun chiffrement ni authentification réelle de l'émetteur. Les données sont envoyées en texte brut ; ces versions du protocole doivent donc être utilisées uniquement dans des environnements sécurisés, tels qu'un réseau privé, et ne doivent jamais être utilisées sur un réseau public ou tiers.

La version 1 de SNMP n'est plus vraiment utilisée de nos jours, car elle ne prend pas en charge les compteurs 64 bits et est considérée comme un protocole obsolète.

Pour autoriser la réception de traps SNMPv1 ou SNMPv2, ajoutez la ligne suivante à `snmptrapd.conf`. Remplacez `secret` par la chaîne de communauté SNMP configurée sur les émetteurs de traps SNMP :

```
authCommunity log,execute,net secret
```

Nous pouvons ensuite envoyer un trap de test à l'aide de `snmptrap`. Dans cet exemple, nous utiliserons l'OID courant « link up » :

```
snmptrap -v 2c -c secret localhost '' linkUp.0
```

SNMPv3

SNMPv3 corrige les problèmes de sécurité de SNMPv1/v2 et fournit une authentification ainsi qu'un chiffrement. Vous pouvez utiliser les méthodes d'authentification MD5 ou SHA, ainsi que DES ou AES comme chiffre.

Configurez `snmptrapd` pour accepter les traps SNMPv3 en ajoutant les lignes suivantes à `snmptrapd.conf` :

```
##### Créer un utilisateur USM SNMPv3 avec authentification et confidentialité
createUser -e <engineID> <securityName> SHA <authKey> AES <privKey>
##### Autoriser l'utilisateur à journaliser les traps et à exécuter les scripts de gestion des traps
authUser log,execute <securityName>
```

Par exemple :

```
createUser -e 0x8000000001020304 traptest SHA myAuthPassword AES myPrivPassword
authUser log,execute traptest
```

Attention:
Notez le mot-clé « execute », qui permet d'exécuter des scripts pour ce modèle de sécurité utilisateur.

Pour tester la configuration des traps SNMPv3, envoyez un trap de test à l'aide de la commande suivante :

```
snmptrap -v 3 -n "" -a SHA -A myAuthPassword -x AES -X myPrivPassword -l authPriv -u traptest -e 0x800000
```

Warning:
Si vous souhaitez utiliser des méthodes de chiffrement fortes telles qu'AES192 ou AES256, utilisez net-snmp à partir de la version 5.8. Vous devrez peut-être le recompiler avec l'option `configure` : `--enable-blumenthal-aes`. Les anciennes versions de net-snmp ne prennent pas en charge AES192/AES256. Voir également : Strong Authentication or Encryption.

Vérification

Dans les deux exemples, vous verrez des lignes similaires dans votre fichier `/var/lib/zabbix/snmptraps/snmptraps.log` :

```
2024-01-30T10:04:23+0200 ZBXTRAP 127.0.0.1
UDP: [127.0.0.1]:56585->[127.0.0.1]:162
DISMAN-EVENT-MIB::sysUpTimeInstance = 2538834
SNMPv2-MIB::snmpTrapOID.0 = IF-MIB::linkUp.0
```

La valeur de l'élément dans Zabbix sera :

```
2024-01-30 10:04:23 2024-01-30 10:04:21

2024-01-30T10:04:21+0200 UDP: [127.0.0.1]:56585->[127.0.0.1]:162
DISMAN-EVENT-MIB::sysUpTimeInstance = 2538834
SNMPv2-MIB::snmpTrapOID.0 = IF-MIB::linkUp.0
```

Exemple avec Perl :

```
2024-01-30T11:42:54+0200 ZBXTRAP 127.0.0.1
PDU INFO:
  receivedfrom                   UDP: [127.0.0.1]:58649->[127.0.0.1]:162
  notificationtype               TRAP
  version                        1
  community                      public
  errorstatus                    0
  transactionid                  1
  requestid                      2101882550
  messageid                      0
  errorindex                     0
VARBINDS:
```

```
DISMAN-EVENT-MIB::sysUpTimeInstance type=67 value=Timeticks: (457671) 1:16:16.71
SNMPv2-MIB::snmpTrapOID.0       type=6  value=OID: IF-MIB::linkUp.0
```

Voir également

- Article du blog Zabbix sur les traps SNMP
- Configuration de snmptrapd (documentation officielle net-snmp)
- Configuration de snmptrapd pour recevoir les notifications SNMPv3 (documentation officielle net-snmp)

Zabbix interne

Vue d'ensemble

Les contrôles internes vous permettent de surveiller les processus internes de Zabbix. En d'autres termes, vous pouvez surveiller ce qui se passe sur le serveur Zabbix ou le proxy Zabbix.

Les contrôles internes sont calculés :

- sur le serveur Zabbix - si l'hôte est surveillé par le serveur
- sur le proxy Zabbix - si l'hôte est surveillé par le proxy

Les contrôles internes sont traités par le serveur ou le proxy, quel que soit l'état de maintenance de l'hôte.

Pour utiliser cet élément, sélectionnez le type d'élément **Zabbix internal**.

> **Note:**
> Les contrôles internes sont traités par les pollers Zabbix.

Performances

L'utilisation de certains éléments internes peut affecter négativement les performances. Ces éléments sont les suivants :

- `zabbix[host,,items]`
- `zabbix[host,,items_unsupported]`
- `zabbix[hosts]`
- `zabbix[items]`
- `zabbix[items_unsupported]`
- `zabbix[queue,,]`
- `zabbix[requiredperformance]`
- `zabbix[stats,,,queue,,]`
- `zabbix[triggers]`

Les sections Informations système et File d'attente de l'interface web sont également affectées.

Vérifications prises en charge

Les clés d'éléments sont répertoriées sans paramètres personnalisables ni informations supplémentaires. Cliquez sur la clé d'élément pour afficher tous les détails.

Clé d'élément	Description
zabbix[apm,]	Les statistiques des données de télémétrie reçues par le collecteur APM.
zabbix[boottime]	L'heure de démarrage du processus du serveur Zabbix ou du proxy Zabbix, en secondes.
zabbix[cluster,discovery,nodes]	Détecte les nœuds du cluster à haute disponibilité.
zabbix[connector_queue]	Le nombre de valeurs placées dans la file d'attente du connecteur.
zabbix[discovery_queue]	Le nombre de vérifications réseau placées dans la file d'attente de détection.
zabbix[host,,items]	Le nombre d'éléments activés (pris en charge et non pris en charge) sur le hôte.
zabbix[host,,items_unsupported]	Le nombre d'éléments non pris en charge activés sur le hôte.
zabbix[host,,maintenance]	L'état actuel de maintenance du hôte.
zabbix[host,active_agent,available]	La disponibilité des vérifications par agent actif sur le hôte.
zabbix[host,discovery,interfaces]	Les détails de toutes les interfaces configurées du hôte dans l'interface Zabbix.
zabbix[host,,available]	La disponibilité de l'interface principale pour un type particulier de vérifications sur le hôte.
zabbix[hosts]	Le nombre de hôtes surveillés.
zabbix[items]	Le nombre d'éléments activés (pris en charge et non pris en charge).
zabbix[items_unsupported]	Le nombre d'éléments non pris en charge.
zabbix[java,,]	Les informations sur le Java gateway Zabbix.
zabbix[lld_queue]	Le nombre de valeurs placées dans la file d'attente de traitement de la détection de bas niveau.
zabbix[preprocessing]	Les statistiques des valeurs reçues par le gestionnaire du prétraitement.

Clé d'élément	Description
zabbix[preprocessing_queue,]	Le nombre de valeurs placées dans la file d'attente de prétraitement.
zabbix[process,,,]	Le pourcentage de temps qu'un processus Zabbix particulier ou un groupe de processus (identifié par <type> et <mode>) a passé dans <state>.
zabbix[proxy,,]	Les informations sur le proxy Zabbix.
zabbix[proxy,discovery]	La liste des proxy Zabbix.
zabbix[proxy group,,available]	Le nombre de proxy en ligne dans un groupe de proxy.
zabbix[proxy group,,pavailable]	Le pourcentage de proxy en ligne dans un groupe de proxy.
zabbix[proxy group,,proxies]	La liste des proxy Zabbix dans un groupe de proxy.
zabbix[proxy group,,state]	L'état d'un groupe de proxy.
zabbix[proxy group,discovery]	Renvoie une liste de groupes de proxy avec les données de configuration et les données en temps réel.
zabbix[proxy_buffer,buffer,]	Renvoie les statistiques d'utilisation du tampon mémoire du proxy.
zabbix[proxy_buffer,state,changes]	Renvoie le nombre de changements d'état entre les modes de tampon disque/mémoire depuis le démarrage.
zabbix[proxy_buffer,state,current]	Renvoie l'état de fonctionnement actuel dans lequel les nouvelles données sont stockées.
zabbix[proxy_history]	Le nombre de valeurs dans la table d'historique du proxy en attente d'être envoyées au serveur.
zabbix[queue,,]	Le nombre d'éléments surveillés dans la file d'attente qui sont retardés d'au moins <from> secondes, mais de moins de <to> secondes.
zabbix[rcache,,]	Les statistiques de disponibilité du cache de configuration Zabbix.
zabbix[requiredperformance,]	Les performances requises du serveur Zabbix ou du proxy Zabbix, exprimées en nouvelles valeurs attendues par seconde.
zabbix[stats,,]	Renvoie les métriques internes d'un serveur Zabbix ou d'un proxy. Si <ip> et <port> sont spécifiés, les métriques sont récupérées depuis l'instance distante ; sinon, depuis l'instance locale.
zabbix[stats,,,queue,,]	Renvoie les métriques internes de la file d'attente d'un serveur Zabbix ou d'un proxy. Si <ip> et <port> sont spécifiés, les métriques sont récupérées depuis l'instance distante ; sinon, depuis l'instance locale.
zabbix[tcache,,]	Les statistiques d'efficacité du cache de la fonction de tendance Zabbix.
zabbix[triggers]	Le nombre de déclencheurs activés dans la base de données Zabbix, avec tous les éléments activés sur les hôtes activés.
zabbix[uptime]	La durée de fonctionnement du processus du serveur Zabbix ou du proxy, en secondes.
zabbix[vcache,buffer,]	Les statistiques de disponibilité du cache de valeurs Zabbix.
zabbix[vcache,cache,]	Les statistiques d'efficacité du cache de valeurs Zabbix.
zabbix[version]	La version du serveur Zabbix ou du proxy.
zabbix[vmware,buffer,]	Les statistiques de disponibilité du cache vmware Zabbix.
zabbix[vps,written]	Le nombre total de valeurs d'historique écrites dans la base de données.
zabbix[wcache,,]	Les statistiques et la disponibilité du cache d'écriture Zabbix.

Détails de la clé d'élément

- Les paramètres sans crochets angulaires sont obligatoires et doivent être utilisés *tels quels* (par exemple, « host » et « available » dans `zabbix[host,<type>,available]`).
- Les paramètres entre crochets angulaires **< >** doivent être remplacés par une valeur valide. Si un paramètre a une valeur par défaut, il peut être omis.
- Les valeurs des éléments et des paramètres d'élément indiqués comme « non pris en charge sur le proxy » ne peuvent être récupérées que si l'hôte est supervisé par le serveur. Inversement, les valeurs « non prises en charge sur le serveur » ne peuvent être récupérées que si l'hôte est supervisé par le proxy.

zabbix[apm,<type>]

 Les statistiques des données de télémétrie reçues par le collecteur APM :

- *requests* - le nombre de requêtes reçues par le collecteur APM (compteur) :
  - *accepted* - le nombre de requêtes acceptées pour traitement ;
  - *dropped* - le nombre de requêtes abandonnées ;
- *rows* - le nombre de lignes de données de télémétrie écrites dans la source de données, par type de données (compteur).

Valeur de retour : *objet JSON*.

Paramètres :

- **type** - *traces*, *metrics* ou *logs*. Si ce paramètre n'est pas spécifié, les statistiques de tous les types sont renvoyées.

Commentaires :

- Cet élément n'est pas pris en charge sur le serveur Zabbix.

Exemple de valeurs renvoyées pour `zabbix[apm]` :


```
{
  "requests": {
    "accepted": 15230,
    "dropped": 12
  },
  "rows": {
    "traces": 48710,
    "metrics_gauge": 31250,
    "metrics_sum": 18400,
    "metrics_histogram": 5120,
    "metrics_exponential_histogram": 640,
    "metrics_summary": 210,
    "logs": 22870
  }
}
```


zabbix[boottime]

 L'heure de démarrage du processus Zabbix server ou Zabbix proxy, en secondes.
 Valeur de retour : *Integer*.

zabbix[cep]

 Complex event processing statistics.
 Return value: *JSON*.

zabbix[cluster,discovery,nodes]

 Détecte les nœuds du cluster à haute disponibilité.
 Valeur renvoyée : *objet JSON*.

Commentaires :

- Cet élément peut être utilisé dans la découverte de bas niveau.
- Les champs renvoyés correspondent principalement aux propriétés de l'objet nœud de haute disponibilité.
- Champs supplémentaires :
  - `db_timestamp` - heure actuelle du serveur (horodatage Unix) incluse dans la réponse
  - `lastaccess_age` - nombre de secondes écoulées depuis le dernier signal de présence du nœud (db_timestamp - lastaccess)
  - `status` - état du nœud :
    - 0 - en attente
    - 1 - arrêté manuellement
    - 2 - indisponible
    - 3 - actif

Exemple de valeurs renvoyées :


```
[
  {
    "id": "ckvupihk70001z8mkpw5cg0u3",
    "name": "zabbix-prod-01",
    "status": 3,
    "address": "10.0.4.12:10051",
    "port": 10051,
    "lastaccess": 1756115995,
    "db_timestamp": 1756116000,
    "lastaccess_age": 5
  },
  {
    "id": "ckvx2a9k70004b1nq2hz9d7f",
    "name": "zabbix-standby-02",
    "status": 0,
    "address": "10.0.4.13:10051",
    "port": 10051,
    "lastaccess": 1756115550,
```

```
    "db_timestamp": 1756116000,
    "lastaccess_age": 450
  },
  {
    "id": "ckw0bq3l70007y4r1a0m5kz8",
    "name": "zabbix-backup-eu1",
    "status": 1,
    "address": "backup.example.com:10051",
    "port": 10051,
    "lastaccess": 1756105080,
    "db_timestamp": 1756116000,
    "lastaccess_age": 10920
  }
]
```


zabbix[connector_queue]

 Le nombre de valeurs mises en file d'attente dans la file d'attente du connecteur.
 Valeur de retour : *Integer*.

zabbix[discovery_queue]

 Le nombre de vérifications réseau mises en file d'attente dans la file de découverte.
 Valeur de retour : *Integer*.

zabbix[host,„items]

 Le nombre d'éléments activés (pris en charge et non pris en charge) sur l'hôte.
 Valeur de retour : *Integer*.

zabbix[host,„items_unsupported]

 Le nombre d'éléments non pris en charge activés sur l'hôte.
 Valeur de retour : *Integer*.

zabbix[host,„maintenance]

 Le statut de maintenance actuel de l'hôte.
 Valeurs de retour : *0* - état normal ; *1* - maintenance avec collecte de données ; *2* - maintenance sans collecte de données.

Commentaires :

- Cet élément est toujours traité par le serveur Zabbix, quel que soit l'emplacement de l'hôte (sur le serveur ou le proxy). Le proxy ne recevra pas cet élément avec les données de configuration.
- Le deuxième paramètre doit être vide et est réservé à une utilisation future.

zabbix[host,active_agent,available]

 La disponibilité des vérifications de l'agent actif sur l'hôte.
 Valeurs de retour : *0* - inconnue ; *1* - disponible ; *2* - non disponible.

zabbix[host,discovery,interfaces]

 The details of all configured interfaces of the host in Zabbix frontend.
 Return value: *JSON object*.

Comments:

- This item can be used in low-level discovery;
- This item is not supported on Zabbix proxy.

zabbix[host,<type>,available]

 La disponibilité de l'interface principale d'un type de contrôles particulier sur l'hôte.
 Valeurs de retour : *0* - non disponible ; *1* - disponible ; *2* - inconnue.

Paramètres :

- **type** - *agent*, *snmp*, *ipmi* ou *jmx*.

Commentaires :

- La valeur de l'élément est calculée conformément aux paramètres de configuration concernant l'injoignabilité/indisponibilité de l'hôte.

zabbix[hosts]

 Le nombre d'hôtes surveillés.
 Valeur de retour : *Integer*.

zabbix[items]

 Le nombre d'éléments de données activés (pris en charge et non pris en charge).
 Valeur de retour : *Integer*.

zabbix[items_unsupported]

 Le nombre d'éléments non pris en charge.
 Valeur de retour : *Integer*.

zabbix[java,„<param>]

 Informations sur le Java gateway Zabbix.
 Valeurs de retour : *1* - si `<param>` est *ping* ; *version du Java gateway* - si `<param>` est *version* (par exemple : "8.0.0").

Paramètres :

- **param** - *ping* ou *version*.

Commentaires :

- Cet élément peut être utilisé pour vérifier la disponibilité du Java gateway à l'aide de la fonction de déclencheur `nodata()`.
- Le deuxième paramètre doit être vide et est réservé à un usage futur.

zabbix[lld_queue]

 Le nombre de valeurs mises en file d'attente dans la file de traitement de la découverte de bas niveau.
 Valeur de retour : *Integer*.

Commentaires :

- Cet élément peut être utilisé pour surveiller la longueur de la file de traitement de la découverte de bas niveau.

zabbix[preprocessing]

 Statistiques des valeurs reçues par le gestionnaire de prétraitement :

- *queued* - le nombre et la taille des valeurs en file d'attente qui nécessitent un prétraitement (compteur)
- *direct* - le nombre et la taille des valeurs en file d'attente qui ne nécessitent pas de prétraitement (compteur)
- *queue* - le nombre de valeurs placées dans la file d'attente de prétraitement (identique à `zabbix[preprocessing_queue]`)

Valeur de retour : *JSON*.

Exemple de valeurs de retour :

```
{"data":
    {
    "queued": {
        "count": 106,
        "size": 58620
    },
    "direct": {
        "count": 395,
        "size": 33843
    },
    "queue": 0
    }
}
```

zabbix[preprocessing_queue]

 Le nombre de valeurs mises en file d'attente dans la file de prétraitement.
 Valeur de retour : *Integer*.

Commentaires :

- Cet élément peut être utilisé pour surveiller la longueur de la file de prétraitement.

zabbix[process,<type>,<mode>,<state>]

 Le pourcentage de temps qu'un processus Zabbix particulier ou un groupe de processus (identifié par `<type>` et `<mode>`) a passé dans l'état `<state>`. Il est calculé uniquement pour la dernière minute.
 Valeur retournée : *Float*.

Paramètres :

- **type** :
  - pour les processus du serveur : *agent poller*, *alert manager*, *alert syncer*, *alerter*, *availability manager*, *browser poller*, *configuration syncer*, *configuration syncer worker*, *connector manager*, *connector worker*, *discovery manager*, *discovery worker*, *escalator*, *ha manager*, *history poller*, *history syncer*, *housekeeper*, *http agent poller*, *http poller*, *icmp pinger*, *internal poller ipmi manager*, *ipmi poller*, *java poller*, *lld manager*, *lld worker*, *odbc poller*, *poller*, *preprocessing*

manager, *preprocessing worker*, *proxy group manager*, *proxy poller*, *self-monitoring*, *service manager*, *snmp poller*, *snmp trapper*, *task manager*, *timer*, *trapper*, *trigger housekeeper*, *unreachable poller*, *vmware collector* ;
  - pour les processus du proxy : *agent poller*, *apm manager*, *apm worker*, *availability manager*, *browser poller*, *configuration syncer*, *data sender*, *discovery manager*, *discovery worker*, *history syncer*, *housekeeper*, *http agent poller*, *http poller*, *icmp pinger*, *internal poller ipmi manager*, *ipmi poller*, *java poller*, *odbc poller*, *poller*, *preprocessing manager*, *preprocessing worker*, *self-monitoring*, *snmp poller*, *snmp trapper*, *task manager*, *trapper*, *unreachable poller*, *vmware collector* ;
- **mode** :
  - *avg* - valeur moyenne pour tous les processus d'un type donné (par défaut) ;
  - *count* - renvoie le nombre de forks pour un type de processus donné ; <state> ne doit pas être spécifié ;
  - *max* - valeur maximale ;
  - *min* - valeur minimale ;
  - *<process number>* - numéro du processus (compris entre 1 et le nombre d'instances préforkées ; par exemple, si 4 trappers sont en cours d'exécution, la valeur est comprise entre 1 et 4) ;
- **state** :
  - *busy* - le processus est occupé, par exemple, en train de traiter une requête (par défaut) ;
  - *idle* - le processus est inactif et n'effectue aucune tâche.

Commentaires :

- Si <mode> est un numéro de processus Zabbix qui n'est pas en cours d'exécution (par exemple, si 5 pollers sont en cours d'exécution et que <mode> est défini sur 6), cet élément devient non pris en charge.
- Les valeurs minimale et maximale font référence au pourcentage d'utilisation d'un seul processus. Ainsi, si, dans un groupe de 3 pollers, les pourcentages d'utilisation par processus sont de 2, 18 et 66, *min* renvoie 2 et *max* renvoie 66.
- Les processus indiquent ce qu'ils font dans la mémoire partagée et le processus d'auto-surveillance résume ces données chaque seconde. Les changements d'état (occupé/inactif) sont enregistrés lors d'une modification : ainsi, un processus qui devient occupé l'enregistre comme tel et ne modifie ni ne met à jour son état jusqu'à ce qu'il devienne inactif. Cela garantit que même les processus complètement bloqués sont correctement enregistrés comme étant occupés à 100 %.
- Actuellement, « occupé » signifie « qui ne dort pas », mais d'autres états pourraient être introduits à l'avenir : attente de verrous, exécution de requêtes de base de données, etc. Notez que les pollers asynchrones sont considérés comme occupés lorsqu'ils ont atteint la limite définie par le paramètre de configuration MaxConcurrentChecksPerPoller du serveur/proxy.
- Sous Linux et sur la plupart des autres systèmes, la résolution est de 1/100 de seconde.

Exemples :

```
zabbix[process,poller,avg,busy] #temps moyen que les processus poller ont passé à effectuer une tâche au c
zabbix[process,"icmp pinger",max,busy] #temps maximal passé à effectuer une tâche par n'importe quel proces
zabbix[process,"history syncer",2,busy] #temps passé à effectuer une tâche par le processus history syncer
zabbix[process,trapper,count] #nombre de processus trapper actuellement en cours d'exécution
```

zabbix[proxy,<name>,<param>]

 Informations sur le proxy Zabbix.
 Valeur retournée : *Integer*.

Paramètres :

- **name** - nom du proxy ;
- **param** :
  - *lastaccess* - horodatage du dernier message de pulsation reçu du proxy ;
  - *delay* - durée pendant laquelle les valeurs collectées restent non envoyées ; calculée comme suit : « délai du proxy » + (« heure actuelle du serveur » - « lastaccess du proxy »), où le « délai du proxy » correspond à la différence entre l'heure actuelle du proxy et l'horodatage de la plus ancienne valeur non envoyée sur le proxy.

Commentaires :

- Cet élément n'est pas pris en charge sur les hôtes surveillés par un proxy ou un groupe de proxies.
- La fonction `fuzzytime()` peut être utilisée pour vérifier la disponibilité du proxy.

Exemple :

```
zabbix[proxy,"Germany",lastaccess] #horodatage du dernier message de pulsation reçu du proxy "Germany"
```

zabbix[proxy,discovery]

 La liste des proxy Zabbix avec le nom, le mode, le chiffrement, la compression, la version, la dernière détection, le nombre d'hôtes, le nombre d'éléments, les valeurs requises par seconde (vps), l'état de la version (actuelle/obsolète/non prise en charge), les délais d'expiration par type d'élément, le nom du groupe de proxy (si le proxy appartient à un groupe), l'état (inconnu/hors ligne/en ligne).
 Valeur renvoyée : *objet JSON*.

Commentaire :

- Cet élément n'est pas pris en charge sur les hôtes surveillés par un proxy ou un groupe de proxy.

Exemple de valeurs renvoyées :


```
[
  {
    "name":"Zabbix proxy",
    "passive":false,
    "unencrypted":true,
    "psk":false,
    "cert":false,
    "items":22,
    "compression":true,
    "version":"8.0.0beta2",
    "compatibility":1,
    "last_seen":1,
    "hosts":1,
    "requiredperformance":0.333333,
    "timeouts": {
        "zabbix_agent":3,
        "simple_check":3,
        "snmp_check":3,
        "external_check":3,
        "db_monitor":3,
        "http_agent":3,
        "ssh_agent":3,
        "telnet_agent":3,
        "script":3,
        "browser":60
    },
    "state":"online"
  }
]
```


zabbix[proxy group,<name>,available]

 Le nombre de proxies en ligne dans un groupe de proxies.
 Valeur renvoyée : *Integer*.

Paramètres :

- **name** - le nom du groupe de proxies.

Commentaire :

- Cet élément n'est pas pris en charge sur les hôtes surveillés par un proxy ou un groupe de proxies.

zabbix[proxy group,<name>,pavailable]

 Le pourcentage de proxies en ligne dans un groupe de proxies.
 Valeur renvoyée : *Float*.

Paramètres :

- **name** - le nom du groupe de proxies.

Commentaire :

- Cet élément n'est pas pris en charge sur les hôtes surveillés par un proxy ou un groupe de proxies.

zabbix[proxy group,<name>,proxies]

 La liste des proxies Zabbix d'un groupe de proxies avec le nom, le mode, le chiffrement, la compression, la version, la dernière détection, le nombre d'hôtes, le nombre d'éléments, les valeurs par seconde requises (vps), l'état de la version (actuelle/obsolète/non prise en charge), les délais d'expiration, le nom du groupe de proxies et l'état (inconnu/hors ligne/en ligne).
 Valeur retournée : *JSON*.

Paramètres :

- **name** - le nom du groupe de proxies.

Commentaire :

- Cet élément n'est pas pris en charge sur les hôtes surveillés par un proxy ou un groupe de proxies.

zabbix[proxy group,<name>,state]

 L'état d'un groupe de proxy.
 Valeur retournée : *0* - inconnu ; *1* - hors ligne ; *2* - en cours de récupération ; *3* - en ligne ; *4* - en dégradation.

Paramètres :

- **name** - nom du groupe de proxy.

Commentaire :

- Cet élément n'est pas pris en charge sur les hôtes surveillés par un proxy ou un groupe de proxy.

zabbix[proxy group,discovery]

 Renvoie une liste de groupes de proxy avec leurs données de configuration et leurs données en temps réel. Les données de configuration incluent le nom du groupe de proxy, le délai de basculement et le nombre minimal de proxy en ligne requis. Les données en temps réel incluent l'état du groupe de proxy (voir les commentaires pour plus de détails), le nombre de proxy en ligne et le pourcentage de proxy en ligne.
 Valeur renvoyée : *JSON*.

Commentaires :

- Cet élément n'est pas pris en charge sur les hôtes surveillés par un proxy ou un groupe de proxy.
- Cet élément ne renvoie pas les proxy sans groupe.
- Si la valeur de `failover_delay` ou de `min_online` n'est pas valide, une valeur spéciale −1 est signalée pour l'indiquer. Des valeurs non valides peuvent apparaître si des macros sont utilisées pour la configuration et qu'elles ne peuvent pas être développées en une valeur valide.
- L'état du groupe de proxy est signalé sous forme d'entier : *0* - inconnu ; *1* - hors ligne ; *2* - récupération ; *3* - en ligne ; *4* - dégradation.

Exemple de valeurs renvoyées :

```
{
    "groups": [
       { "name": "Riga", "failover_delay": 60, "min_online": 1 },
       { "name": "Tokyo", "failover_delay": 60, "min_online": 2 },
       { "name": "Porto Alegre", "failover_delay": 60, "min_online": 3 }
    ],
    "details": {
        "Riga": { "state": 3, "available": 10, "pavailable": 20 },
        "Tokyo": { "state": 3, "available": 10, "pavailable": 20 },
        "Porto Alegre": { "state": 1, "available": 0, "pavailable": 0 }
    }
}
```

zabbix[proxy_buffer,buffer,<mode>]

 Statistiques d'utilisation du tampon mémoire du proxy.
 Valeurs retournées : *Integer* (pour la taille) ; *Float* (pour le pourcentage).

Paramètres :

- **mode** :
  - *total* - taille totale du tampon (peut être utilisé pour vérifier si le tampon mémoire est activé) ;
  - *free* - taille du tampon libre ;
  - *pfree* - pourcentage du tampon libre ;
  - *used* - taille du tampon utilisé ;
  - *pused* - pourcentage du tampon utilisé.

Commentaires :

- Retourne une erreur « Proxy memory buffer is disabled » lorsque le tampon mémoire est désactivé ;
- Cet élément n'est pas pris en charge sur le serveur Zabbix.

zabbix[proxy_buffer,state,changes]

 Returns the number of state changes between disk/memory buffer modes since start.
 Return values: *Integer*; *0* - the memory buffer is disabled.

Comments:

- Frequent state changes indicate that either the memory buffer size or age must be increased;

- If the memory buffer state is monitored infrequently (for example, once a minute) then the buffer might flip its state without it being registered.

zabbix[proxy_buffer,state,current]

 Renvoie l'état de fonctionnement actuel dans lequel les nouvelles données sont stockées.
 Valeurs renvoyées : *0* - disque ; *1* - mémoire.

Commentaire :

- 0 est également renvoyé lorsque le tampon mémoire est désactivé.

zabbix[proxy_history]

 Le nombre de valeurs dans la table d'historique du proxy en attente d'être envoyées au serveur.
 Valeurs retournées : *Integer*.

Commentaire :

- Cet élément n'est pas pris en charge sur le serveur Zabbix.

zabbix[queue,<from>,<to>]

 Le nombre d'éléments surveillés dans la file d'attente qui sont retardés d'au moins `<from>` secondes, mais de moins de `<to>` secondes.
 Valeur de retour: *Integer*.

Paramètres:

- **from** - retardé d'au moins (la valeur par défaut est 6 secondes);
- **to** - retardé d'au plus (la valeur par défaut est l'infini).

Commentaires:

- Les suffixes de temps (`s,m,h,d,w`) sont pris en charge dans les paramètres.

> **Attention:**
> Le proxy Zabbix n'est pas informé des périodes de maintenance; voir Calcul des files d'attente pendant la maintenance pour plus de détails.

zabbix[rcache,<cache>,<mode>]

 Les statistiques de disponibilité du cache de configuration Zabbix.
 Valeurs de retour: *Integer* (pour la taille); *Float* (pour le pourcentage).

Paramètres:

- **cache** - *buffer*;
- **mode**:
  - *total* - la taille totale du buffer;
  - *free* - la taille du buffer libre;
  - *pfree* - le pourcentage du buffer libre;
  - *used* - la taille du buffer utilisé;
  - *pused* - le pourcentage du buffer utilisé.

zabbix[requiredperformance]

 Les performances requises du serveur Zabbix ou du proxy Zabbix, en nombre de nouvelles valeurs par seconde attendu.
 Valeur de retour : *Float*.

Commentaires :

- Correspond approximativement à "Required server performance, new values per second" dans *Reports* > *System information*.

zabbix[stats,<ip>,<port>]

 Renvoie les métriques internes d'un serveur Zabbix ou d'un proxy. Si `<ip>` et `<port>` sont spécifiés, les métriques sont récupérées depuis l'instance distante ; sinon, depuis l'instance locale.
 Valeurs de retour : *objet JSON*.

Paramètres :

- **ip** - la liste des adresses IP/DNS/masques réseau des serveurs/proxy à interroger à distance (la valeur par défaut est 127.0.0.1) ;
- **port** - le port du serveur/proxy à interroger à distance (la valeur par défaut est 10051).

Commentaires :

- La requête de statistiques ne sera acceptée que depuis les adresses répertoriées dans le paramètre `StatsAllowedIP` serveur/proxy de l'instance cible.
- Un ensemble sélectionné de métriques internes est renvoyé par cet élément. Pour plus de détails, voir Surveillance à distance des statistiques Zabbix.

zabbix[stats,<ip>,<port>,queue,<from>,<to>]

 Renvoie les métriques internes de la file d'attente (voir `zabbix[queue,<from>,<to>]`) d'un serveur Zabbix ou d'un proxy. Si `<ip>` et `<port>` sont spécifiés, les métriques sont récupérées depuis l'instance distante ; sinon, depuis l'instance locale.
 Valeurs de retour : *objet JSON*.

Paramètres :

- **ip** - la liste des adresses IP/DNS/masques réseau des serveurs/proxy à interroger à distance (par défaut : `127.0.0.1`) ;
- **port** - le port du serveur/proxy à interroger à distance (par défaut : `10051`) ;
- **from** - retardé d'au moins (par défaut : 6 secondes) ;
- **to** - retardé d'au plus (par défaut : infini).

Commentaires :

- La requête stats ne sera acceptée que depuis les adresses répertoriées dans le paramètre `StatsAllowedIP` serveur/proxy de l'instance cible.
- Un ensemble sélectionné de métriques internes est renvoyé par cet élément. Pour plus de détails, voir Surveillance à distance des statistiques Zabbix.

> **Attention:**
> Le proxy Zabbix n'a pas connaissance des périodes de maintenance ; voir Calcul des files d'attente pendant la maintenance pour plus de détails.

zabbix[tcache,cache,<parameter>]

 Les statistiques d'efficacité du cache de la fonction de tendances de Zabbix.
 Valeurs de retour: *Integer* (pour la taille) ; *Float* (pour le pourcentage).

Paramètres:

- **parameter**:
  - *all* - nombre total de requêtes du cache (par défaut) ;
  - *hits* - accès réussis au cache ;
  - *phits* - pourcentage d'accès réussis au cache ;
  - *misses* - échecs du cache ;
  - *pmisses* - pourcentage d'échecs du cache ;
  - *items* - nombre d'éléments mis en cache ;
  - *requests* - nombre de requêtes mises en cache ;
  - *pitems* - pourcentage d'éléments mis en cache par rapport aux éléments mis en cache + requêtes. Une faible valeur de pourcentage signifie probablement que la taille du cache peut être réduite.
  - *paccessed* - pourcentage d'entrées consultées au cours des dernières 24 heures.

Commentaires:

- Cet élément n'est pas pris en charge sur le proxy Zabbix.

zabbix[triggers]

 Le nombre de déclencheurs activés dans la base de données Zabbix, avec tous les éléments de données activés sur les hôtes activés.
 Valeur de retour : *Integer*.

Commentaires :

- Cet élément n'est pas pris en charge sur le proxy Zabbix.

zabbix[uptime]

 Le temps de fonctionnement du processus du serveur ou du proxy Zabbix, en secondes.
 Valeur renvoyée : *Integer*.

zabbix[vcache,buffer,<mode>]

 Les statistiques de disponibilité du cache de valeurs Zabbix.
 Valeurs de retour : *Integer* (pour la taille) ; *Float* (pour le pourcentage).

Paramètres :

- **mode** :

  - *total* - la taille totale du buffer ;
  - *free* - la taille du buffer libre ;
  - *pfree* - le pourcentage de buffer libre ;
  - *used* - la taille du buffer utilisé ;
  - *pused* - le pourcentage de buffer utilisé.

Commentaires :

- Cet élément n'est pas pris en charge sur le proxy Zabbix.

zabbix[vcache,cache,<parameter>]

 Les statistiques d'efficacité du cache de valeurs de Zabbix.
 Valeur de retour: *Integer*. Si <parameter> est défini sur *mode*, la valeur de retour est: *0* - mode normal; *1* - mode faible consommation de mémoire.

Paramètres:

- **parameter**:
  - *requests* - le nombre total de requêtes;
  - *hits* - le nombre de hits du cache (valeurs d'historique prises depuis le cache);
  - *misses* - le nombre de misses du cache (valeurs d'historique prises depuis la base de données);
  - *mode* - le mode de fonctionnement du cache de valeurs.

Commentaires:

- Une fois le mode faible consommation de mémoire activé, le cache de valeurs restera dans cet état pendant 24 heures, même si le problème qui a déclenché ce mode est résolu plus tôt.
- Vous pouvez utiliser cette clé avec l'étape de prétraitement *Change per second* afin d'obtenir des statistiques de valeurs par seconde.
- Cet élément n'est pas pris en charge sur le proxy Zabbix.

zabbixversion

 La version du serveur ou du proxy Zabbix.
 Valeur de retour: *String*. Par exemple: `8.0.0`.

zabbix[vmware,buffer,<mode>]

 Les statistiques de disponibilité du cache VMware de Zabbix.
 Valeurs de retour : *Integer* (pour la taille) ; *Float* (pour le pourcentage).

Paramètres :

- **mode** :
  - *total* - la taille totale du tampon ;
  - *free* - la taille du tampon libre ;
  - *pfree* - le pourcentage de tampon libre ;
  - *used* - la taille du tampon utilisé ;
  - *pused* - le pourcentage de tampon utilisé.

zabbix[vps,written]

 Le nombre total de valeurs d'historique écrites dans la base de données.
 Valeur de retour : *Integer*.

zabbix[wcache,<cache>,<mode>]

 Les statistiques et la disponibilité du cache d'écriture de Zabbix.
 Valeurs de retour: *Integer* (pour le nombre/la taille); *Float* (pour le pourcentage).

Paramètres:

- **cache** - *values*, *history*, *index*, ou *trend*;
- **mode**:
  - avec le paramètre de cache *values*:
    * *all* - le nombre total de valeurs traitées par le serveur/proxy Zabbix, à l'exception des éléments non pris en charge (compteur) (par défaut);
    * *float* - le nombre de valeurs flottantes traitées (compteur);
    * *uint* - le nombre de valeurs entières non signées traitées (compteur);
    * *str* - le nombre de valeurs de caractères/chaînes traitées (compteur);
    * *log* - le nombre de valeurs de journal traitées (compteur);
    * *text* - le nombre de valeurs textuelles traitées (compteur);
    * *bin* - le nombre de valeurs binaires traitées (compteur);
    * *json* - le nombre de valeurs JSON traitées (compteur);

    - *not supported* - le nombre de fois où le traitement d'un élément a entraîné le passage de l'élément à l'état non pris en charge ou le maintien de cet état (compteur);
  - avec les paramètres de cache *history*, *index*, *trend*:
    - *pfree* - le pourcentage de mémoire tampon libre (par défaut);
    - *total* - la taille totale de la mémoire tampon;
    - *free* - la taille de la mémoire tampon libre;
    - *used* - la taille de la mémoire tampon utilisée;
    - *pused* - le pourcentage de mémoire tampon utilisée.

Commentaires:

- Le paramètre de cache `trend` n'est pas pris en charge avec le proxy Zabbix.
- Le cache d'historique est utilisé pour stocker les valeurs des éléments. Un faible nombre indique des problèmes de performance du côté de la base de données.
- Le cache d'index de l'historique est utilisé pour indexer les valeurs stockées dans le cache d'historique.
- Après que le cache d'historique est rempli puis vidé, le cache d'index de l'historique conservera encore certaines données. Ce comportement est attendu et aide le système à fonctionner plus efficacement en évitant le traitement supplémentaire nécessaire pour redimensionner constamment la mémoire.
- Le cache de tendances stocke l'agrégat de l'heure en cours pour tous les éléments qui reçoivent des données.
- Vous pouvez utiliser la clé `zabbix[wcache,values]` avec l'étape de prétraitement *Change per second* afin d'obtenir des statistiques de valeurs par seconde.

Zabbix trapper

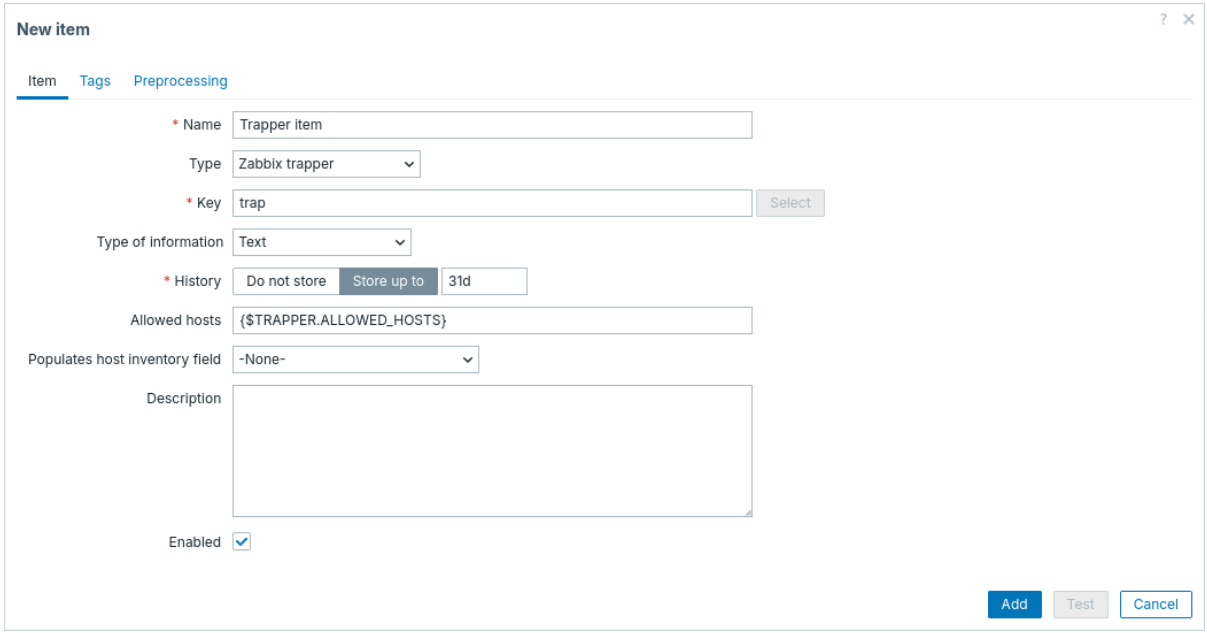
Vue d'ensemble

Les éléments trapper acceptent des données entrantes au lieu de les interroger. C'est utile pour toute donnée que vous souhaitez envoyer à Zabbix.

Configuration

Pour configurer un élément trapper :

1. Accédez à *Collecte de données* > *Hôtes*.
2. Cliquez sur *Éléments* dans la ligne de l'hôte.
3. Cliquez sur *Créer un élément*.
4. Saisissez les paramètres de l'élément dans le formulaire de configuration.



Tous les champs obligatoires sont marqués d'un astérisque rouge.

Les champs qui nécessitent des informations spécifiques pour les éléments trapper sont les suivants :

Type	Sélectionnez "Zabbix trapper".
Key	Saisissez une clé qui sera utilisée pour reconnaître l'élément lors de l'envoi de données au serveur Zabbix.
Type of information	Sélectionnez le type d'information qui correspondra au format des données envoyées.

Allowed hosts	Liste d'adresses IP séparées par des virgules (éventuellement en notation CIDR) ou de noms DNS. La valeur par défaut est la macro utilisateur `{$TRAPPER.ALLOWED_HOSTS}`. Par défaut, `{$TRAPPER.ALLOWED_HOSTS}` vaut `127.0.0.1,::1`. Vous pouvez soit saisir directement les adresses dans le champ, soit modifier la valeur de la macro. Si le champ est laissé vide, les connexions entrantes seront refusées depuis tous les hôtes. Si une valeur est spécifiée, les connexions entrantes ne seront acceptées que depuis les hôtes répertoriés ici. Si la prise en charge d'IPv6 est activée, alors '127.0.0.1', '::127.0.0.1', '::ffff:127.0.0.1' sont traités de la même manière et '::/0' autorisera toute adresse IPv4 ou IPv6. '0.0.0.0/0' peut être utilisé pour autoriser toute adresse IPv4. Notez que les "adresses IPv6 compatibles IPv4" (préfixe 0000::/96) sont prises en charge mais sont obsolètes selon RFC4291. Exemple : 127.0.0.1, 192.168.1.0/24, 192.168.3.1-255, 192.168.1-10.1-255, ::1,2001:db8::/32, mysqlserver1, zabbix.example.com, {HOST.HOST} Les espaces, les macros utilisateur et les macros d'hôte {HOST.HOST}, {HOST.NAME}, {HOST.IP}, {HOST.DNS}, {HOST.CONN} sont pris en charge.

Note:
Avant d'envoyer des valeurs, vous devrez peut-être attendre jusqu'à 60 secondes après l'enregistrement de l'élément, le temps que le serveur Zabbix prenne en compte les modifications issues de la mise à jour du cache de configuration.

Envoi de données

L'envoi de données au serveur Zabbix ou au proxy est possible à l'aide de l'utilitaire Zabbix sender ou du protocole Zabbix sender. L'envoi de données au serveur Zabbix est également possible à l'aide de la méthode API `history.push`.

Zabbix sender

Pour envoyer des données au serveur Zabbix ou au proxy à l'aide de l'utilitaire Zabbix sender, vous pouvez exécuter la commande suivante pour envoyer la "test value" :

```
zabbix_sender -z <server IP address> -p 10051 -s "New host" -k trap -o "test value"
```

Pour envoyer la "test value", les options de commande suivantes sont utilisées :

- `-z` pour spécifier l'adresse IP du serveur Zabbix
- `-p` pour spécifier le numéro de port du serveur Zabbix (10051 par défaut)
- `-s` pour spécifier l'hôte (assurez-vous d'utiliser le nom technique plutôt que le nom d'hôte visible)
- `-k` pour spécifier la clé de l'élément configuré dans l'élément trapper
- `-o` pour spécifier la valeur à envoyer

Attention:
Le processus Zabbix trapper n'étend pas les macros utilisées dans la clé de l'élément pour vérifier l'existence de la clé d'élément correspondante pour l'hôte ciblé.

Pour plus d'informations sur la communication entre Zabbix sender et le serveur Zabbix ou le proxy, consultez *Zabbix sender protocol*.

history.push

Pour envoyer des données au serveur Zabbix à l'aide de la méthode d'API `history.push`, vous pouvez effectuer la requête HTTP POST suivante contenant quelques valeurs de test :

```
curl --request POST \
  --url 'https://example.com/zabbix/api_jsonrpc.php' \
  --header 'Authorization: Bearer 0424bd59b807674191e7d77572075f33' \
  --header 'Content-Type: application/json-rpc' \
  --data '{"jsonrpc":"2.0","method":"history.push","params":[{"itemid":10600,"value":"test value 1"},{"ite
```

Si la requête est correcte, la réponse renvoyée par l'API peut ressembler à ceci :

```
{
    "jsonrpc": "2.0",
    "result": {
        "response": "success",
        "data": [
            {
                "itemid": "10600"
            },
            {
                "itemid": "10601",
                "error": "Item is disabled."
            },
            {
                "error": "No permissions to referred object or it does not exist."
            }
        ]
    },
    "id": 1
}
```


Les erreurs dans les données de réponse indiquent que l'envoi de données pour des éléments spécifiques n'a pas passé la validation du serveur Zabbix. Cela peut se produire pour les raisons suivantes :

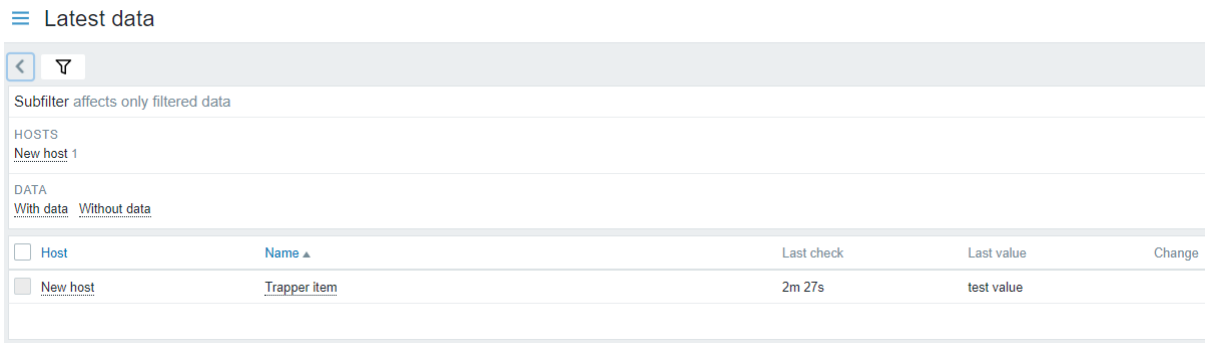
- l'utilisateur qui envoie les données n'a pas l'autorisation *read* sur l'hôte de l'élément ;
- l'hôte est désactivé ou en maintenance sans collecte de données ;
- l'élément n'existe pas ou n'est pas encore inclus dans le cache de configuration du serveur ;
- l'élément est désactivé ou son type est autre que Zabbix trapper ou HTTP agent (avec le trapping activé) ;
- l'adresse IP ou le DNS de l'utilisateur n'est pas défini dans la liste *Allowed hosts* de l'élément ;
- un autre élément possède une valeur avec un horodatage en double au niveau de la nanoseconde.

L'absence d'erreurs indique que les valeurs envoyées ont été acceptées pour traitement, ce qui inclut le prétraitement (le cas échéant), le traitement des déclencheurs et l'enregistrement dans la base de données. Notez toutefois que le traitement d'une valeur acceptée peut également échouer (par exemple, lors du preprocessing), ce qui entraîne le rejet de la valeur.

Pour plus d'informations sur l'utilisation de l'API Zabbix, consultez API.

Affichage des données

Une fois les données envoyées, vous pouvez accéder à *Monitoring* > *Dernières données* pour voir le résultat :



Note:
Si une seule valeur numérique est envoyée, le graphique des données affichera une ligne horizontale à gauche et à droite du point temporel de la valeur.

Vérification externe

Aperçu

Une vérification externe est une vérification exécutée par le serveur Zabbix en lançant un script shell ou un binaire. Cependant, lorsque les hôtes sont surveillés par un proxy Zabbix, les vérifications externes sont exécutées par le proxy.

Les vérifications externes ne nécessitent aucun agent en cours d'exécution sur un hôte surveillé.

La syntaxe de la clé d'élément est la suivante :

```
script[<parameter1>,<parameter2>,...]
```

Où :

ARGUMENT	DÉFINITION
script	Nom d'un script shell ou d'un binaire.
parameter(s)	Paramètres de ligne de commande facultatifs.

Si vous ne souhaitez transmettre aucun paramètre au script, vous pouvez utiliser :

```
script[] ou
script
```

Le serveur Zabbix ou le proxy recherchera le répertoire spécifié pour les scripts externes et exécutera la commande (voir le paramètre `ExternalScripts` dans le fichier de configuration du serveur/proxy Zabbix). La commande sera exécutée sous le même utilisateur que le serveur/proxy Zabbix, donc toute permission d'accès ou variable d'environnement doit, si nécessaire, être gérée dans un script d'enveloppe. Les permissions sur la commande doivent également autoriser cet utilisateur à l'exécuter. Seules les commandes situées dans le répertoire spécifié sont disponibles à l'exécution.

Warning:
N'utilisez pas excessivement les vérifications externes, car chaque script nécessite le lancement d'un processus fork par le serveur/proxy Zabbix, et l'exécution d'un grand nombre de scripts peut réduire considérablement les performances de Zabbix.

Exemple d'utilisation

Exécution du script `check_oracle.sh` avec le premier paramètre `'-h'`. Le deuxième paramètre sera remplacé par l'adresse IP ou le nom DNS, selon la sélection dans les propriétés de l'hôte.

```
check_oracle.sh["-h","{HOST.CONN}"]
```

En supposant que l'hôte est configuré pour utiliser l'adresse IP, le serveur/proxy Zabbix exécutera :

```
check_oracle.sh '-h' '192.168.1.4'
```

Résultat de la vérification externe

La valeur de retour d'une vérification externe correspond à la sortie standard ainsi qu'à la sortie d'erreur standard produites par la vérification.

Attention:
Un élément qui renvoie du texte (type d'information caractère, journal ou texte) ne deviendra pas non pris en charge en cas de sortie sur la sortie d'erreur standard.

La valeur de retour est limitée à 16 Mo (y compris les espaces de fin qui sont tronqués) ; les limites de la base de données s'appliquent également.

Si le script demandé est introuvable ou si le serveur/proxy Zabbix n'a pas les autorisations nécessaires pour l'exécuter, l'élément deviendra non pris en charge et un message d'erreur correspondant sera affiché.

En cas de dépassement du délai d'attente, l'élément deviendra non pris en charge, un message d'erreur correspondant sera affiché et le processus créé pour le script sera terminé.

Moniteur de base de données

Vue d'ensemble

Le type d'élément *Database monitor* dans l'interface Zabbix est utilisé pour la supervision ODBC (vérifications ODBC).

ODBC est une API middleware du langage de programmation C permettant d'accéder aux systèmes de gestion de bases de données (DBMS). Le concept ODBC a été développé par Microsoft, puis porté ultérieurement vers d'autres plateformes.

Zabbix peut interroger n'importe quelle base de données prise en charge par ODBC. Pour cela, Zabbix ne se connecte pas directement aux bases de données, mais utilise l'interface ODBC et les pilotes configurés dans ODBC. Cela permet une supervision plus efficace de différentes bases de données à des fins multiples (par exemple, la vérification de files d'attente spécifiques de base de données, de statistiques d'utilisation, etc.).

Zabbix prend en charge unixODBC, qui est l'une des implémentations open source de l'API ODBC les plus couramment utilisées.

Attention:
Voir aussi : problèmes connus pour les vérifications ODBC.

Installation de unixODBC

La méthode recommandée pour installer unixODBC consiste à utiliser les dépôts de paquets par défaut du système d'exploitation Linux. Dans les distributions Linux les plus populaires, unixODBC est inclus par défaut dans le dépôt de paquets. Si les paquets ne sont pas disponibles, les fichiers source peuvent être obtenus sur la page d'accueil de unixODBC: http://www.unixodbc.org/download.html.

Pour installer unixODBC, utilisez le gestionnaire de paquets du système de votre choix:

```
##### Pour les systèmes Ubuntu/Debian:
apt install unixodbc unixodbc-dev

##### Pour les systèmes basés sur RedHat/Fedora:
dnf install unixODBC unixODBC-devel

##### Pour les systèmes basés sur SUSE:
zypper in unixODBC-devel
```

Attention:
Le paquet `unixodbc-dev` ou `unixODBC-devel` est nécessaire pour compiler Zabbix avec la prise en charge de unixODBC.
Pour activer la prise en charge ODBC, Zabbix doit être compilé avec l'option de configuration suivante:

`--with-unixodbc[=ARG] # Utiliser le pilote ODBC avec le paquet unixODBC.`

Installation des pilotes unixODBC

Le pilote de base de données unixODBC doit être installé pour la base de données qui sera surveillée. Pour obtenir la liste des bases de données et des pilotes pris en charge, consultez la page d'accueil de unixODBC : http://www.unixodbc.org/drivers.html.

Note:
Dans certaines distributions Linux, les pilotes de base de données sont inclus dans les dépôts de paquets.

MySQL

Pour installer le pilote de base de données MySQL unixODBC, utilisez le gestionnaire de paquets du système de votre choix :

```
##### Pour les systèmes Ubuntu/Debian :
apt install odbc-mariadb

##### Pour les systèmes basés sur RedHat/Fedora :
dnf install mariadb-connector-odbc

##### Pour les systèmes basés sur SUSE :
zypper install mariadb-connector-odbc
```

Pour installer le pilote de base de données sans gestionnaire de paquets, veuillez consulter la documentation MySQL pour `mysql-connector-odbc`, ou la documentation MariaDB pour `mariadb-connector-odbc`.

PostgreSQL

Pour installer le pilote de base de données PostgreSQL unixODBC, utilisez le gestionnaire de paquets du système de votre choix :

```
##### Pour les systèmes Ubuntu/Debian :
apt install odbc-postgresql

##### Pour les systèmes basés sur RedHat/Fedora :
dnf install postgresql-odbc

##### Pour les systèmes basés sur SUSE :
zypper install psqlODBC
```

Pour installer le pilote de base de données sans gestionnaire de paquets, veuillez consulter la documentation PostgreSQL.

Oracle

Pour installer le pilote de base de données unixODBC, veuillez consulter la documentation Oracle.

MSSQL

Pour installer le pilote de base de données MSSQL unixODBC, utilisez le gestionnaire de paquets du système de votre choix :

```
##### Pour les systèmes Ubuntu/Debian :
apt install tdsodbc

##### Pour les systèmes basés sur RedHat/Fedora (paquets EPEL : https://docs.fedoraproject.org/en-US/epel/
dnf install epel-release
dnf install freetds

##### Pour les systèmes basés sur SUSE :
zypper install libtdsodbc0
```

Pour installer le pilote de base de données sans gestionnaire de paquets, veuillez consulter le guide de l'utilisateur FreeTDS.

Configuration de unixODBC

Pour configurer unixODBC, vous devez modifier les fichiers `odbcinst.ini` et `odbc.ini`. Vous pouvez vérifier l'emplacement de ces fichiers en exécutant la commande suivante :

```
odbcinst -j
```

Le résultat de la commande doit contenir des informations similaires à ce qui suit :

```
unixODBC 2.3.9
DRIVERS............: /etc/odbcinst.ini
SYSTEM DATA SOURCES: /etc/odbc.ini
FILE DATA SOURCES..: /etc/ODBCDataSources
```

odbcinst.ini

Le fichier `odbcinst.ini` répertorie les pilotes de base de données ODBC installés. Si `odbcinst.ini` est absent, il est nécessaire de le créer manuellement.

```
[TEST_MYSQL]
Description=ODBC for MySQL
Driver=/usr/lib/libmyodbc5.so
FileUsage=1
```

Parameter	Description
TEST_MYSQL	Nom du pilote de base de données.
Description	Description du pilote de base de données.
Driver	Emplacement de la bibliothèque du pilote de base de données.
FileUsage	Détermine si le pilote de base de données prend en charge la connexion à un serveur de base de données sans prise en charge de l'accès aux fichiers locaux (0) ; prend en charge la lecture des données à partir de fichiers (1) ; prend en charge l'écriture des données dans des fichiers (2).
Threading	Niveau de sérialisation des threads. Pris en charge pour PostgreSQL. Depuis la version 1.6, si le gestionnaire de pilotes est compilé avec la prise en charge des threads, vous pouvez ajouter une autre entrée de pilote.

odbc.ini

Le fichier `odbc.ini` est utilisé pour configurer les sources de données. Notez que la liste des paramètres pris en charge dépend du pilote de base de données (par exemple, les bases de données Oracle peuvent utiliser ServerName au lieu de Server, etc.).

```
[TEST_MYSQL]
Description=MySQL Test Database
Driver=mysql
Server=127.0.0.1
User=root
Password=
Port=3306
Socket=
Database=zabbix
```

Parameter	Description
TEST_MYSQL	Nom de la source de données (DSN).
Description	Description de la source de données.
Driver	Nom du pilote de base de données (tel que spécifié dans `odbcinst.ini`).
Server	IP/DNS du serveur de base de données.
User	Utilisateur de la base de données pour la connexion.
Password	Mot de passe de l'utilisateur de la base de données.
Port	Port de connexion à la base de données.
Socket	Socket de connexion à la base de données.
Database	Nom de la base de données.

Pour d'autres options possibles de paramètres de configuration, consultez la documentation MySQL.

Le fichier `odbc.ini` pour PostgreSQL peut contenir des paramètres supplémentaires :

```
[TEST_PSQL]
Description=Base de données de test PostgreSQL
Driver=postgresql
Username=zbx_test
Password=zabbix
Servername=127.0.0.1
Database=zabbix
Port=5432
ReadOnly=No
Protocol=8.0+
ShowOidColumn=No
FakeOidIndex=No
RowVersioning=No
ShowSystemTables=No
Fetch=Yes
BoolsAsChar=Yes
SSLmode=Require
ConnSettings=
```

Paramètre	Description
ReadOnly	Indique si la connexion à la base de données autorise uniquement les opérations de lecture (requêtes `SELECT`) et limite les modifications (instructions `INSERT`, `UPDATE` et `DELETE`) ; utile dans les scénarios où les données doivent rester inchangées.
Protocol	Version du protocole backend PostgreSQL (ignorée lors de l'utilisation de connexions SSL).
ShowOidColumn	Indique s'il faut inclure l'Object ID (OID) dans SQLColumns.
FakeOidIndex	Indique s'il faut créer un faux index unique sur l'OID.
RowVersioning	Indique s'il faut permettre aux applications de détecter si des données ont été modifiées par d'autres utilisateurs pendant que vous tentez de mettre à jour une ligne. Notez que ce paramètre peut accélérer le processus de mise à jour, car, pour mettre à jour une ligne, il n'est pas nécessaire de spécifier chaque colonne individuellement dans la clause `WHERE`.
ShowSystemTables	Indique si le pilote de base de données doit traiter les tables système comme des tables ordinaires dans SQLTables ; utile pour l'accessibilité, en permettant la visibilité sur les tables système.
Fetch	Indique si le pilote doit utiliser automatiquement declare cursor/fetch pour gérer les instructions `SELECT` et maintenir un cache de 100 lignes.
BoolsAsChar	Contrôle le mappage des types booléens. S'il est défini sur "Yes", les booléens sont mappés sur `SQL_CHAR` ; sinon, ils sont mappés sur `SQL_BIT`.
SSLmode	Indique le mode SSL pour la connexion.
ConnSettings	Paramètres supplémentaires envoyés au backend lors de la connexion.

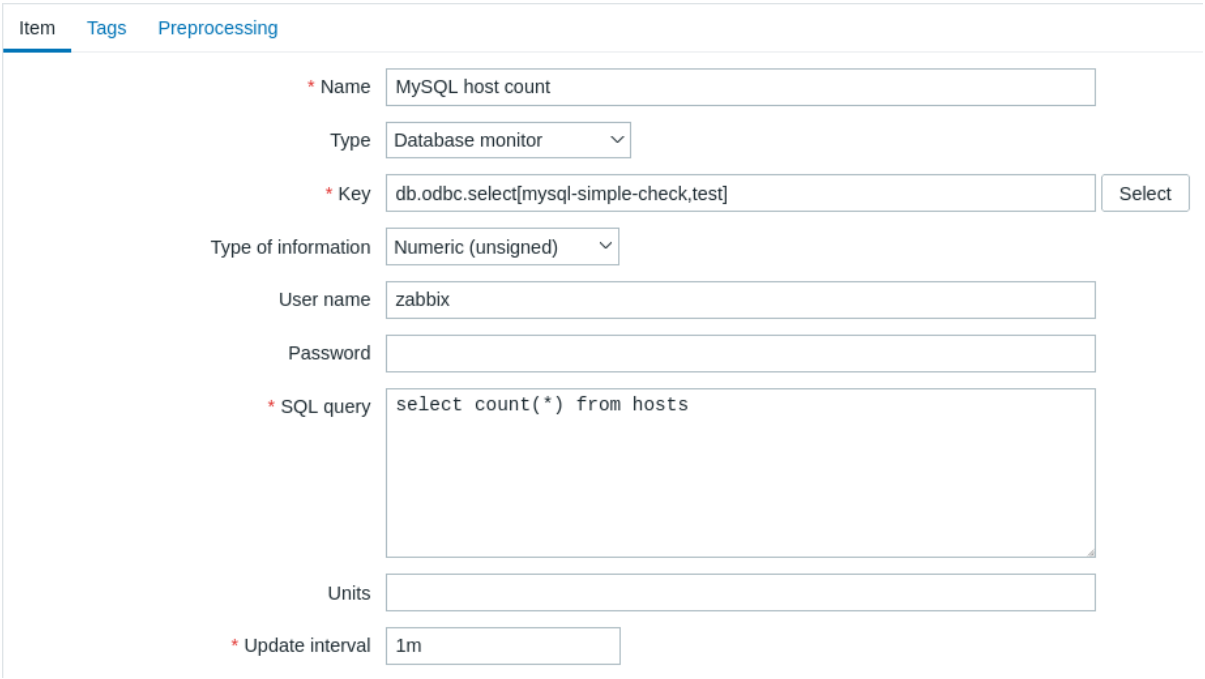
Test de la connexion ODBC

Pour vérifier si la connexion ODBC fonctionne correctement, vous pouvez utiliser l'utilitaire `isql` (inclus dans le paquet `unixODBC`) :

```
isql test
+---------------------------------------+
| Connected!                            |
|                                       |
| sql-statement                         |
| help [tablename]                      |
| quit                                  |
|                                       |
+---------------------------------------+
```

Configuration d'un élément dans l'interface Zabbix

Configurez un élément de **surveillance de base de données**.



Tous les champs de saisie obligatoires sont marqués d'un astérisque rouge.

Pour les éléments de surveillance de base de données, vous devez spécifier :

Type	Sélectionnez ici "Database monitor".
Key	Saisissez l'une des clés d'élément prises en charge : **db.odbc.select[]** - cet élément renvoie une seule valeur (la première colonne de la première ligne du résultat de la requête SQL) ; **db.odbc.get[]** - cet élément renvoie plusieurs lignes/colonnes au format JSON ; **db.odbc.discovery[]** - cet élément renvoie des données de découverte de bas niveau.
User name	Saisissez le nom d'utilisateur de la base de données (jusqu'à 255 caractères). Ce paramètre est facultatif si le nom d'utilisateur de la base de données est spécifié dans le fichier `odbc.ini`. Si une chaîne de connexion est utilisée et que le champ *User name* n'est pas vide, il est alors ajouté à la chaîne de connexion sous la forme `UID=<user>`.
Password	Saisissez le mot de passe de l'utilisateur de la base de données (jusqu'à 255 caractères). Ce paramètre est facultatif si le mot de passe est spécifié dans le fichier `odbc.ini`. Si une chaîne de connexion est utilisée et que le champ *Password* n'est pas vide, il est alors ajouté à la chaîne de connexion sous la forme `PWD=<password>`. Les caractères spéciaux sont pris en charge dans ce champ. Pour toutes les bases de données sauf Oracle, les mots de passe contenant `}` doivent être placés entre accolades `{}`. Le mot de passe sera ajouté à la chaîne de connexion après le nom d'utilisateur, par exemple `UID=<username>;PWD=P?;)*word`. Pour tester la chaîne obtenue, vous pouvez exécuter la commande suivante : `isql -v -k 'Driver=libmaodbc.so;Database=zabbix;UID=zabbix;PWD=P?;)*word'`
SQL query	Saisissez la requête SQL. Notez qu'avec `db.odbc.select[]`, la requête doit renvoyer une seule valeur.

Type of information	Sélectionnez ici le type d'information qui sera renvoyé par la requête. Si le type d'information est sélectionné incorrectement, l'élément deviendra non pris en charge.

Notes importantes

- Les éléments de surveillance de base de données deviendront non pris en charge si aucun processus *odbc poller* n'est démarré dans la configuration du serveur ou du proxy. Pour activer les ODBC pollers, définissez le paramètre `StartODBCPollers` dans le fichier de configuration du serveur Zabbix ou, pour les vérifications effectuées par le proxy, dans le fichier de configuration du proxy Zabbix.
- La valeur du paramètre *Timeout* dans le formulaire de configuration de l'élément est utilisée comme délai d'attente de connexion ODBC et délai d'attente d'exécution de la requête. Notez que ces paramètres de délai d'attente peuvent être ignorés si le pilote ODBC installé ne les prend pas en charge.
- La commande SQL doit renvoyer un jeu de résultats, comme toute requête utilisant l'instruction `select`. La syntaxe de la requête dépendra du SGBDR qui la traitera. La syntaxe de la requête vers une procédure de stockage doit commencer par le mot-clé `call`.

Détails de la clé d'élément

Les paramètres sans chevrons sont obligatoires. Les paramètres marqués par des chevrons **< >** sont facultatifs.

db.odbc.select[<description courte unique>,<dsn>,<chaîne de connexion>]

 Renvoie une valeur, c'est-à-dire la première colonne de la première ligne du résultat de la requête SQL.
 Valeur de retour : selon la requête SQL.

Paramètres :

- **description courte unique** - une description courte unique permettant d'identifier l'élément (à utiliser dans les déclencheurs, etc.) ;
- **dsn** - le nom de la source de données (tel que spécifié dans `odbc.ini`) ;
- **chaîne de connexion** - la chaîne de connexion (peut contenir des arguments spécifiques au pilote).

Commentaires :

- Bien que `dsn` et `chaîne de connexion` soient des paramètres facultatifs, au moins l'un des deux est requis ; si les deux sont définis, `dsn` sera ignoré.
- Si une requête renvoie plus d'une colonne, seule la première colonne est lue. Si une requête renvoie plus d'une ligne, seule la première ligne est lue.

db.odbc.get[<unique short description>,<dsn>,<connection string>]

 Transforme le résultat de la requête SQL en un tableau JSON.
 Valeur de retour: *objet JSON*.

Paramètres:

- **unique short description** - une brève description unique permettant d'identifier l'élément (pour une utilisation dans les déclencheurs, etc.) ;
- **dsn** - le nom de la source de données (tel que spécifié dans `odbc.ini`) ;
- **connection string** - la chaîne de connexion (peut contenir des arguments spécifiques au pilote).

Commentaires:

- Bien que `dsn` et `connection string` soient des paramètres facultatifs, au moins l'un d'eux est requis ; si les deux sont définis, `dsn` sera ignoré.
- Plusieurs lignes/colonnes au format JSON peuvent être renvoyées. Cet élément peut être utilisé comme élément maître qui collecte toutes les données en un seul appel système, tandis que le prétraitement JSONPath peut être utilisé dans les éléments dépendants pour extraire des valeurs individuelles. Pour plus d'informations, voir un exemple du format renvoyé, utilisé dans la découverte de bas niveau.

Exemple:

```
# Connection for MySQL ODBC driver 5:
db.odbc.get[MySQL example,,"Driver=/usr/local/lib/libmyodbc5a.so;Database=master;Server=127.0.0.1;Port=330
```

db.odbc.discovery[<description courte unique>,<dsn>,<chaîne de connexion>]

 Transforme le résultat de la requête SQL en un tableau JSON, utilisé pour la découverte de bas niveau. Les noms de colonnes du résultat de la requête sont convertis en noms de macros de découverte de bas niveau, associés aux valeurs des champs découverts. Ces macros peuvent être utilisées lors de la création de prototypes d'éléments, de déclencheurs, etc.
 Valeur de retour : *objet JSON*.

Paramètres :

- **description courte unique** - une description courte unique permettant d'identifier l'élément (à utiliser dans les déclencheurs, etc.) ;
- **dsn** - le nom de la source de données (tel que spécifié dans `odbc.ini`) ;
- **chaîne de connexion** - la chaîne de connexion (peut contenir des arguments spécifiques au pilote).

Commentaires :

- Bien que `dsn` et `chaîne de connexion` soient des paramètres facultatifs, au moins l'un des deux est requis ; si les deux sont définis, `dsn` sera ignoré.

Messages d'erreur

Les messages d'erreur ODBC sont structurés en champs afin de fournir des informations détaillées. Par exemple, un message d'erreur peut ressembler à ceci :

```
Cannot execute ODBC query: [SQL_ERROR]:[42601][7][ERROR: syntax error at or near ";"; Error while executin
```

- "`Cannot execute ODBC query`" - message Zabbix
- "`[SQL_ERROR]`" - code de retour ODBC
- "`[42601]`" - SQLState
- "`[7]`" - code d'erreur natif
- "`[ERROR: syntax error at or near ";"; Error while executing the query]`" - message d'erreur natif

Notez que la longueur du message d'erreur est limitée à 2048 octets ; le message peut donc être tronqué. S'il y a plus d'un enregistrement de diagnostic ODBC, Zabbix essaie de les concaténer (séparés par `|`) dans la mesure permise par la limite de longueur.

agent HTTP

Vue d'ensemble

Ce type d'élément permet l'interrogation des données via le protocole HTTP/HTTPS. La réception de données est également possible à l'aide de l'utilitaire Zabbix sender ou du protocole Zabbix sender (pour envoyer des données au serveur Zabbix ou au proxy), ou à l'aide de la méthode API `history.push` (pour envoyer des données au serveur Zabbix).

Les vérifications des éléments HTTP sont exécutées par le serveur Zabbix. Cependant, lorsque les hôtes sont surveillés par un proxy Zabbix, les vérifications des éléments HTTP sont exécutées par le proxy.

Les vérifications des éléments HTTP ne nécessitent aucun agent en cours d'exécution sur un hôte surveillé.

HTTP agent prend en charge HTTP et HTTPS. Zabbix suivra éventuellement les redirections (voir l'option *Suivre les redirections* ci-dessous). Le nombre maximal de redirections est codé en dur à 10 (en utilisant l'option cURL `CURLOPT_MAXREDIRS`).

> **Attention:**
> Le serveur Zabbix/proxy doit être initialement configuré avec la prise en charge de cURL (libcurl).

Les vérifications HTTP sont exécutées de manière asynchrone - il n'est pas nécessaire de recevoir la réponse à une requête avant que d'autres vérifications ne soient lancées. La résolution DNS est également asynchrone.

La concurrence maximale des vérifications asynchrones est de 1000 (définie par MaxConcurrentChecksPerPoller).

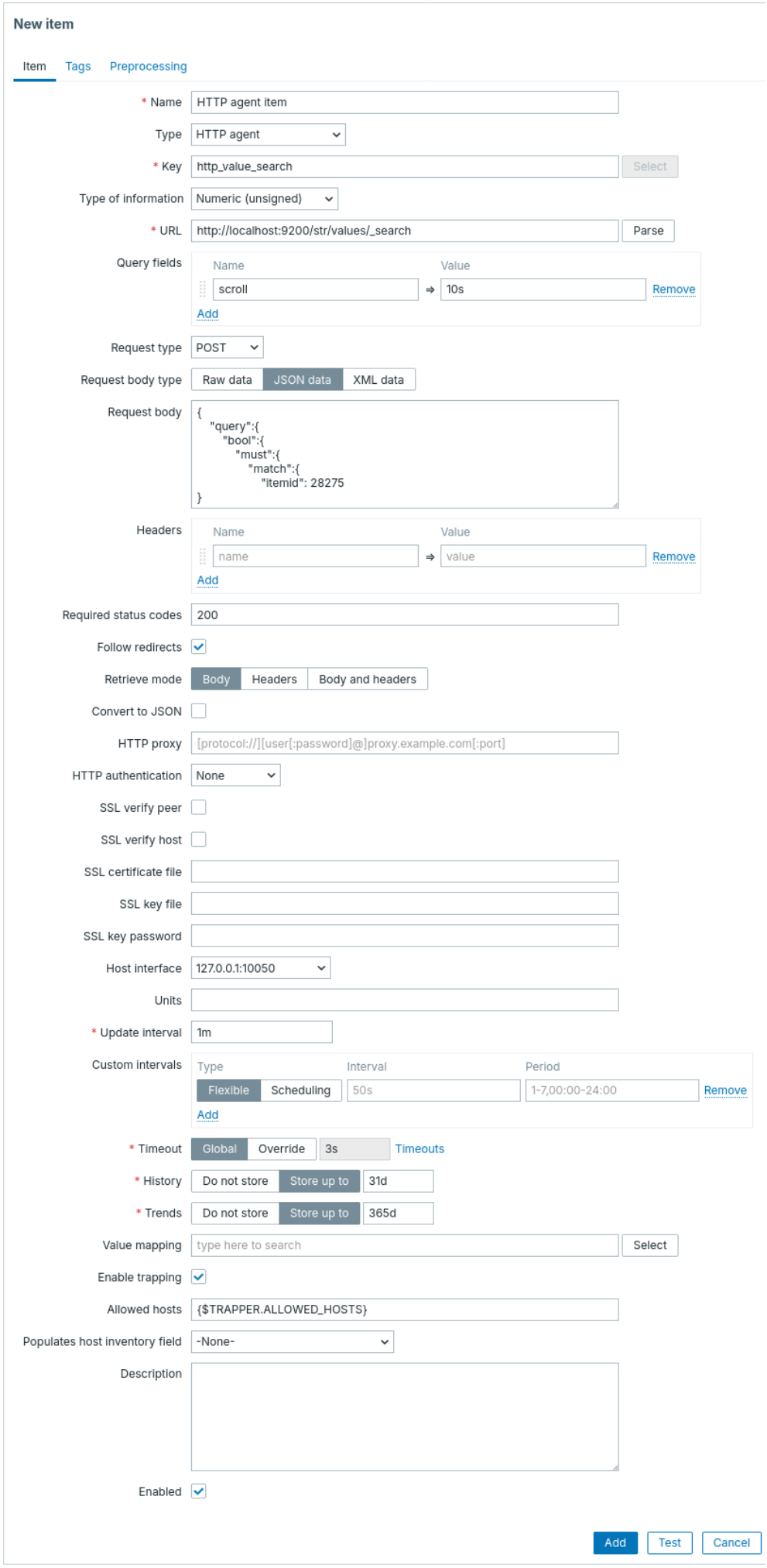
Le nombre de pollers HTTP agent asynchrones est défini par le paramètre StartHTTPAgentPollers.

La fonctionnalité cURL des connexions persistantes a été ajoutée aux vérifications HTTP agent depuis Zabbix 7.0.

Configuration

Pour configurer un élément HTTP :

1. Accédez à *Collecte de données* > *Hôtes*.
2. Cliquez sur *Éléments* dans la ligne du hôte.
3. Cliquez sur *Créer un élément*.
4. Saisissez les paramètres de l'élément dans le formulaire.


New item
Item
Tags
Preprocessing
* Name
HTTP agent item
Type
HTTP agent
* Key
http_value_search
Select
Type of information
Numeric (unsigned)
* URL
http://localhost:9200/str/values/_search
Parse
Query fields
Name
Value
scroll
10s
Remove
Add
Request type
POST
Request body type
Raw data
JSON data
XML data
Request body
{
"query":{
"bool":{
"must":{
"match":{
"itemid": 28275
}
Headers
Name
Value
name
value
Remove
Add
Required status codes
200
Follow redirects
Retrieve mode
Body
Headers
Body and headers
Convert to JSON
HTTP proxy
[protocol://][user[:password]@]proxy.example.com[:port]
HTTP authentication
None
SSL verify peer
SSL verify host
SSL certificate file
SSL key file
SSL key password
Host interface
127.0.0.1:10050
Units
* Update interval
1m
Custom intervals
Type
Interval
Period
Flexible
Scheduling
50s
1-7,00:00-24:00
Remove
Add
* Timeout
Global
Override
3s
Timeouts
* History
Do not store
Store up to
31d
* Trends
Do not store
Store up to
365d
Value mapping
type here to search
Select
Enable trapping
Allowed hosts
{$TRAPPER.ALLOWED_HOSTS}
Populates host inventory field
-None-
Description
Enabled
Add
Test
Cancel

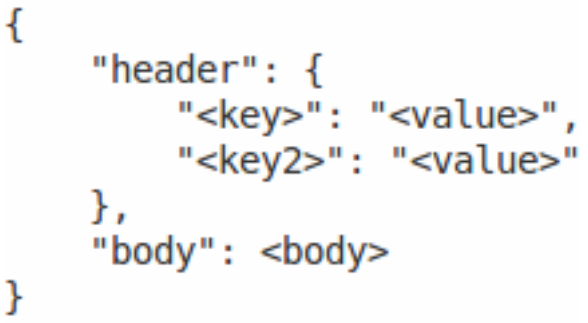
Tous les champs obligatoires sont marqués d'un astérisque rouge.

Les champs qui nécessitent des informations spécifiques aux éléments HTTP sont les suivants :

Paramètre	Description
Type	Sélectionnez **HTTP agent** ici.
Clé	Saisissez une clé d'élément unique.

URL	URL à laquelle se connecter pour récupérer les données. Par exemple :
	`https://www.example.com`
	`http://www.example.com/download`
	Les noms de domaine peuvent être spécifiés avec des caractères Unicode. Ils sont automatiquement convertis en punycode ASCII lors de l'exécution de la vérification HTTP.
	Le bouton *Analyser* peut être utilisé pour séparer les champs de requête facultatifs (tels que `?key=value&limit=10`) de l'URL, en déplaçant les attributs et les valeurs vers *Champs de requête* pour un encodage automatique dans l'URL.
	Pour des raisons de sécurité, n'incluez pas de noms d'utilisateur ou de mots de passe réels dans les URL. Utilisez plutôt des macros utilisateur afin d'éviter d'exposer les identifiants.
	Limité à 2048 caractères.
	Macros prises en charge : `{HOST.IP}`, `{HOST.CONN}`, `{HOST.DNS}`, `{HOST.HOST}`, `{HOST.NAME}`, `{HOST.PORT}`, `{ITEM.ID}`, `{ITEM.KEY}`, `{ITEM.KEY.ORIG}`, macros utilisateur, macros de découverte de bas niveau.
	Cette option définit l'option cURL `CURLOPT_URL`.

Champs de requête	Variables pour l'URL (voir ci-dessus). Spécifiées sous forme de paires attribut-valeur. Les valeurs sont automatiquement encodées dans l'URL. Les valeurs provenant de macros sont résolues, puis automatiquement encodées dans l'URL. Macros prises en charge : `{HOST.IP}`, `{HOST.CONN}`, `{HOST.DNS}`, `{HOST.HOST}`, `{HOST.NAME}`, `{HOST.PORT}`, `{ITEM.ID}`, `{ITEM.KEY}`, `{ITEM.KEY.ORIG}`, macros utilisateur, macros de découverte de bas niveau. Cette option définit l'option cURL `CURLOPT_URL`.
Type de requête	Sélectionnez le type de méthode de requête : *GET*, *POST*, *PUT* ou *HEAD*.
Type de corps de la requête	Sélectionnez le type de corps de la requête : **Données brutes** - corps personnalisé de la requête HTTP ; les macros sont remplacées, mais aucun encodage n'est effectué. **Données JSON** - corps de la requête HTTP au format JSON. Les macros peuvent être utilisées comme chaîne, nombre, true et false ; les macros utilisées comme chaînes doivent être placées entre guillemets doubles. Les valeurs provenant de macros sont résolues, puis automatiquement échappées. Si « Content-Type » n'est pas spécifié dans les en-têtes, la valeur par défaut sera « Content-Type: application/json ». **Données XML** - corps de la requête HTTP au format XML. Les macros peuvent être utilisées comme nœud de texte, attribut ou section CDATA. Les valeurs provenant de macros sont résolues, puis automatiquement échappées dans un nœud de texte et un attribut. Si « Content-Type » n'est pas spécifié dans les en-têtes, la valeur par défaut sera « Content-Type: application/xml ».
Corps de la requête	Saisissez le corps de la requête. Macros prises en charge : `{HOST.IP}`, `{HOST.CONN}`, `{HOST.DNS}`, `{HOST.HOST}`, `{HOST.NAME}`, `{HOST.PORT}`, `{ITEM.ID}`, `{ITEM.KEY}`, `{ITEM.KEY.ORIG}`, macros utilisateur, macros de découverte de bas niveau.
En-têtes	En-têtes HTTP personnalisés qui seront envoyés lors de l'exécution d'une requête. Spécifiés sous forme de paires attribut-valeur. Macros prises en charge : `{HOST.IP}`, `{HOST.CONN}`, `{HOST.DNS}`, `{HOST.HOST}`, `{HOST.NAME}`, `{HOST.PORT}`, `{ITEM.ID}`, `{ITEM.KEY}`, `{ITEM.KEY.ORIG}`, macros utilisateur, macros de découverte de bas niveau. Cette option définit l'option cURL `CURLOPT_HTTPHEADER`.
Codes d'état requis	Liste des codes d'état HTTP attendus. Si Zabbix reçoit un code qui ne figure pas dans la liste, l'élément deviendra non pris en charge. Si la liste est vide, aucune vérification n'est effectuée. Par exemple : 200,201,210-299. Macros prises en charge dans la liste : macros utilisateur, macros de découverte de bas niveau. Cette option utilise l'option cURL `CURLINFO_RESPONSE_CODE`.
Suivre les redirections	Cochez la case pour suivre les redirections HTTP. Cette option définit l'option cURL `CURLOPT_FOLLOWLOCATION`.

Paramètre	Description
Mode de récupération	Sélectionnez la partie de la réponse qui doit être récupérée : **Corps** - corps uniquement **En-têtes** - en-têtes uniquement **Corps et en-têtes** - corps et en-têtes
Convertir en JSON	Les en-têtes sont enregistrés sous forme de paires attribut-valeur sous la clé « header » si le *Mode de récupération* n'est pas défini sur *Corps*. Si « Content-Type: application/json » est détecté, le corps est enregistré comme un objet ; sinon, il est stocké comme une chaîne, par exemple : 
Proxy HTTP	Vous pouvez spécifier un proxy HTTP à utiliser, au format `[protocol://][username[:password]@]proxy.example.com[:port]`. Le préfixe facultatif `protocol://` peut être utilisé pour spécifier d'autres protocoles de proxy (par exemple, `https`, `socks4`, `socks5` ; voir la documentation ; la prise en charge du préfixe de protocole a été ajoutée dans cURL 7.21.7). Si aucun protocole n'est spécifié, le proxy sera traité comme un proxy HTTP. Si vous spécifiez un protocole incorrect, la connexion échouera et l'élément deviendra non pris en charge. Par défaut, le port 1080 sera utilisé. S'il est spécifié, le proxy remplacera les variables d'environnement liées au proxy telles que `http_proxy` et `HTTPS_PROXY`. S'il n'est pas spécifié, le proxy ne remplacera pas les variables d'environnement liées au proxy. La valeur saisie est transmise « telle quelle », sans aucune vérification de cohérence. Notez que seule l'authentification simple est prise en charge avec un proxy HTTP. Macros prises en charge : `{HOST.IP}`, `{HOST.CONN}`, `{HOST.DNS}`, `{HOST.HOST}`, `{HOST.NAME}`, `{HOST.PORT}`, `{ITEM.ID}`, `{ITEM.KEY}`, `{ITEM.KEY.ORIG}`, macros utilisateur, macros de découverte de bas niveau. Cette option définit l'option cURL `CURLOPT_PROXY`.
Authentification HTTP	Sélectionnez l'option d'authentification : **Aucune** - aucune authentification n'est utilisée ; **Basic** - l'authentification Basic est utilisée ; **NTLM** - l'authentification NTLM (Windows NT LAN Manager) est utilisée ; **Kerberos** - l'authentification Kerberos est utilisée (voir également : Configuration de Kerberos avec Zabbix) ; **Digest** - l'authentification Digest est utilisée. Cette option définit l'option cURL `CURLOPT_HTTPAUTH`.
Nom d'utilisateur	Saisissez le nom d'utilisateur (255 caractères maximum). Ce champ est disponible si l'*Authentification HTTP* est définie sur *Basic*, *NTLM*, *Kerberos* ou *Digest*. Les macros utilisateur et les macros de découverte de bas niveau sont prises en charge.
Mot de passe	Saisissez le mot de passe utilisateur (255 caractères maximum). Ce champ est disponible si l'*Authentification HTTP* est définie sur *Basic*, *NTLM*, *Kerberos* ou *Digest*. Les macros utilisateur et les macros de découverte de bas niveau sont prises en charge.
Vérifier le pair SSL	Cochez la case pour vérifier le certificat SSL du serveur web. Le certificat du serveur est vérifié par rapport aux certificats des autorités de certification (CA) approuvées provenant de l'emplacement système des CA. Vous pouvez remplacer l'emplacement des fichiers CA à l'aide du paramètre de configuration du serveur Zabbix ou du proxy SSLCALocation. Cette option définit l'option cURL `CURLOPT_SSL_VERIFYPEER`.
Vérifier l'hôte SSL	Cochez la case pour vérifier que le champ *Common Name* ou le champ *Subject Alternate Name* du certificat du serveur web correspond. Cette option définit l'option cURL `CURLOPT_SSL_VERIFYHOST`.

Paramètre	Description
Fichier de certificat SSL	Nom du fichier de certificat SSL utilisé pour l'authentification du client. Le fichier de certificat doit être au format PEM[1]. Si le fichier de certificat contient également la clé privée, laissez le champ *Fichier de clé SSL* vide. Si la clé est chiffrée, spécifiez le mot de passe dans le champ *Mot de passe de la clé SSL*. Le répertoire contenant ce fichier est spécifié par le paramètre de configuration du serveur Zabbix ou du proxy `SSLCertLocation`. Macros prises en charge : `{HOST.IP}`, `{HOST.CONN}`, `{HOST.DNS}`, `{HOST.HOST}`, `{HOST.NAME}`, `{HOST.PORT}`, `{ITEM.ID}`, `{ITEM.KEY}`, `{ITEM.KEY.ORIG}`, macros utilisateur, macros de découverte de bas niveau. Cette option définit l'option cURL `CURLOPT_SSLCERT`.
Fichier de clé SSL	Nom du fichier de clé privée SSL utilisé pour l'authentification du client. Le fichier de clé privée doit être au format PEM[1]. Le répertoire contenant ce fichier est spécifié par le paramètre de configuration du serveur Zabbix ou du proxy `SSLKeyLocation`. Macros prises en charge : `{HOST.IP}`, `{HOST.CONN}`, `{HOST.DNS}`, `{HOST.HOST}`, `{HOST.NAME}`, `{HOST.PORT}`, `{ITEM.ID}`, `{ITEM.KEY}`, `{ITEM.KEY.ORIG}`, macros utilisateur, macros de découverte de bas niveau. Cette option définit l'option cURL `CURLOPT_SSLKEY`.
Mot de passe de la clé SSL	Mot de passe du fichier de clé privée SSL. Macros prises en charge : macros utilisateur, macros de découverte de bas niveau. Cette option définit l'option cURL `CURLOPT_KEYPASSWD`.
Délai d'attente	Zabbix ne consacrera pas plus de la durée définie au traitement de l'URL (1 à 600 secondes). En réalité, ce paramètre définit la durée maximale pour établir une connexion à l'URL et la durée maximale pour effectuer une requête HTTP. Par conséquent, Zabbix ne consacrera pas plus de 2 x *Délai d'attente* secondes à une vérification. Cette option définit l'option cURL `CURLOPT_TIMEOUT`. Pour plus d'informations sur le paramètre *Délai d'attente*, consultez les attributs généraux des éléments.
Activer la réception	Lorsque cette case est cochée, l'élément fonctionne également comme un élément trapper et accepte les données envoyées au serveur Zabbix ou au proxy à l'aide de l'utilitaire Zabbix sender ou du protocole Zabbix sender, ou accepte les données envoyées au serveur Zabbix à l'aide de la méthode API `history.push`. Pour plus d'informations sur l'envoi de données, consultez : Éléments trapper.
Hôtes autorisés	Visible uniquement si la case *Activer la réception* est cochée. Liste d'adresses IP séparées par des virgules (éventuellement en notation CIDR) ou de noms DNS. La valeur par défaut est la macro utilisateur `{$TRAPPER.ALLOWED_HOSTS}`. Par défaut, `{$TRAPPER.ALLOWED_HOSTS}` vaut `127.0.0.1,::1`. Vous pouvez saisir directement les adresses dans le champ ou modifier la valeur de la macro. Si le champ est laissé vide, les connexions entrantes seront refusées pour tous les hôtes. Si une valeur est spécifiée, les connexions entrantes seront acceptées uniquement depuis les hôtes répertoriés ici. Si la prise en charge d'IPv6 est activée, `127.0.0.1`, `::127.0.0.1` et `::ffff:127.0.0.1` sont traités de la même manière, et `::/0` autorisera toute adresse IPv4 ou IPv6. `0.0.0.0/0` peut être utilisé pour autoriser toute adresse IPv4. Notez que les « adresses IPv6 compatibles IPv4 » (préfixe `0000::/96`) sont prises en charge, mais déconseillées par la RFC4291. Exemple : `127.0.0.1`, `192.168.1.0/24`, `192.168.3.1-255`, `192.168.1-10.1-255`, `::1,2001:db8::/32`, `mysqlserver1`, `zabbix.example.com`, `{HOST.HOST}` Les espaces et les macros utilisateur sont autorisés dans ce champ. Macros d'hôte autorisées dans ce champ : `{HOST.HOST}`, `{HOST.NAME}`, `{HOST.IP}`, `{HOST.DNS}`, `{HOST.CONN}`.

Note:
Si le champ *Proxy HTTP* est laissé vide, une autre méthode pour utiliser un proxy HTTP consiste à définir des variables d'environnement liées au proxy.
Pour HTTP - définissez la variable d'environnement `http_proxy` pour l'utilisateur du serveur Zabbix. Par exemple :
 `http_proxy=http://proxy_ip:proxy_port`.
Pour HTTPS - définissez la variable d'environnement `HTTPS_PROXY`. Par exemple :
 `HTTPS_PROXY=http://proxy_ip:proxy_port`. Pour plus de détails, exécutez une commande shell : *# man curl*.

Attention:
[1] Zabbix prend uniquement en charge les fichiers de certificat et de clé privée au format PEM. Si les données de votre certificat et de votre clé privée se trouvent dans un fichier au format PKCS #12 (généralement avec l'extension `*.p12` ou `*.pfx`), vous pouvez générer le fichier PEM à l'aide des commandes suivantes :

```
openssl pkcs12 -in ssl-cert.p12 -clcerts -nokeys -out ssl-cert.pem
openssl pkcs12 -in ssl-cert.p12 -nocerts -nodes  -out ssl-cert.key
```

Exemples

Exemple 1

Envoyez de simples requêtes GET pour récupérer des données depuis des services tels qu'Elasticsearch :

- Créez un élément GET avec l'URL : `localhost:9200/?pretty`
- Observez la réponse :

```
{
  "name" : "YQ2VAY-",
  "cluster_name" : "elasticsearch",
  "cluster_uuid" : "kH4CYqh5QfqgeTsjh2F9zg",
  "version" : {
    "number" : "6.1.3",
    "build_hash" : "af51318",
    "build_date" : "2018-01-26T18:22:55.523Z",
    "build_snapshot" : false,
    "lucene_version" : "7.1.0",
    "minimum_wire_compatibility_version" : "5.6.0",
    "minimum_index_compatibility_version" : "5.0.0"
  },
  "tagline" : "You know, for search"
}
```

- Extrayez maintenant le numéro de version à l'aide d'une étape de prétraitement JSONPath : `$.version.number`

Exemple 2

Envoyez de simples requêtes POST pour récupérer des données depuis des services tels qu'Elasticsearch :

- Créez un élément POST avec l'URL : `http://localhost:9200/str/values/_search?scroll=10s`
- Configurez le corps POST suivant pour obtenir la charge du processeur (moyenne sur 1 min par cœur)

```
{
    "query": {
        "bool": {
            "must": [{
                "match": {
                    "itemid": 28275
                }
            }],
            "filter": [{
                "range": {
                    "clock": {
                        "gt": 1517565836,
                        "lte": 1517566137
                    }
                }
            }]
        }
    }
}
```

- Reçu :

```
{
    "_scroll_id": "DnF1ZXJ5VGhlbkZldGNoBQAAAAAAAAAkFllRMlZBWS1UU1pxTmdEeGVwQjRBTFEAAAAAAAAAJRZZUTJWQVk
    "took": 18,
    "timed_out": false,
```

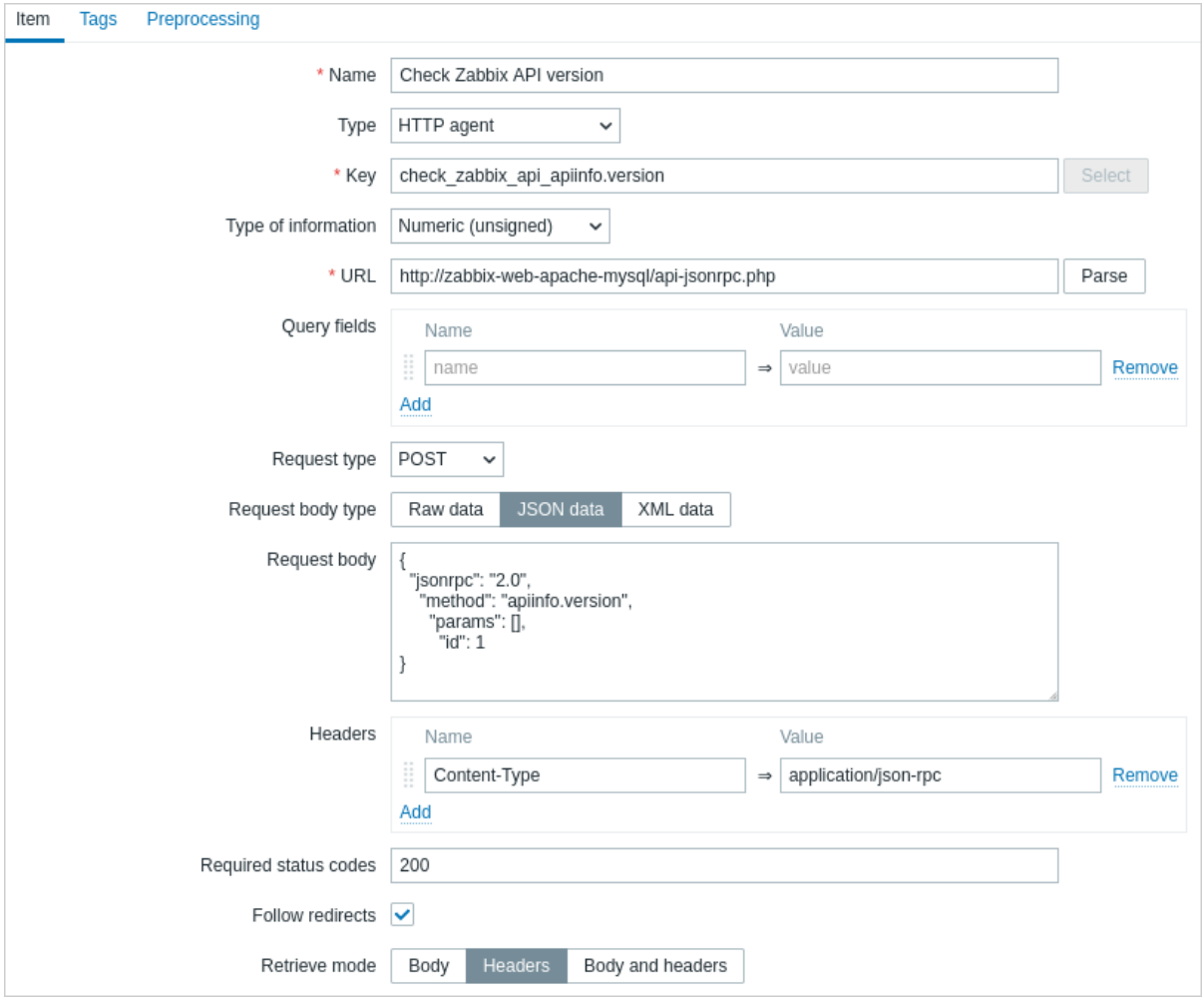
```
        "_shards": {
            "total": 5,
            "successful": 5,
            "skipped": 0,
            "failed": 0
        },
        "hits": {
            "total": 1,
            "max_score": 1.0,
            "hits": [{
                "_index": "dbl",
                "_type": "values",
                "_id": "dqX9VWEBV6sEKSMyk6sw",
                "_score": 1.0,
                "_source": {
                    "itemid": 28275,
                    "value": "0.138750",
                    "clock": 1517566136,
                    "ns": 25388713,
                    "ttl": 604800
                }
            }]
        }
    }
```


- Utilisez maintenant une étape de prétraitement JSONPath pour obtenir la valeur de l'élément : `$.hits.hits[0]._source.value`

Exemple 3

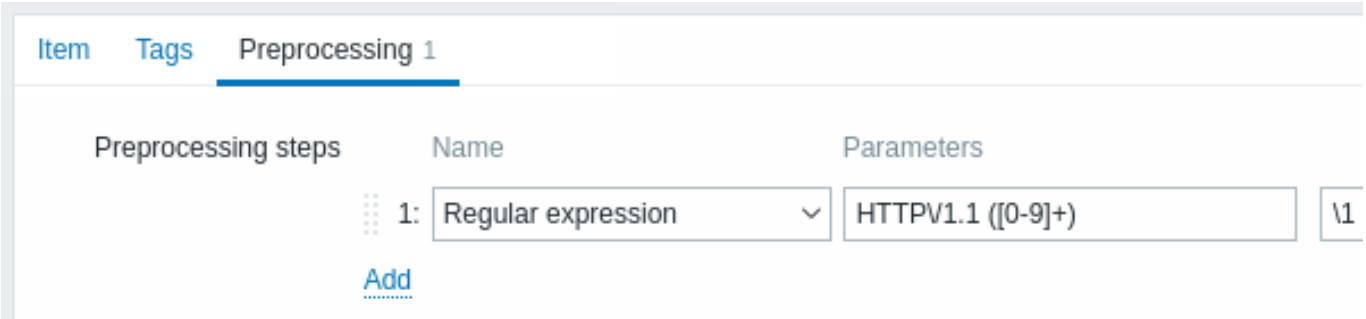
Vérification que l'API Zabbix est active, en utilisant apiinfo.version.

- Configuration de l'élément :

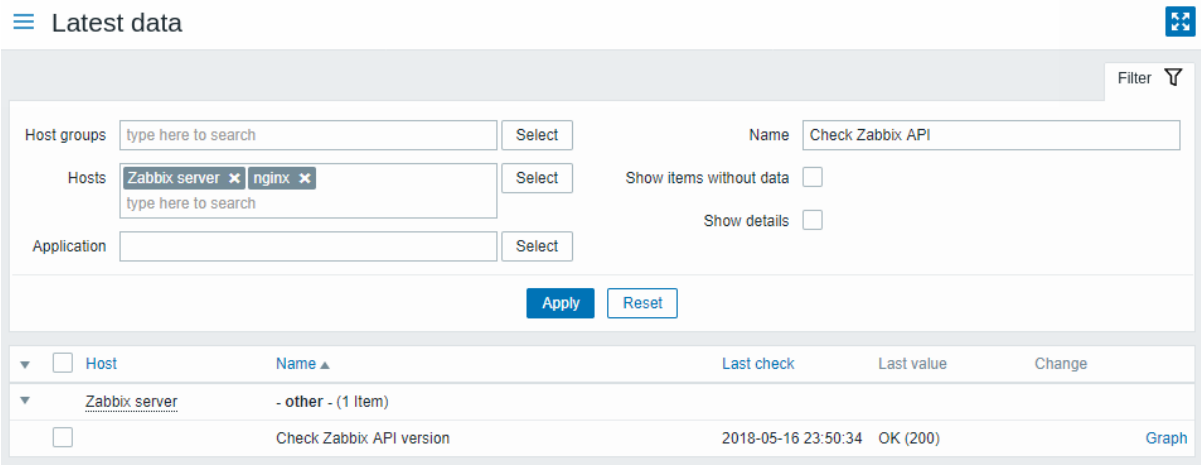


Notez l'utilisation de la méthode POST avec des données JSON, la définition des en-têtes de requête et la demande de retour des en-têtes uniquement :

- Prétraitement de la valeur de l'élément avec une expression régulière pour obtenir le code HTTP :



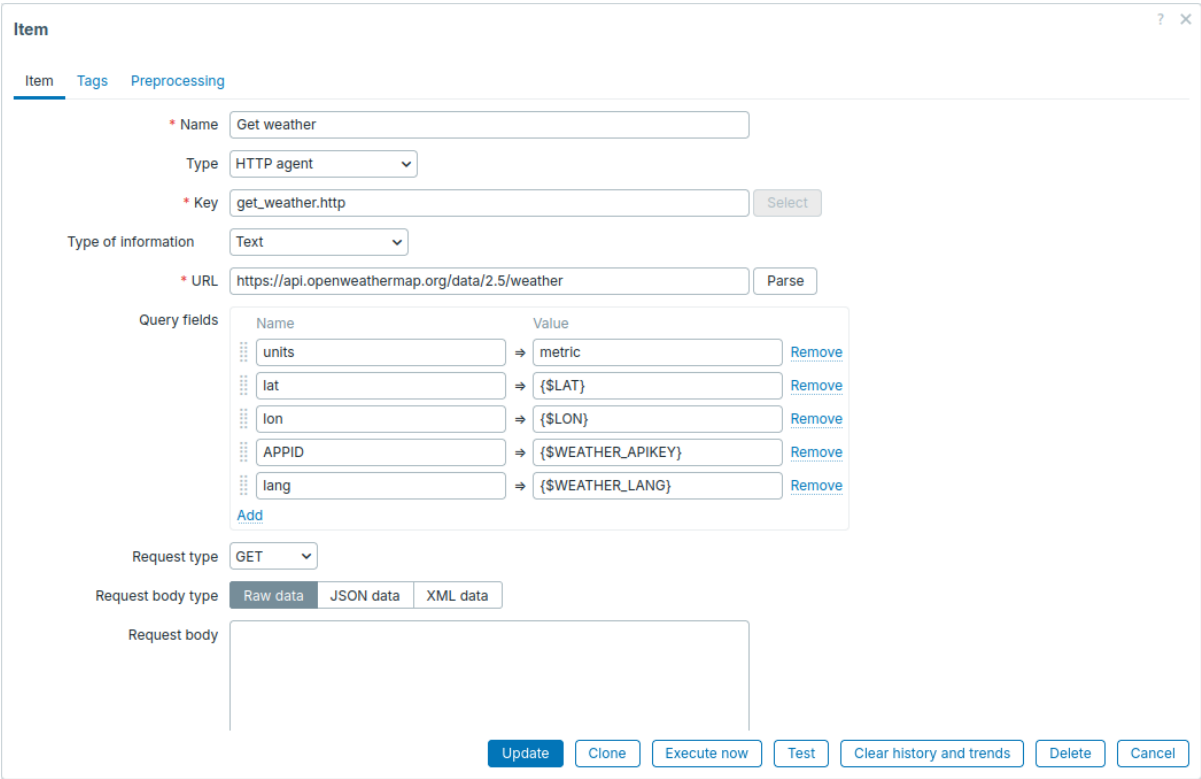
- Vérification du résultat dans *Latest data* :



Exemple 4

Récupération d'informations météorologiques en se connectant au service public Openweathermap.

- Configurez un élément maître pour la collecte de données en masse dans un seul JSON :



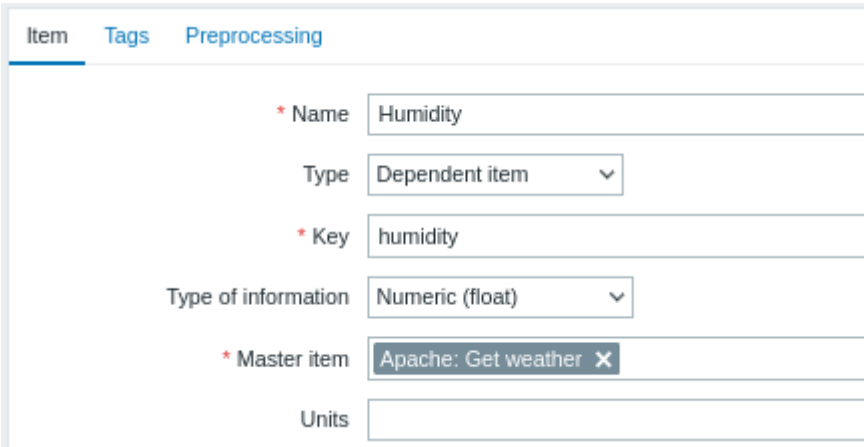
Notez l'utilisation de macros dans les champs de requête. Reportez-vous à l'API Openweathermap pour savoir comment les renseigner.

Exemple de JSON renvoyé en réponse à l'agent HTTP :


```
{
    "body": {
        "coord": {
            "lon": 40.01,
            "lat": 56.11
        },
        "weather": [{
            "id": 801,
            "main": "Clouds",
            "description": "few clouds",
            "icon": "02n"
        }],
        "base": "stations",
        "main": {
            "temp": 15.14,
            "pressure": 1012.6,
            "humidity": 66,
            "temp_min": 15.14,
            "temp_max": 15.14,
            "sea_level": 1030.91,
            "grnd_level": 1012.6
        },
        "wind": {
            "speed": 1.86,
            "deg": 246.001
        },
        "clouds": {
            "all": 20
        },
        "dt": 1526509427,
        "sys": {
            "message": 0.0035,
            "country": "RU",
            "sunrise": 1526432608,
            "sunset": 1526491828
        },
        "id": 487837,
        "name": "Stavrovo",
        "cod": 200
    }
}
```

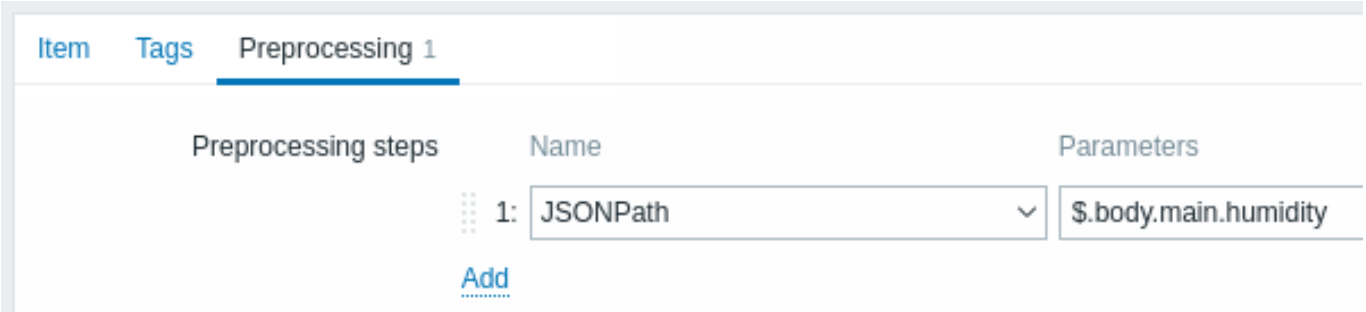

L'étape suivante consiste à configurer des éléments dépendants qui extraient les données du JSON.

- Configurez un exemple d'élément dépendant pour l'humidité :

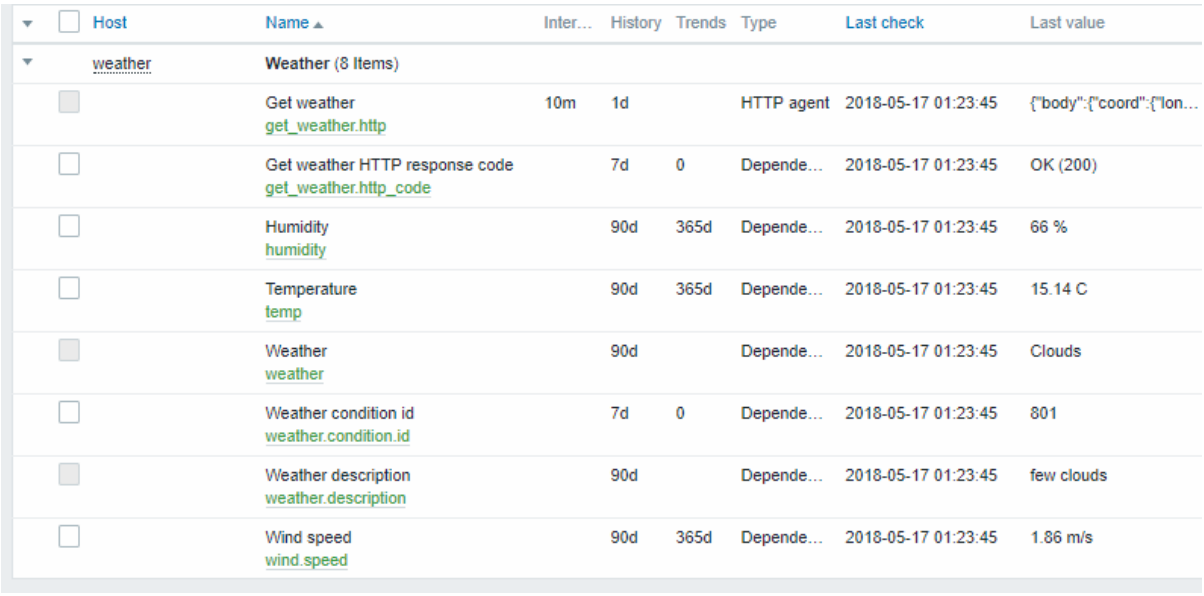


D'autres métriques météorologiques, telles que « Température », sont ajoutées de la même manière.

- Exemple de prétraitement de la valeur d'un élément dépendant avec JSONPath :



- Vérifiez le résultat des données météorologiques dans *Dernières données* :

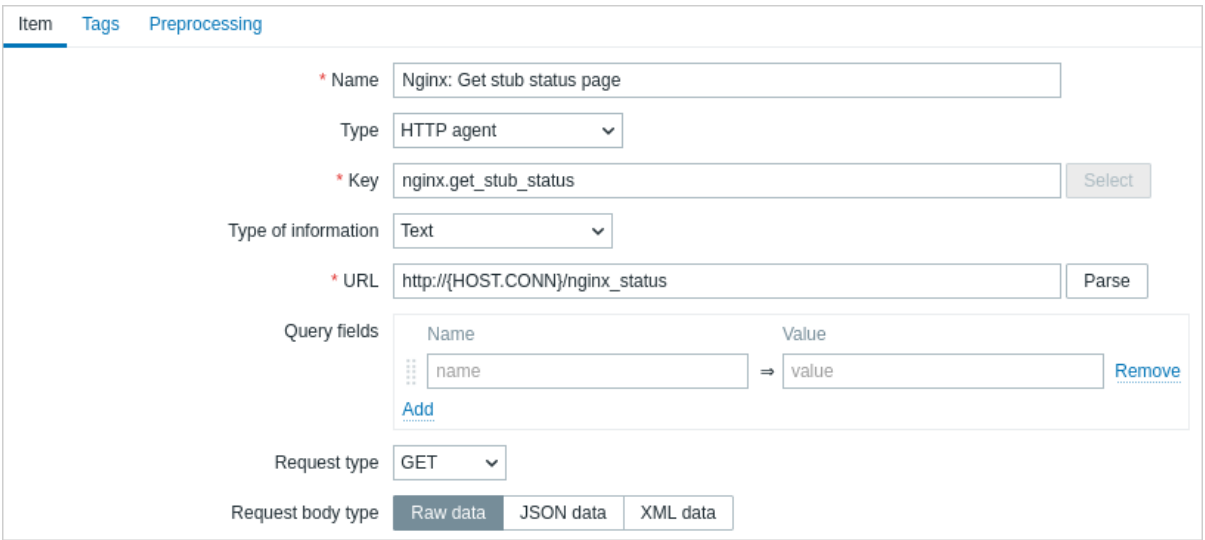


Host	Name ▲	Inter...	History	Trends	Type	Last check	Last value
weather	**Weather** (8 Items)						
	Get weather get_weather.http	10m	1d		HTTP agent	2018-05-17 01:23:45	{"body":{"coord":{"lon...
	Get weather HTTP response code get_weather.http_code		7d	0	Depende...	2018-05-17 01:23:45	OK (200)
	Humidity humidity		90d	365d	Depende...	2018-05-17 01:23:45	66 %
	Temperature temp		90d	365d	Depende...	2018-05-17 01:23:45	15.14 C
	Weather weather		90d		Depende...	2018-05-17 01:23:45	Clouds
	Weather condition id weather.condition.id		7d	0	Depende...	2018-05-17 01:23:45	801
	Weather description weather.description		90d		Depende...	2018-05-17 01:23:45	few clouds
	Wind speed wind.speed		90d	365d	Depende...	2018-05-17 01:23:45	1.86 m/s

Exemple 5

Connexion à la page d'état NGINX et récupération de ses métriques en bloc.

- Configurez NGINX en suivant le guide officiel.
- Configurez un élément maître pour la collecte de données en bloc :

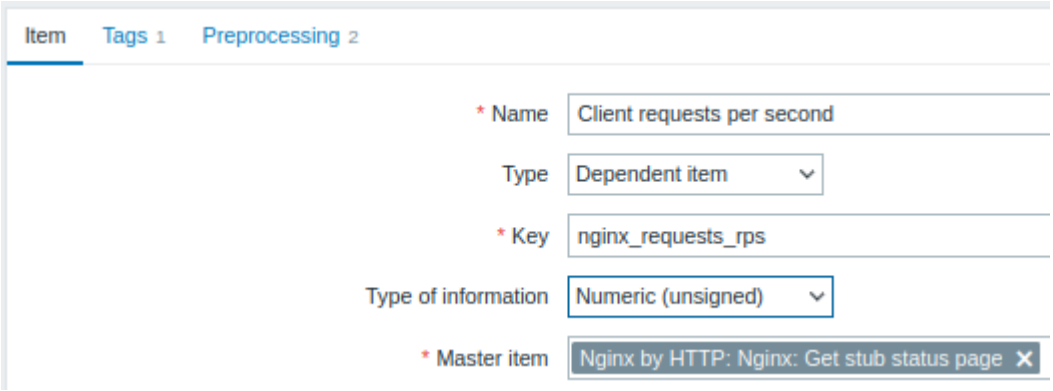


Exemple de sortie du module stub de NGINX :

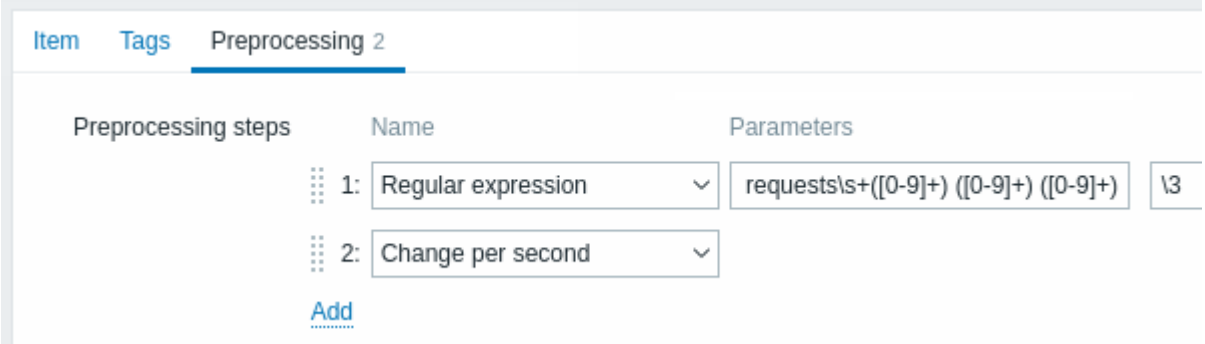
```
Active connections: 1 Active connections:
server accepts handled requests
 52 52 52
Reading: 0 Writing: 1 Waiting: 0
```

La tâche suivante consiste à configurer les éléments dépendants qui extraient les données.

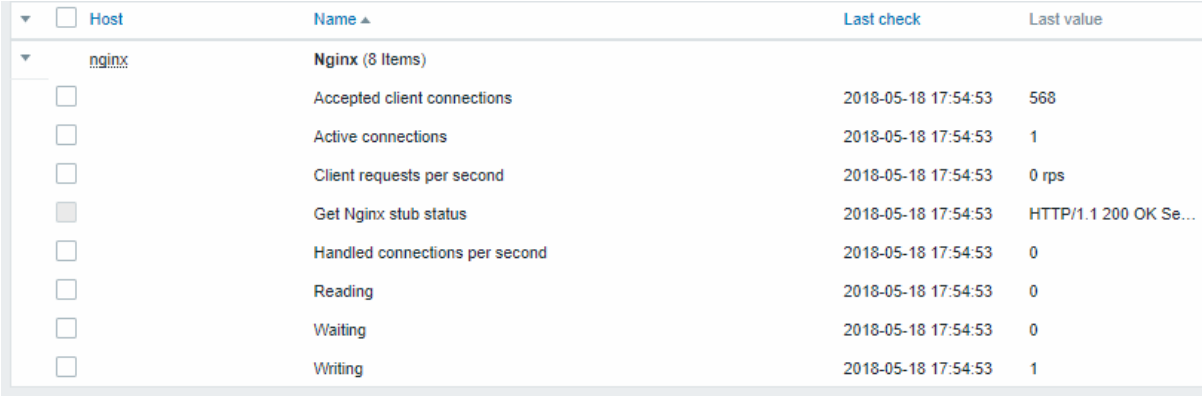
- Configurez un exemple d'élément dépendant pour les requêtes par seconde :



- Exemple de prétraitement de la valeur d'un élément dépendant avec l'expression régulière `server accepts handled requests\s+([0-9]+) ([0-9]+) ([0-9]+)` :



- Vérifiez le résultat complet du module stub dans *Dernières données* :



Vérification Prometheus

Vue d'ensemble

Zabbix peut interroger des métriques exposées au format de ligne Prometheus.

Deux étapes sont nécessaires pour commencer à collecter des données Prometheus :

- un élément maître HTTP pointant vers le point de terminaison de données approprié, par exemple `https://<prometheus host>/metrics`
- des éléments dépendants utilisant une option de prétraitement Prometheus pour interroger les données requises à partir des métriques collectées par l'élément maître

Il existe deux options de prétraitement des données Prometheus :

- *Motif Prometheus* - utilisé dans les éléments normaux pour interroger les données Prometheus
- *Prometheus vers JSON* - utilisé dans les éléments normaux et pour la découverte de bas niveau. Dans ce cas, les données Prometheus interrogées sont renvoyées au format JSON.

Traitement par lot

Le traitement par lot est pris en charge pour les éléments dépendants. Pour activer la mise en cache et l'indexation, la prétraitement *Prometheus pattern* doit être la **première** étape de prétraitement. Lorsque *Prometheus pattern* est la première étape de

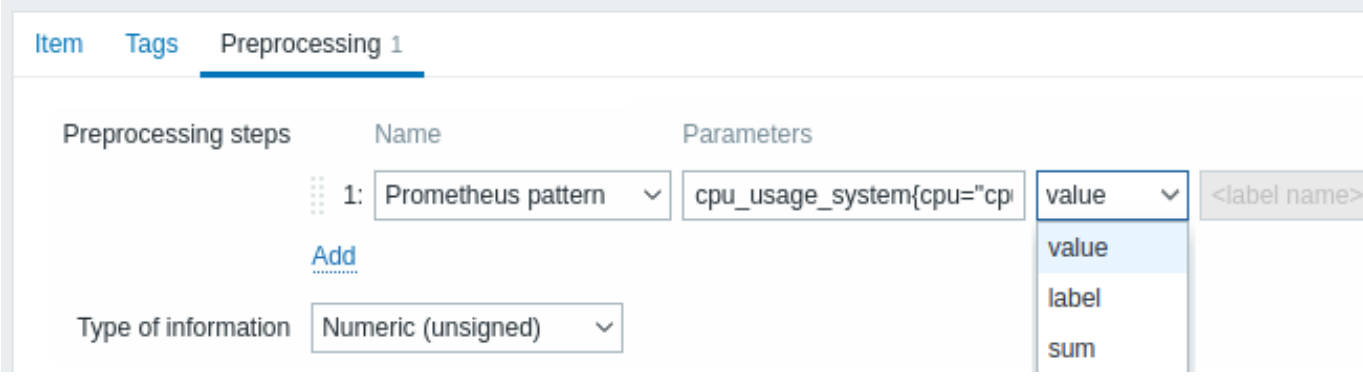
prétraitement, les données Prometheus analysées sont mises en cache et indexées par la première condition `<label>==<value>` dans l'étape de prétraitement *Prometheus pattern*. Ce cache est réutilisé lors du traitement d'autres éléments dépendants dans ce lot. Pour des performances optimales, le premier label doit être celui qui possède le plus grand nombre de valeurs différentes.

S'il existe un autre prétraitement à effectuer avant la première étape, il doit être déplacé soit vers l'élément maître, soit vers un nouvel élément dépendant qui serait utilisé comme élément maître pour les éléments dépendants.

Configuration

À condition d'avoir configuré l'élément maître HTTP, vous devez créer un élément dépendant qui utilise une étape de prétraitement Prometheus :

1. Saisissez les paramètres généraux de l'élément dépendant dans le formulaire de configuration.
2. Accédez à l'onglet *Prétraitement*.
3. Sélectionnez une option de prétraitement Prometheus (*Prometheus pattern* ou *Prometheus to JSON*).



Les paramètres suivants sont spécifiques à l'option de prétraitement *Prometheus pattern* :

Paramètre	Description	Exemples
Pattern	Pour définir le modèle de données requis, vous pouvez utiliser un langage de requête similaire au langage de requête Prometheus (voir le tableau comparatif), par exemple : <metric name> - sélectionner par nom de métrique {__name__="<metric name>"} - sélectionner par nom de métrique {__name__=~"<regex>"} - sélectionner par nom de métrique correspondant à une expression régulière {<label name>="<label value>",...} - sélectionner par nom de libellé {<label name>=~"<regex>",...} - sélectionner par nom de libellé correspondant à une expression régulière {__name__=~".*"}==<value> - sélectionner par valeur de métrique Ou une combinaison des éléments ci-dessus : <metric name>{<label1 name>="<label1 value>",<label2 name>=~"<regex>",...}==<value> La valeur du libellé peut être toute séquence de caractères UTF-8, mais les caractères barre oblique inverse, guillemet double et saut de ligne doivent être échappés respectivement sous la forme \\, \" et \n ; les autres caractères ne doivent pas être échappés.	*wmi_os_physical_memory_free_bytes* *cpu_usage_system{cpu="cpu-total"}* *cpu_usage_system{cpu=".*"}* *cpu_usage_system{cpu="cpu-total",host="* *wmi_service_state{name="dhcp"}==1* *wmi_os_timezone{timezone=~".*"}==1*

Paramètre	Description	Exemples
Result processing	Indiquez s'il faut retourner la valeur, le libellé ou appliquer la fonction appropriée (si le modèle correspond à plusieurs lignes et que le résultat doit être agrégé) : **value** - retourner la valeur de la métrique (erreur si plusieurs lignes correspondent) **label** - retourner la valeur du libellé indiqué dans le champ *Label* (erreur si plusieurs métriques correspondent) **sum** - retourner la somme des valeurs **min** - retourner la valeur minimale **max** - retourner la valeur maximale **avg** - retourner la valeur moyenne **count** - retourner le nombre de valeurs Ce champ est disponible uniquement pour l'option *Prometheus pattern*.	Voir également les exemples d'utilisation des paramètres ci-dessous.
Output	Définissez le nom du libellé (facultatif). Dans ce cas, la valeur correspondant au nom du libellé est retournée. Ce champ est disponible uniquement pour l'option *Prometheus pattern*, si « Label » est sélectionné dans le champ *Result processing*.	

Exemples d'utilisation des paramètres

1. Le cas d'utilisation le plus courant consiste à retourner la **valeur**. Pour retourner la valeur de `/var/db` à partir de :

   `node_disk_usage_bytes{path="/var/cache"} 2.1766144e+09<br> node_disk_usage_bytes{path="/var/db"} 20480<br> node_disk_usage_bytes{path="/var/dpkg"} 8192<br> node_disk_usage_bytes{path="/var/empty"} 4096`

utilisez les paramètres suivants :

- *Pattern* - `node_disk_usage_bytes{path="/var/db"}`
- *Result processing* - sélectionnez « value »

2. Vous pouvez également être intéressé par la valeur **moyenne** de tous les paramètres `node_disk_usage_bytes` :

- *Pattern* - `node_disk_usage_bytes`
- *Result processing* - sélectionnez « avg »

3. Bien que Prometheus ne prenne en charge que les données numériques, une solution de contournement est couramment utilisée pour retourner également la description textuelle correspondante. Cela peut être réalisé à l'aide d'un filtre et en indiquant le libellé. Ainsi, pour retourner la valeur du libellé « color » à partir de :

   `elasticsearch_cluster_health_status{cluster="elasticsearch",color="green"} 1<br> elasticsearch_cluster 0<br> elasticsearch_cluster_health_status{cluster="elasticsearch",color="yellow"} 0`

utilisez les paramètres suivants :

- *Pattern* - `elasticsearch_cluster_health_status {cluster="elasticsearch"} == 1`
- *Result processing* - sélectionnez « label »
- *Label* - indiquez « color »

Le filtre (basé sur la valeur numérique « 1 ») correspondra à la ligne appropriée, tandis que le libellé retournera la description de l'état de santé (actuellement « green », mais potentiellement aussi « red » ou « yellow »).

Prometheus vers JSON

Les données de Prometheus peuvent être utilisées pour la découverte de bas niveau. Dans ce cas, des données au format JSON sont nécessaires et l'option de prétraitement *Prometheus to JSON* renverra exactement cela.

Pour plus de détails, consultez Découverte à l'aide des données Prometheus.

Comparaison des langages de requête

Le tableau suivant répertorie les différences et les similitudes entre PromQL et le langage de requête de prétraitement Prometheus de Zabbix.

Sélecteur de vecteur instantané PromQL		Langage de prétraitement Prometheus de Zabbix
Différences		
Cible de la requête	serveur Prometheus	Texte brut au format d'exposition Prometheus
Renvoie	Vecteur instantané	Valeur de métrique ou de libellé (modèle Prometheus) Tableau de métriques pour une valeur unique en JSON (Prometheus vers JSON)
Opérateurs de correspondance des libellés	**=**, **!=**, **=~**, **!~**	**=**, **!=**, **=~**, **!~**
Expression régulière utilisée dans la correspondance du nom de libellé ou de métrique	RE2	PCRE
Opérateurs de comparaison	Voir la liste	Seul **==** (égal) est pris en charge pour le filtrage des valeurs
Similitudes		
Sélection par nom de métrique égal à une chaîne	<nom de métrique> ou {__name__="<nom de métrique>"}	<nom de métrique> ou {__name__="<nom de métrique>"}
Sélection par nom de métrique correspondant à une expression régulière	{__name__=~"<regex>"}	{__name__=~"<regex>"}
Sélection par valeur de <nom de libellé> égale à une chaîne	{<nom de libellé>="<valeur de libellé>",...}	{<nom de libellé>="<valeur de libellé>",...}
Sélection par valeur de <nom de libellé> correspondant à une expression régulière	{<nom de libellé>=~"<regex>",...}	{<nom de libellé>=~"<regex>",...}
Sélection par valeur égale à une chaîne	{__name__=~".*"} == <value>	{__name__=~".*"} == <value>

agent IPMI

Présentation

Vous pouvez surveiller l'état et la disponibilité des périphériques Intelligent Platform Management Interface (IPMI) dans Zabbix. Pour effectuer des vérifications IPMI, le serveur Zabbix doit être initialement configuré avec la prise en charge d'IPMI.

IPMI est une interface standardisée destinée à la gestion à distance « lights-out » ou « hors bande » des systèmes informatiques. Elle vous permet de surveiller directement l'état du matériel à partir des cartes de gestion dites « hors bande », indépendamment du système d'exploitation ou du fait que la machine soit sous tension ou non.

La surveillance IPMI de Zabbix fonctionne uniquement pour les périphériques prenant en charge IPMI (HP iLO, DELL DRAC, IBM RSA, Sun SSP, etc.).

Un processus de gestionnaire IPMI planifie les vérifications IPMI par l'intermédiaire des pollers IPMI. Un hôte est toujours interrogé par un seul poller IPMI à la fois, ce qui réduit le nombre de connexions ouvertes aux contrôleurs BMC. Il est donc possible d'augmenter le nombre de pollers IPMI sans craindre de surcharger le contrôleur BMC. Le processus de gestionnaire IPMI est automatiquement démarré lorsqu'au moins un poller IPMI est démarré.

Consultez également les problèmes connus concernant les vérifications IPMI.

Configuration

Configuration de l'hôte

Un hôte doit être configuré pour traiter les vérifications IPMI. Une interface IPMI doit être ajoutée, avec les adresses IP et les numéros de port respectifs, et les paramètres d'authentification IPMI doivent être définis.

Voir la configuration des hôtes pour plus de détails.

Configuration du serveur

Par défaut, le serveur Zabbix n'est pas configuré pour démarrer des pollers IPMI, donc tous les éléments IPMI ajoutés ne fonctionneront pas. Pour modifier cela, ouvrez le fichier de configuration du serveur Zabbix (`zabbix_server.conf`) en tant que root et recherchez la ligne suivante :

```
# StartIPMIPollers=0
```

Décommentez-la et définissez le nombre de pollers, par exemple sur 3, de sorte qu'elle devienne :

```
StartIPMIPollers=3
```

Enregistrez le fichier, puis redémarrez zabbix_server.

Configuration de l'élément

Lors de la configuration d'un élément au niveau de l'hôte :

- Sélectionnez 'IPMI agent' comme *Type*
- Saisissez une clé unique sur l'hôte (telle que, ipmi.fan.rpm)
- Pour l'*interface hôte*, sélectionnez l'interface IPMI adéquate (IP et port). Notez que l'interface IPMI doit exister sur l'hôte.
- Spécifiez le *capteur IPMI* (par exemple 'FAN MOD 1A RPM' sur Dell Poweredge) sur lequel récupérer la valeur. Par défaut, l'ID du capteur doit être spécifié. Il est aussi possible d'utiliser des préfixes avant la valeur :
  - `id:` - pour spécifier l'ID du capteur ;
  - `name:` - pour spécifier le nom complet du capteur. Cela peut être utile dans certaines situations quand les capteurs peuvent seulement être distingués en spécifiant le nom complet.
- Sélectionnez le type d'information correspondant ('Numérique (flottant)' dans ce cas, pour les capteurs discrets - 'Numérique (non signé)'), les unités (très probablement 'rpm') et tout autre attribut d'élément requis

Vérifications prises en charge

L'agent IPMI prend en charge l'élément intégré **ipmi.get**, qui renvoie des informations relatives aux capteurs IPMI et peut être utilisé pour la découverte des capteurs IPMI.
 Valeur de retour : *objet JSON*

Délai d'expiration et fin de session

Les délais d'expiration des messages IPMI et le nombre de tentatives sont définis dans la bibliothèque OpenIPMI. En raison de la conception actuelle d'OpenIPMI, il n'est pas possible de rendre ces valeurs configurables dans Zabbix, ni au niveau de l'interface ni au niveau de l'élément.

Le délai d'inactivité de session IPMI pour le LAN est de 60 +/-3 secondes. Actuellement, il n'est pas possible d'implémenter l'envoi périodique de la commande Activate Session avec OpenIPMI. S'il n'y a aucune vérification d'élément IPMI de Zabbix vers un BMC particulier pendant une durée supérieure au délai d'expiration de session configuré dans le BMC, alors la vérification IPMI suivante, une fois ce délai expiré, expirera en raison des délais d'expiration des messages individuels, des tentatives de nouvelle émission ou d'une erreur de réception. Après cela, une nouvelle session est ouverte et une réanalyse complète du BMC est lancée. Un nouveau port UDP peut être ouvert pour gérer la nouvelle session. L'inactivité est définie par l'absence à la fois de requêtes sortantes et de réponses entrantes. Si vous souhaitez éviter des réanalyses inutiles du BMC, il est conseillé de définir l'intervalle d'interrogation de l'élément IPMI en dessous du délai d'inactivité de session IPMI configuré dans le BMC.

Remarques sur les capteurs discrets IPMI

Pour trouver les capteurs sur un hôte, démarrez le serveur Zabbix avec **DebugLevel=4** activé. Attendez quelques minutes, puis recherchez les enregistrements de découverte des capteurs dans le fichier journal du serveur Zabbix :

```
$ grep 'Added sensor' zabbix_server.log
8358:20130318:111122.170 Added sensor: host:'192.168.1.12:623' id_type:0 id_sz:7 id:'CATERR' reading_type:
8358:20130318:111122.170 Added sensor: host:'192.168.1.12:623' id_type:0 id_sz:15 id:'CPU Therm Trip' read
8358:20130318:111122.171 Added sensor: host:'192.168.1.12:623' id_type:0 id_sz:17 id:'System Event Log' re
8358:20130318:111122.171 Added sensor: host:'192.168.1.12:623' id_type:0 id_sz:17 id:'PhysicalSecurity' re
8358:20130318:111122.171 Added sensor: host:'192.168.1.12:623' id_type:0 id_sz:14 id:'IPMI Watchdog' readi
8358:20130318:111122.171 Added sensor: host:'192.168.1.12:623' id_type:0 id_sz:16 id:'Power Unit Stat' rea
8358:20130318:111122.171 Added sensor: host:'192.168.1.12:623' id_type:0 id_sz:16 id:'P1 Therm Ctrl %' rea
8358:20130318:111122.172 Added sensor: host:'192.168.1.12:623' id_type:0 id_sz:16 id:'P1 Therm Margin' rea
8358:20130318:111122.172 Added sensor: host:'192.168.1.12:623' id_type:0 id_sz:13 id:'System Fan 2' readin
8358:20130318:111122.172 Added sensor: host:'192.168.1.12:623' id_type:0 id_sz:13 id:'System Fan 3' readin
8358:20130318:111122.172 Added sensor: host:'192.168.1.12:623' id_type:0 id_sz:14 id:'P1 Mem Margin' readi
8358:20130318:111122.172 Added sensor: host:'192.168.1.12:623' id_type:0 id_sz:17 id:'Front Panel Temp' re
8358:20130318:111122.173 Added sensor: host:'192.168.1.12:623' id_type:0 id_sz:15 id:'Baseboard Temp' read
8358:20130318:111122.173 Added sensor: host:'192.168.1.12:623' id_type:0 id_sz:9 id:'BB +5.0V' reading_typ
8358:20130318:111122.173 Added sensor: host:'192.168.1.12:623' id_type:0 id_sz:14 id:'BB +3.3V STBY' readi
8358:20130318:111122.173 Added sensor: host:'192.168.1.12:623' id_type:0 id_sz:9 id:'BB +3.3V' reading_typ
8358:20130318:111122.173 Added sensor: host:'192.168.1.12:623' id_type:0 id_sz:17 id:'BB +1.5V P1 DDR3' re
8358:20130318:111122.173 Added sensor: host:'192.168.1.12:623' id_type:0 id_sz:17 id:'BB +1.1V P1 Vccp' re
8358:20130318:111122.174 Added sensor: host:'192.168.1.12:623' id_type:0 id_sz:14 id:'BB +1.05V PCH' readi
```

Pour décoder les types et états des capteurs IPMI, une copie des spécifications IPMI 2.0 est disponible (veuillez noter qu'aucune mise à jour supplémentaire de la spécification IPMI n'est prévue).

Le premier paramètre à examiner est "reading_type". Utilisez "Table 42-1, Event/Reading Type Code Ranges" des spécifications pour décoder le code "reading_type". La plupart des capteurs de notre exemple ont "reading_type:0x1", ce qui signifie un capteur de type "threshold". "Table 42-3, Sensor Type Codes" montre que "type:0x1" signifie capteur de température, "type:0x2" - capteur de tension, "type:0x4" - ventilateur, etc. Les capteurs à seuil sont parfois appelés capteurs "analogiques", car ils mesurent des paramètres continus comme la température, la tension, les tours par minute.

Autre exemple : un capteur avec "reading_type:0x3". "Table 42-1, Event/Reading Type Code Ranges" indique que les codes de type de lecture 02h-0Ch signifient un capteur "Generic Discrete". Les capteurs discrets ont jusqu'à 15 états possibles (autrement dit, jusqu'à 15 bits significatifs). Par exemple, pour le capteur 'CATERR' avec "type:0x7", la "Table 42-3, Sensor Type Codes" montre que ce type signifie "Processor" et que la signification des bits individuels est : 00h (le bit de poids faible) - IERR, 01h - Thermal Trip, etc.

Il y a quelques capteurs avec "reading_type:0x6f" dans notre exemple. Pour ces capteurs, la "Table 42-1, Event/Reading Type Code Ranges" recommande d'utiliser la "Table 42-3, Sensor Type Codes" pour décoder la signification des bits. Par exemple, le capteur 'Power Unit Stat' a le type "type:0x9", qui signifie "Power Unit". L'offset 00h signifie "PowerOff/Power Down". En d'autres termes, si le bit de poids faible vaut 1, alors le serveur est hors tension. Pour tester ce bit, la fonction `bitand` avec le masque '1' peut être utilisée. L'expression de déclencheur pourrait être :

```
bitand(last(/www.example.com/Power Unit Stat,#1),1)=1
```

pour avertir qu'un serveur est hors tension.

Notes sur les noms de capteurs discrets dans OpenIPMI-2.0.16, 2.0.17, 2.0.18 et 2.0.19

Les noms de capteurs discrets dans OpenIPMI-2.0.16, 2.0.17 et 2.0.18 ont souvent un "0" supplémentaire (ou un autre chiffre ou lettre) ajouté à la fin. Par exemple, si `ipmitool` et OpenIPMI-2.0.19 affichent les noms des capteurs comme "`PhysicalSecurity`" ou "`CATERR`", dans OpenIPMI-2.0.16, 2.0.17 et 2.0.18, les noms sont "`PhysicalSecurity0`" ou "`CATERR0`", respectivement.

Lors de la configuration d'un élément IPMI avec le serveur Zabbix à l'aide d'OpenIPMI-2.0.16, 2.0.17 et 2.0.18, utilisez ces noms se terminant par "0" dans le champ *Capteur IPMI* des éléments de l'agent IPMI. Lorsque votre serveur Zabbix est mis à jour vers une nouvelle distribution Linux, qui utilise OpenIPMI-2.0.19 (ou une version ultérieure), les éléments avec ces capteurs IPMI discrets deviendront "NON SUPPORTE". Vous devrez changer leur nom de *capteur IPMI* (enlevez le '0' à la fin) et attendre un moment avant de les réactiver.

Remarques sur la disponibilité simultanée des capteurs à seuil et des capteurs discrets

Certains agents IPMI fournissent à la fois un capteur à seuil et un capteur discret sous le même nom. La préférence est toujours donnée au capteur à seuil.

Remarques sur la fin de la connexion

Si les vérifications IPMI ne sont pas effectuées (pour une raison quelconque : tous les éléments IPMI hôte désactivés/non pris en charge, hôte désactivé/supprimé, hôte en maintenance, etc.) la connexion IPMI sera terminée du serveur ou du proxy Zabbix en 3 à 4 heures selon l'heure lorsque le serveur/proxy Zabbix a été démarré.

agent SSH

Vue d'ensemble

Les vérifications SSH sont effectuées en tant que supervision sans agent. L'agent Zabbix n'est pas nécessaire pour les vérifications SSH.

Pour effectuer des vérifications SSH, le serveur Zabbix doit d'abord être configuré avec la prise en charge de SSH2 (libssh ou libssh2). Voir aussi : Exigences.

> **Attention:**
> À partir de RHEL 8, seul libssh est pris en charge. Pour les autres distributions, libssh est recommandé par rapport à libssh2.

Configuration

Authentification par phrase secrète

Les vérifications SSH offrent deux méthodes d'authentification : une paire nom d'utilisateur/mot de passe et une authentification basée sur un fichier de clé.

Si vous n'avez pas l'intention d'utiliser des clés, aucune configuration supplémentaire n'est requise, à part l'association de libssh ou libssh2 à Zabbix, si vous effectuez une compilation à partir des sources.

Authentification par fichier de clé

Pour utiliser l'authentification par clé pour les éléments SSH, certaines modifications de la configuration du serveur sont nécessaires.

Ouvrez le fichier de configuration du serveur Zabbix (`zabbix_server.conf`) en tant que `root` et recherchez la ligne suivante :

```
##### SSHKeyLocation=
```

Décommentez-la et définissez le chemin complet vers le dossier où se trouveront les clés publique et privée :

```
SSHKeyLocation=/home/zabbix/.ssh
```

Enregistrez le fichier, puis redémarrez le serveur Zabbix.

Le chemin `/home/zabbix` correspond ici au répertoire personnel du compte utilisateur *zabbix*, et `.ssh` est un répertoire dans lequel, par défaut, les clés publique et privée seront générées par une commande ssh-keygen dans le répertoire personnel.

En général, les paquets d'installation du serveur Zabbix provenant de différentes distributions OS créent le compte utilisateur *zabbix* avec un répertoire personnel situé ailleurs, par exemple `/var/lib/zabbix` (comme pour les comptes système).

Avant de générer les clés, vous pouvez réattribuer le répertoire personnel à `/home/zabbix`, afin qu'il corresponde au paramètre de configuration `SSHKeyLocation` du serveur Zabbix mentionné ci-dessus.

> **Note:**
> Les étapes suivantes peuvent être ignorées si le compte *zabbix* a été ajouté manuellement conformément à la section d'installation. Dans ce cas, le répertoire personnel du compte *zabbix* est très probablement déjà `/home/zabbix`.

Pour modifier le répertoire personnel du compte utilisateur *zabbix*, tous les processus actifs qui l'utilisent doivent être arrêtés :

```
systemctl stop zabbix-agent
systemctl stop zabbix-server
```

Pour modifier l'emplacement du répertoire personnel en tentant de le déplacer (s'il existe), la commande suivante doit être exécutée :

```
usermod -m -d /home/zabbix zabbix
```

Il est également possible qu'un répertoire personnel n'ait pas existé à l'ancien emplacement ; il doit alors être créé au nouvel emplacement. Une tentative sûre consiste à faire :

```
test -d /home/zabbix || mkdir /home/zabbix
```

Pour s'assurer que tout est sécurisé, des commandes supplémentaires peuvent être exécutées pour définir les permissions du répertoire personnel :

```
chown zabbix:zabbix /home/zabbix
chmod 700 /home/zabbix
```

Les processus arrêtés précédemment peuvent maintenant être redémarrés :

```
systemctl start zabbix-agent
systemctl start zabbix-server
```

Les étapes de génération des clés publique et privée peuvent maintenant être effectuées avec les commandes suivantes (pour une meilleure lisibilité, les invites de commande sont commentées) :

```
sudo -u zabbix ssh-keygen -t rsa
##### Generating public/private rsa key pair.
##### Enter file in which to save the key (/home/zabbix/.ssh/id_rsa):
/home/zabbix/.ssh/id_rsa
##### Enter passphrase (empty for no passphrase):
<Leave empty>
##### Enter same passphrase again:
<Leave empty>
##### Your identification has been saved in /home/zabbix/.ssh/id_rsa.
##### Your public key has been saved in /home/zabbix/.ssh/id_rsa.pub.
##### The key fingerprint is:
##### 90:af:e4:c7:e3:f0:2e:5a:8d:ab:48:a2:0c:92:30:b9 zabbix@it0
##### The key's randomart image is:
##### +--[ RSA 2048]----+
```

```
##### |                 |
##### |       .         |
##### |      o          |
##### | .      o        |
##### |+      . S       |
##### |.+    o =        |
##### |E .    * =       |
##### |=o . ..* .       |
##### |... oo.o+        |
##### +-----------------+
```

Note:
Les clés publique et privée (`id_rsa.pub` et `id_rsa`) ont été générées par défaut dans le répertoire `/home/zabbix/.ssh`, ce qui correspond au paramètre de configuration `SSHKeyLocation` du serveur Zabbix.

Attention:
Les types de clés autres que "rsa" peuvent être pris en charge par l'outil ssh-keygen et les serveurs SSH, mais ils peuvent ne pas être pris en charge par libssh2 utilisé par Zabbix.

Formulaire de configuration du shell

Cette étape doit être effectuée une seule fois pour chaque hôte qui sera surveillé par des vérifications SSH.

En utilisant les commandes suivantes, le fichier de clé **publique** peut être installé sur un hôte distant *10.10.10.10*, afin que les vérifications SSH puissent être effectuées avec un compte *root* (pour une meilleure lisibilité, les invites de commande sont commentées) :

```
sudo -u zabbix ssh-copy-id root@10.10.10.10
##### The authenticity of host '10.10.10.10 (10.10.10.10)' can't be established.
##### RSA key fingerprint is 38:ba:f2:a4:b5:d9:8f:52:00:09:f7:1f:75:cc:0b:46.
##### Are you sure you want to continue connecting (yes/no)?
yes
##### Warning: Permanently added '10.10.10.10' (RSA) to the list of known hosts.
##### root@10.10.10.10's password:
<Enter root password>
##### Now try logging into the machine, with "ssh 'root@10.10.10.10'",
##### and check to make sure that only the key(s) you wanted were added.
```

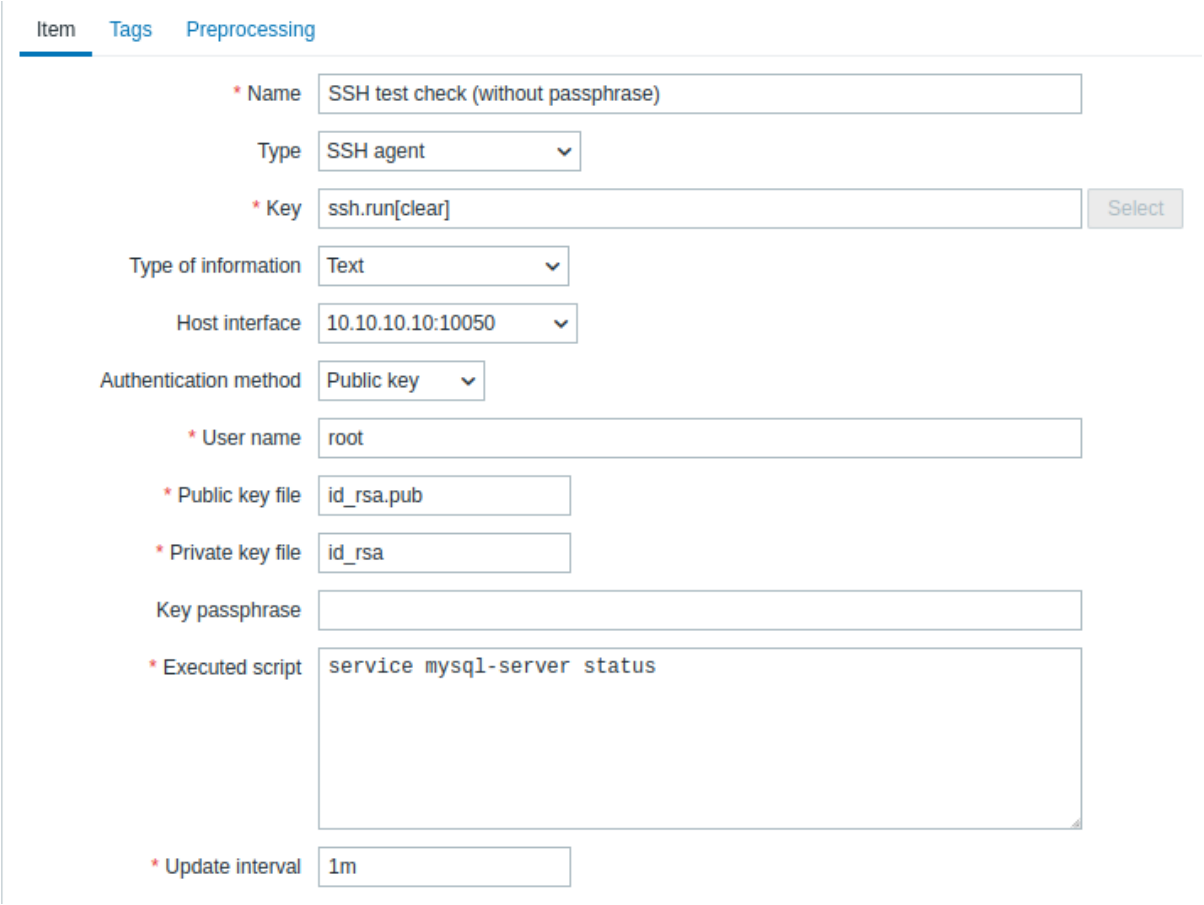
Il est maintenant possible de vérifier la connexion SSH à l'aide de la clé privée par défaut (`/home/zabbix/.ssh/id_rsa`) pour le compte utilisateur *zabbix* :

```
sudo -u zabbix ssh root@10.10.10.10
```

Si la connexion réussit, la partie configuration dans le shell est terminée et la session SSH distante peut être fermée.

Configuration de l'élément

La ou les commandes à exécuter doivent être placées dans le champ *Script exécuté* de la configuration de l'élément. Plusieurs commandes peuvent être exécutées l'une après l'autre en les plaçant sur une nouvelle ligne. Dans ce cas, les valeurs renvoyées seront également formatées sur plusieurs lignes.



Tous les champs obligatoires sont marqués d'un astérisque rouge.

Les champs qui nécessitent des informations spécifiques pour les éléments SSH sont les suivants :

Paramètre	Description	Commentaires
Type	Sélectionnez **SSH agent** ici.	

Paramètre	Description	Commentaires

Clé	Clé d'élément unique (par hôte) au format `ssh.run[unique short description,<ip>,<port>,<encoding>,<ssh options>,<subsystem>]`	**unique short description** est obligatoire et doit être unique pour chaque élément SSH par hôte.
		
		Le port par défaut est 22, et non le port spécifié dans l'interface à laquelle cet élément est affecté.
		
		ssh options permet de transmettre des options SSH supplémentaires au format `key1=value1;key2=value2,value3`. Plusieurs valeurs pour une même clé peuvent être transmises en les séparant par des virgules (dans ce cas, le paramètre doit être mis entre guillemets) ; plusieurs clés d'option peuvent être transmises en les séparant par des points-virgules.
		
		Les clés d'option suivantes sont prises en charge : `KexAlgorithms`, `HostkeyAlgorithms`, `Ciphers`, `MACs`, `PubkeyAcceptedKeyTypes`. La prise en charge de la clé et de la valeur de l'option dépend de la bibliothèque SSH (par exemple, `PubkeyAcceptedKeyTypes` est pris en charge uniquement avec libssh) ; si une option n'est pas prise en charge, une erreur est renvoyée et l'élément devient non pris en charge.
		
		Notez que le signe « + » pour ajouter des paramètres de chiffrement et le signe « ! » pour désactiver des paramètres de chiffrement spécifiques (comme dans GnuTLS et OpenSSL) ne sont pas pris en charge.
		
		Exemples :
		=>
		`ssh.run[KexAlgorithms,127.0.0.1,,,Ciphers=aes128-`
		=>
		`ssh.run[KexAlgorithms,,,,"KexAlgorithms=diffie-h`
		=>
		`ssh.run[PubkeyAcceptedKeyTypes,127.0.0.1,,,Pubke`
		
		subsystem permet de transmettre un sous-système SSH, limitant la connexion SSH aux opérations spécifiques autorisées par le sous-système (par exemple, les transferts de fichiers avec SFTP ou la gestion des périphériques réseau avec NETCONF). Notez que l'utilisation d'un sous-système peut également nécessiter une syntaxe de script spécifique dans le paramètre *Script exécuté*.
		
		Exemples :
		=>
		`ssh.run[SFTPBackup,192.0.2.18,,,,sftp]`
		=>
		`ssh.run[Cisco1234,192.0.2.18,,,,netconf]`

Méthode d'authentification	L'une des méthodes « Mot de passe » ou « Clé publique ».	
Nom d'utilisateur	Nom d'utilisateur (jusqu'à 255 caractères) utilisé pour s'authentifier sur l'hôte distant. Obligatoire.	
Fichier de clé publique	Nom du fichier de clé publique si la *Méthode d'authentification* est « Clé publique ». Obligatoire.	Exemple : *id_rsa.pub* - nom de fichier de clé publique par défaut généré par la commande ssh-keygen.

Paramètre	Description	Commentaires
Fichier de clé privée	Nom du fichier de clé privée si la *Méthode d'authentification* est « Clé publique ». Obligatoire.	Exemple : *id_rsa* - nom de fichier de clé privée par défaut.
Mot de passe ou *Phrase secrète de la clé*	Mot de passe (jusqu'à 255 caractères) utilisé pour l'authentification ou Phrase secrète **si** elle a été utilisée pour la clé privée.	Laissez le champ *Phrase secrète de la clé* vide si aucune phrase secrète n'a été utilisée. Consultez également les problèmes connus concernant l'utilisation des phrases secrètes.

Script exécuté	Commande(s) shell exécutée(s) à l'aide d'une session SSH distante.	La valeur renvoyée par la ou les commandes shell exécutées est limitée à 16 Mo (y compris les espaces blancs finaux qui sont tronqués) ; les limites de la base de données s'appliquent également.
		
		Notez que la bibliothèque libssh2 peut tronquer les scripts exécutables à environ 32 ko.
		
		Exemples :
		`date +%s`
		`systemctl status mysql-server`
		`ps auxww | grep httpd | wc -l`
		
		Exemple (pour le sous-système NETCONF) :
		`<rpc>`
		`<get-software-information/>`
		`</rpc>`
		`]]>]]>`
		`<rpc>`
		`<close-session/>`
		`</rpc>`
		`]]>]]>`

Agent Telnet

Aperçu

Les vérifications Telnet sont effectuées en tant que surveillance sans agent. L'agent Zabbix n'est pas nécessaire pour les vérifications Telnet.

Champs configurables

La ou les commande(s) réelles à exécuter doivent être placées dans le champ **Executed script** de la configuration de l'élément.
 Plusieurs commandes peuvent être exécutées l'une après l'autre en les plaçant sur une nouvelle ligne. Dans ce cas, la valeur renvoyée sera également formatée sur plusieurs lignes.

Caractères pris en charge par lesquels l'invite du shell peut se terminer :

- $
- #
- >
- %

Note:
Une ligne d'invite telnet se terminant par l'un de ces caractères sera supprimée de la valeur renvoyée, mais uniquement pour la première commande de la liste des commandes, c'est-à-dire seulement au début de la session telnet.

Clé	Description
telnet.run[<unique short description>,<ip>,<port>,<encoding>]	Exécuter une commande sur un périphérique distant à l'aide d'une connexion telnet

Attention:
Si une vérification telnet renvoie une valeur contenant des caractères non ASCII et dans un encodage non UTF8, alors le paramètre *<encoding>* de la clé doit être correctement spécifié. Voir la page encoding of returned values pour plus de détails.

agent JMX

Aperçu

La supervision JMX peut être utilisée pour surveiller les compteurs JMX d'une application Java.

La supervision JMX est prise en charge nativement dans Zabbix sous la forme d'un démon Zabbix appelé « Zabbix Java gateway ».

Pour récupérer la valeur d'un compteur JMX particulier sur un hôte, le serveur Zabbix interroge le **Java gateway** de Zabbix, qui utilise à son tour l'API de gestion JMX pour interroger à distance l'application concernée.

Pour plus de détails et la configuration, consultez la section Zabbix Java gateway.

Warning:
La communication entre Java gateway et l'application JMX surveillée ne doit pas être bloquée par un pare-feu.

Activation de la surveillance JMX à distance pour une application Java

Une application Java n'a pas besoin de logiciel supplémentaire installé, mais elle doit être démarrée avec les options de ligne de commande indiquées ci-dessous pour prendre en charge la surveillance JMX à distance.

Au minimum, si vous souhaitez simplement commencer par surveiller une application Java simple sur un hôte local sans sécurité appliquée, démarrez-la avec ces options :

```
java \
-Dcom.sun.management.jmxremote \
-Dcom.sun.management.jmxremote.port=12345 \
-Dcom.sun.management.jmxremote.authenticate=false \
-Dcom.sun.management.jmxremote.ssl=false \
-Dcom.sun.management.jmxremote.registry.ssl=false \
-jar /path/to/your/application.jar
```

Cela fait que Java écoute les connexions JMX entrantes sur le port 12345, depuis l'hôte local uniquement, et lui indique de ne pas exiger d'authentification ni de SSL.

Si vous souhaitez autoriser les connexions sur une autre interface, définissez le paramètre -Djava.rmi.server.hostname sur l'adresse IP de cette interface.

Si vous souhaitez être plus strict en matière de sécurité, de nombreuses autres options Java sont à votre disposition. Par exemple, l'exemple suivant démarre l'application avec un ensemble d'options plus polyvalent et l'ouvre à un réseau plus large, pas seulement à l'hôte local.

```
java \
-Djava.rmi.server.hostname=192.168.3.14 \
-Dcom.sun.management.jmxremote \
-Dcom.sun.management.jmxremote.port=12345 \
-Dcom.sun.management.jmxremote.authenticate=true \
-Dcom.sun.management.jmxremote.password.file=/etc/java-6-openjdk/management/jmxremote.password \
-Dcom.sun.management.jmxremote.access.file=/etc/java-6-openjdk/management/jmxremote.access \
-Dcom.sun.management.jmxremote.ssl=true \
-Dcom.sun.management.jmxremote.registry.ssl=true \
-Djavax.net.ssl.keyStore=$YOUR_KEY_STORE \
-Djavax.net.ssl.keyStorePassword=$YOUR_KEY_STORE_PASSWORD \
-Djavax.net.ssl.trustStore=$YOUR_TRUST_STORE \
-Djavax.net.ssl.trustStorePassword=$YOUR_TRUST_STORE_PASSWORD \
-Dcom.sun.management.jmxremote.ssl.need.client.auth=true \
-jar /path/to/your/application.jar
```

La plupart (sinon la totalité) de ces paramètres peuvent être spécifiés dans `$JRE/lib/management/management.properties` (ou à l'emplacement où se trouve ce fichier sur votre système).

Notez que si vous souhaitez utiliser SSL, vous devez modifier le script startup.sh en ajoutant les options `-Djavax.net.ssl.*` à Java gateway, afin qu'il sache où trouver les magasins de clés et de certificats de confiance.

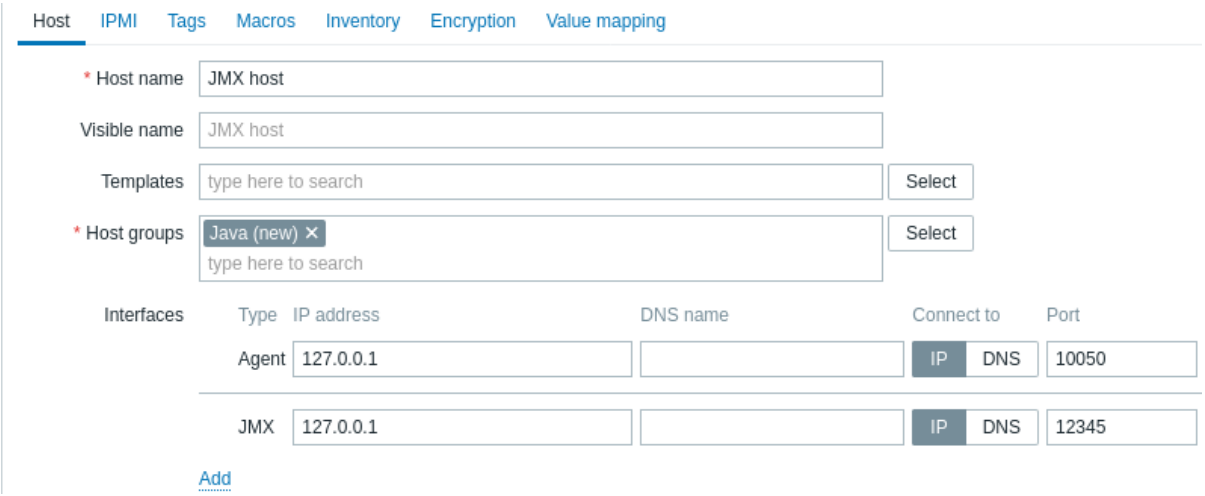
Consultez Monitoring and Management Using JMX pour une description détaillée.

Configuration des interfaces et des éléments JMX dans l'interface Zabbix

Avec la passerelle Java en cours d'exécution, le serveur sachant où trouver la passerelle et une application Java démarrée avec la prise en charge de la surveillance JMX à distance, il est temps de configurer les interfaces et les éléments dans l'interface web de Zabbix.

Configuration de l'interface JMX

Commencez par créer une interface de type JMX sur l'hôte concerné.

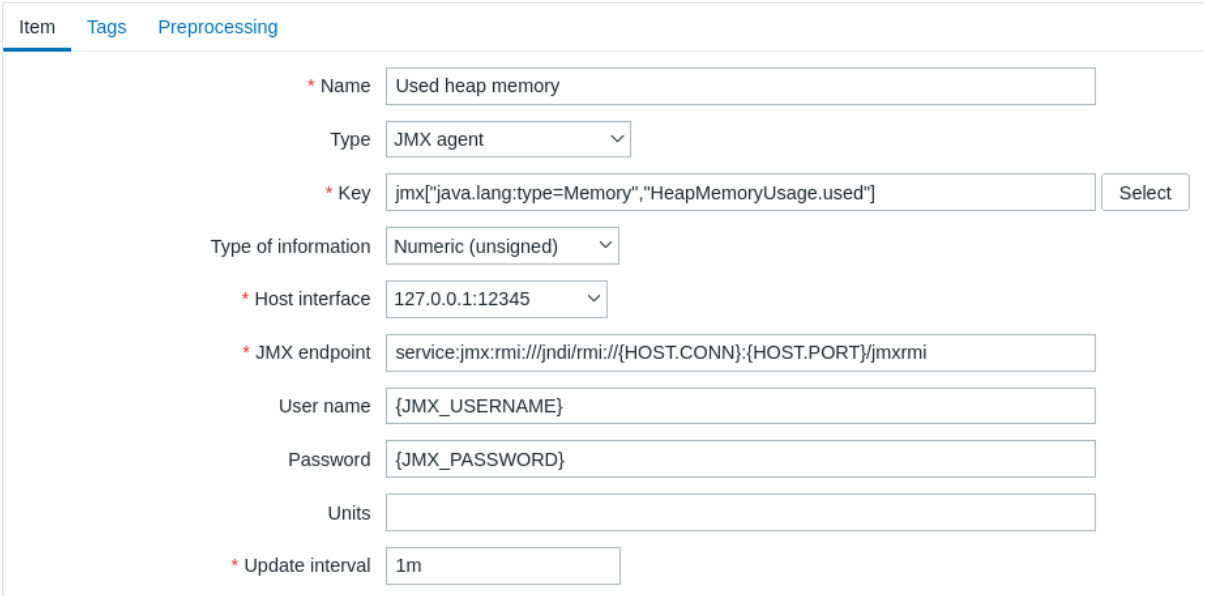


Tous les champs de saisie obligatoires sont marqués d'un astérisque rouge.

Ajout d'un élément agent JMX

Pour chaque compteur JMX qui vous intéresse, vous ajoutez un élément **JMX agent** rattaché à cette interface.

La clé dans la capture d'écran ci-dessous est `jmx["java.lang:type=Memory","HeapMemoryUsage.used"]`.



Tous les champs de saisie obligatoires sont marqués d'un astérisque rouge.

Les champs qui nécessitent des informations spécifiques pour les éléments JMX sont les suivants :

Type	Définissez ici **JMX agent**.

Key	La clé d'élément `jmx[]` contient trois paramètres : **object name** - le nom d'objet d'un MBean **attribute name** - un nom d'attribut MBean avec, en option, des noms de champs de données composites séparés par des points **unique short description** - une description unique qui permet plusieurs éléments JMX avec le même nom d'objet et le même nom d'attribut sur l'hôte (facultatif) Voir ci-dessous pour plus de détails sur les clés d'élément JMX. Vous pouvez découvrir les MBeans et les attributs MBean à l'aide d'un élément `jmx.discovery[]` low-level discovery.
JMX endpoint	Vous pouvez spécifier un point de terminaison JMX personnalisé. Assurez-vous que les paramètres de connexion du point de terminaison JMX correspondent à l'interface JMX. Cela peut être réalisé en utilisant les macros `{HOST.*}` comme dans le point de terminaison JMX par défaut. Les macros `{HOST.*}` et les macros utilisateur sont prises en charge.
User name	Spécifiez le nom d'utilisateur (jusqu'à 255 caractères), si vous avez configuré l'authentification sur votre application Java. Les macros utilisateur sont prises en charge.
Password	Spécifiez le mot de passe (jusqu'à 255 caractères), si vous avez configuré l'authentification sur votre application Java. Les macros utilisateur sont prises en charge.

Si vous souhaitez surveiller un compteur booléen qui vaut soit "true" soit "false", vous devez alors définir le type d'information sur "Numérique (non signé)" et sélectionner l'étape de prétraitement "Boolean to decimal" dans l'onglet Prétraitement. Le serveur stockera les valeurs booléennes respectivement sous la forme 1 ou 0.

Clés d'élément JMX plus en détails

Attributs simples

Un nom d'objet MBean n'est rien d'autre qu'une chaîne que vous définissez dans votre application Java. En revanche, un nom d'attribut peut être plus complexe. Si un attribut renvoie un type de données primitif (un entier, une chaîne, etc.), il n'y a rien de particulier à craindre, la clé ressemblera à ceci :

```
jmx[com.example:Type=Hello,weight]
```

Dans cet exemple, le nom de l'objet est "com.example:Type=Hello", le nom de l'attribut est "weight", et le type de valeur renvoyé devrait probablement être "Numérique (flottant)".

Attributs renvoyant des données composites

Cela devient plus compliqué lorsque votre attribut renvoie des données composites. Par exemple : votre nom d'attribut est "apple" et il renvoie un hachage représentant ses paramètres, comme "poids", "couleur", etc. Votre clé peut ressembler à ceci :

```
jmx[com.example:Type=Hello,apple.weight]
```

C'est ainsi qu'un nom d'attribut et une clé de hachage sont séparés, en utilisant le symbole point. De même, si un attribut renvoie des données composites imbriquées, les parties sont séparées par un point :

```
jmx[com.example:Type=Hello,fruits.apple.weight]
```

Attributs renvoyant des données tabulaires

Les attributs de données tabulaires consistent en un ou plusieurs attributs composites. Si un tel attribut est spécifié dans le paramètre de nom d'attribut, cette valeur d'élément renverra la structure complète de l'attribut au format JSON. Les valeurs d'élément individuelles à l'intérieur de l'attribut de données tabulaires peuvent être récupérées à l'aide du prétraitement.

Exemple d'attribut de données tabulaire :

```
jmx[com.example:type=Hello,foodinfo]
```

Valeur de l'article :

```
[
  {
    "a": "apple",
    "b": "banana",
    "c": "cherry"
  },
  {
```

```
    "a": "potato",
    "b": "lettuce",
    "c": "onion"
  }
]
```

Problème avec les points

Jusqu'ici, tout va bien. Mais que faire si un nom d'attribut ou une clé de hachage contient un point ? Voici un exemple :

`jmx[com.example:Type=Hello,all.fruits.apple.weight]`

C'est un problème. Comment indiquer à Zabbix que le nom de l'attribut est "all.fruits", et pas seulement "all" ? Comment distinguer un point qui fait partie du nom d'un point qui sépare un nom d'attribut et des clés de hachage ?

C'est possible, il suffit d'échapper les points qui font partie du nom avec une barre oblique inverse :

`jmx[com.example:Type=Hello,all\.fruits.apple.weight]`

De la même manière, si votre clé de hachage contient un point, vous devez l'échapper :

`jmx[com.example:Type=Hello,all\.fruits.apple.total\.weight]`

Autres problèmes

Un caractère barre oblique inverse dans un nom d'attribut doit être échappé :

`jmx[com.example:type=Hello,c:\\documents]`

Pour la gestion de tout autre caractère spécial dans une clé d'élément JMX, veuillez consulter la section sur le format des clés d'élément section.

C'est en fait tout ce qu'il y a à savoir. Bonne surveillance JMX !

Types de données non primitifs

Il est possible de travailler avec des MBeans personnalisés renvoyant des types de données non primitifs, qui redéfinissent la méthode **toString()**.

Utilisation d'un endpoint personnalisé avec JBoss EAP 6.4

Les endpoints personnalisés permettent de travailler avec différents protocoles de transport autres que le RMI par défaut.

Pour illustrer cette possibilité, essayons de configurer la surveillance de JBoss EAP 6.4 comme exemple. Tout d'abord, faisons quelques hypothèses :

- Vous avez déjà installé le Java gateway de Zabbix. Si ce n'est pas le cas, vous pouvez le faire conformément à la documentation.
- Le serveur Zabbix et le Java gateway sont installés avec le préfixe `/usr/local/`
- JBoss est déjà installé dans `/opt/jboss-eap-6.4/` et s'exécute en mode autonome
- Nous supposerons que tous ces composants fonctionnent sur le même hôte
- Le pare-feu et SELinux sont désactivés (ou configurés en conséquence)

Faisons quelques réglages simples dans `zabbix_server.conf` :

```
JavaGateway=127.0.0.1
StartJavaPollers=5
```

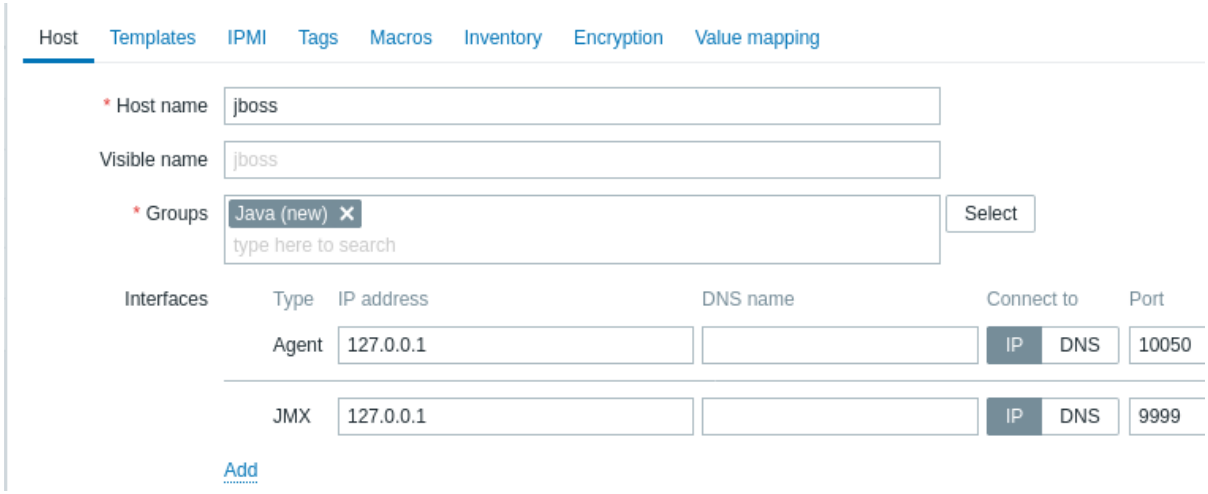
Et dans le fichier de configuration `zabbix_java/settings.sh` (ou `zabbix_java_gateway.conf`) :

```
START_POLLERS=5
```

Vérifiez que JBoss écoute sur son port de gestion standard :

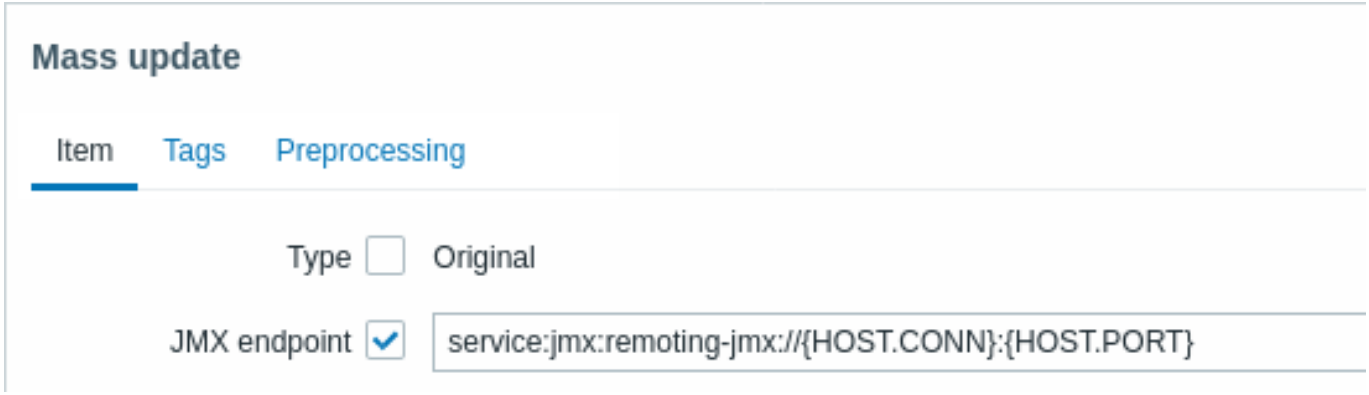
```
$ netstat -natp | grep 9999
tcp        0      0 127.0.0.1:9999          0.0.0.0:*               LISTEN      10148/java
```

Créons maintenant un hôte avec l'interface JMX 127.0.0.1:9999 dans Zabbix.



Comme nous savons que cette version de JBoss utilise le protocole JBoss Remoting au lieu de RMI, nous pouvons mettre à jour en masse le paramètre d'endpoint JMX pour les éléments de notre modèle JMX en conséquence :

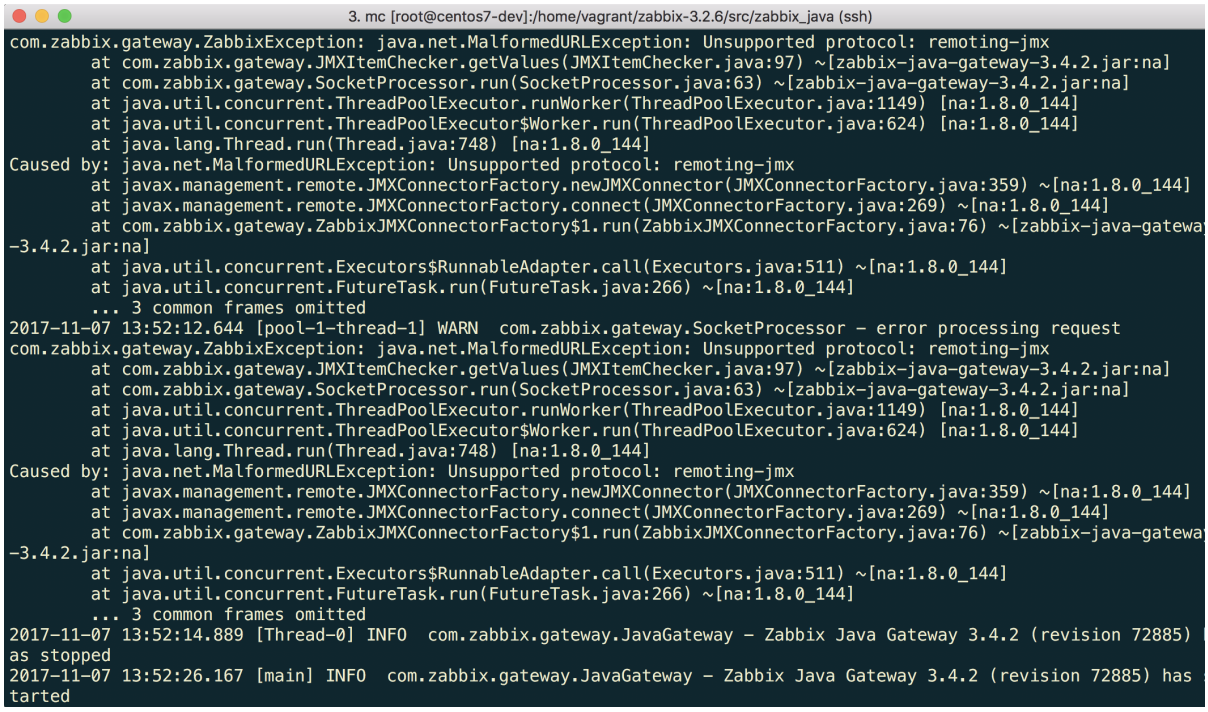
`service:jmx:remoting-jmx://{HOST.CONN}:{HOST.PORT}`



Mettons à jour le cache de configuration :

```
/usr/local/sbin/zabbix_server -R config_cache_reload
```

Notez que vous pouvez d'abord rencontrer une erreur.



"Unsupported protocol: remoting-jmx" signifie que le Java gateway ne sait pas comment fonctionner avec le protocole spécifié. Cela peut être corrigé en créant un fichier `~/needed_modules.txt` avec le contenu suivant :

```
jboss-as-remoting
jboss-logging
jboss-logmanager
jboss-marshalling
jboss-remoting
jboss-sasl
jcl-over-slf4j
jul-to-slf4j-stub
log4j-jboss-logmanager
remoting-jmx
slf4j-api
xnio-api
xnio-nio
```

puis en exécutant la commande :

```
for i in $(cat ~/needed_modules.txt); do find /opt/jboss-eap-6.4 -iname "${i}*.jar" -exec cp '{}' /usr/loc
```

Ainsi, le Java gateway disposera de tous les modules nécessaires pour fonctionner avec jmx-remoting. Il ne reste plus qu'à redémarrer le Java gateway, attendre un peu et, si tout a été fait correctement, constater que les données de surveillance JMX commencent à arriver dans Zabbix (voir aussi : Dernières données).

Élément calculé

Présentation générale

Ce type d'élément vous permet d'effectuer des calculs sur les métriques qui existent déjà. Un élément calculé ne collecte aucune donnée par lui-même.

Par exemple, vous pouvez vouloir :

- Compter deux valeurs.
- Calculer la moyenne de plusieurs valeurs d'éléments.
- Calculer la valeur totale pour un groupe d'éléments.

La valeur obtenue est stockée dans la base de données Zabbix comme pour tout autre élément ; les valeurs d'historique et de tendance sont stockées, et des graphiques peuvent être générés.

Le calcul est basé sur une formule. La syntaxe de la formule est la même que celle des expressions de déclencheur.

En fait, il est possible de conserver des déclencheurs simples en plaçant une logique complexe de transformation des données dans un élément calculé, puis en faisant référence à cet élément calculé dans un déclencheur à l'aide de la clé de l'élément.

Deux possibilités existent pour spécifier la source de données des calculs :

1. Spécifier les éléments dans la formule.
2. Effectuer une agrégation de données basée sur un filtre qui sélectionne les éléments.

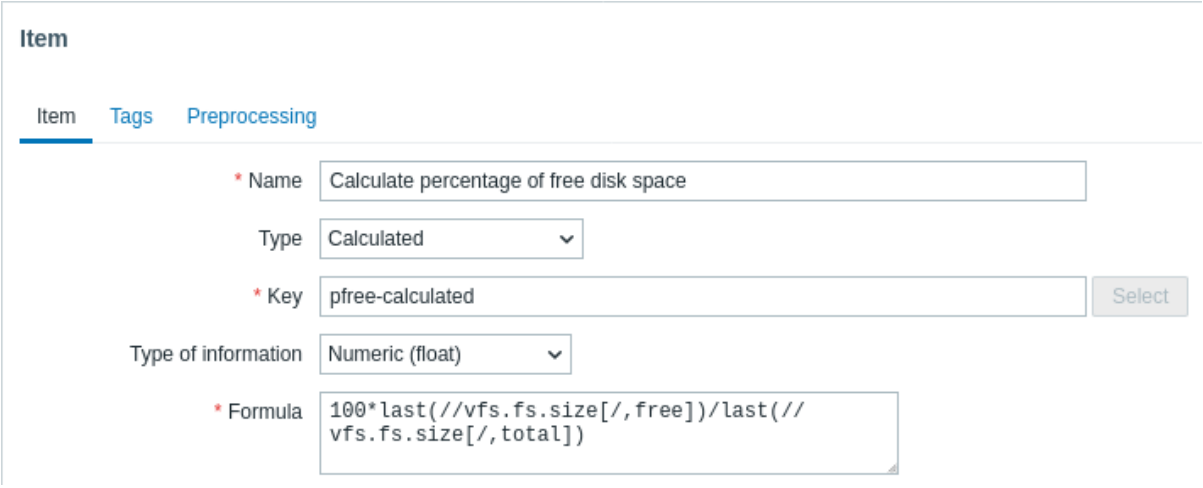
La deuxième option vous permet de calculer des moyennes de groupe, des totaux, etc.

Tous les calculs sont effectués par le serveur Zabbix et pris en charge par les processus `history poller` ; le proxy et les agents ne sont absolument pas impliqués. Notez que l'utilisation intensive d'éléments calculés avec de longues périodes peut affecter les performances du serveur.

Veuillez noter que les éléments calculés peuvent accéder à l'historique d'autres éléments, y compris aux éléments auxquels l'utilisateur qui crée l'élément calculé n'a pas accès. Si ce comportement n'est pas souhaité, vous pouvez désactiver les éléments calculés en définissant le paramètre de configuration `StartHistoryPollers` sur 0.

Champs configurables

Pour utiliser des éléments calculés, sélectionnez le type d'élément **Calculated**.



La **clé** est un identifiant unique de l'élément (par hôte). Vous pouvez créer n'importe quel nom de clé à l'aide des symboles pris en charge. La clé vous permet de référencer l'élément calculé dans les déclencheurs.

La définition du calcul doit être saisie dans le champ **Formula**.

Il n'existe aucun lien entre la formule et la clé. Les paramètres de la clé ne sont en aucun cas utilisés dans la formule.

Contrairement aux expressions de déclencheur, Zabbix traite les éléments calculés selon l'intervalle de mise à jour de l'élément, et non lors de la réception d'une nouvelle valeur.

Notez que si vous sélectionnez *Numeric (unsigned)* comme type d'information et que le résultat du calcul est une valeur flottante, celle-ci est tronquée en entier.

Formule de calcul simple

La syntaxe d'une formule simple est :

`fonction(/hôte/clé,<paramètre1>,<paramètre2>,...)`

où :

Élément	Description
`function`	L'une des fonctions prises en charge : last, min, max, avg, count, etc.
`host`	Hôte de l'élément utilisé pour le calcul. L'hôte actuel peut être omis (c'est-à-dire comme dans `function(//key,parameter,...)`).
`key`	Clé de l'élément utilisé pour le calcul. Les éléments renvoyant des valeurs avec le type de données binaire ou JSON ne sont pas pris en charge. Si vous modifiez la clé d'élément d'un élément référencé, la clé doit être mise à jour manuellement dans la formule. Tous les éléments référencés par des fonctions d'historique doivent exister et collecter des données.
`parameter(s)`	Paramètres de la fonction, si nécessaire. Les suffixes de temps et les suffixes de taille mémoire sont pris en charge.

Une formule plus complexe peut utiliser une combinaison de fonctions, d'opérateurs et de parenthèses.

Voir des exemples de formules d'éléments calculés :

- (1) Trafic total sur l'interface
- (2) Pourcentage d'espace disque libre
- (3) Pourcentage de trafic entrant
- (4) Nombre moyen de valeurs traitées

Certaines variables peuvent être transmises à la formule à l'aide de macros utilisateur. Notez que les macros utilisateur :

- seront résolues - si elles sont utilisées pour référencer un paramètre de fonction, un paramètre de filtre d'élément ou une constante.
- **ne seront pas** résolues - si elles référencent une fonction, un nom d'hôte, une clé d'élément, un paramètre de clé d'élément ou un opérateur.

La comparaison avec des chaînes est autorisée.

S'il n'y a pas de données récentes dans le cache et qu'aucune période d'interrogation n'est définie dans la fonction, Zabbix remontera par défaut jusqu'à une semaine dans le passé pour interroger la base de données à la recherche de valeurs historiques.

Un élément calculé peut devenir non pris en charge dans plusieurs cas :

- élément référencé
  - introuvable
  - désactivé, ou appartenant à un hôte désactivé
  - non pris en charge (sauf avec la fonction nodata() et les opérateurs avec des valeurs inconnues)
- aucune donnée pour calculer une fonction
- division par zéro
- syntaxe incorrecte utilisée

Agrégations de données

Les agrégations de données sont basées sur un filtre qui sélectionne plusieurs éléments. Les agrégations de données sont également un type d'élément calculé, mais vous utilisez une syntaxe légèrement différente dans la formule.

Pour récupérer des agrégats, utilisez l'une des fonctions d'agrégation prises en charge : `avg`, `max`, `min`, `sum`, etc. Ajoutez ensuite la fonction **foreach** comme seul paramètre ainsi que son filtre d'élément pour sélectionner les éléments requis :

```
aggregate_function(function_foreach(/host/key?[group="host group"],timeperiod))
```

Une fonction **foreach** (par exemple *avg_foreach*, *count_foreach*, etc.) renvoie une valeur agrégée pour chaque élément sélectionné. Les éléments sont sélectionnés à l'aide du filtre d'élément (`/host/key?[group="host group"]`), à partir de l'historique des éléments. Pour plus de détails, voir fonctions foreach.

Si certains éléments n'ont pas de données pour la période demandée, ils sont ignorés dans le calcul. Si aucun élément n'a de données, la fonction renverra une erreur.

Vous pouvez également lister plusieurs éléments comme paramètres de l'agrégation :

```
aggregate_function(function(/host/key,parameter),function(/host2/key2,parameter),...)
```

Notez que `function` doit ici être une fonction d'historique/tendance.

Voir des exemples de calculs agrégés :

- (5) Trafic entrant total pour l'hôte
- (6) Espace disque total pour le groupe
- (7) Charge processeur moyenne pour le groupe
- (8) Charge processeur moyenne pour le groupe et le tag
- (9) Nombre moyen de valeurs traitées
- (10) Nombre moyen de requêtes pour le groupe
- (11) Nombre total de vérifications DNS réussies
- (12) Nombre total d'éléments non pris en charge pour le groupe
- (13) Statistiques d'interface pour le groupe

Les macros utilisateur et les macros de découverte de bas niveau sont prises en charge dans :

- les paramètres de clé d'élément
- les paramètres de fonction
- les conditions de filtre (nom du groupe d'hôtes et nom du tag)
- les constantes d'expression

Un calcul agrégé peut devenir non pris en charge si :

- aucun des éléments référencés n'est trouvé (ce qui peut se produire si la clé d'élément est incorrecte, si aucun des éléments n'existe ou si tous les groupes inclus sont incorrects)
- aucune donnée n'est disponible pour calculer une fonction

Exemples d'éléments calculés

(1) Trafic total sur l'interface

```
last(//net.if.in[eth0,bytes])+last(//net.if.out[eth0,bytes])
```

Calcule la bande passante totale sur eth0.

(2) Pourcentage d'espace disque libre

```
100*last(//vfs.fs.size[/,free])/last(//vfs.fs.size[/,total])
```

Zabbix prendra les dernières valeurs des espaces disque libre et total (sur /) et calculera le pourcentage selon la formule donnée.

(3) Pourcentage du trafic entrant

```
100*last(//net.if.in[eth0,bytes])/(last(//net.if.in[eth0,bytes])+last(//net.if.out[eth0,bytes]))
```

Calcule le pourcentage du trafic entrant par rapport au trafic total.

(4) Nombre moyen de valeurs traitées

```
avg(/Zabbix server/zabbix[wcache,values],10m)
```

Calcule une moyenne sur 10 minutes du nombre de valeurs traitées par Zabbix.

(5) Trafic entrant total pour l'hôte

```
sum(last_foreach(/host/net.if.in[*]))
```

Calcule le total de tous les éléments correspondant à `net.if.in[*]` sur l'hôte.

(6) Espace disque total pour le groupe

```
sum(last_foreach(/*/vfs.fs.size[/,total]?[group="MySQL Servers"]))
```

Calcule l'espace disque total pour tous les `MySQL Servers`.

(7) Charge processeur moyenne pour le groupe

```
avg(last_foreach(/*/system.cpu.load[,avg1]?[group="MySQL Servers"]))
```

Calcule la charge processeur moyenne pour tous les `MySQL Servers`.

(8) Charge processeur moyenne pour le groupe et le tag

```
avg(last_foreach(/*/system.cpu.load?[(group="Servers A" or group="Servers B" or group="Servers C") and (ta
```

Calcule la charge CPU moyenne sur tous les hôtes de plusieurs groupes d'hôtes ayant des tags spécifiques.

(9) Nombre moyen de requêtes pour le groupe

```
avg(avg_foreach(/*/mysql.qps?[group="MySQL Servers"],5m))
```

Calcule la moyenne sur 5 minutes du nombre de requêtes par seconde pour `MySQL Servers`.

(10) Nombre total de vérifications DNS réussies

```
sum(last_foreach(/*/net.dns[*,*,*]))
```

Calcule le nombre total de vérifications DNS réussies sur tous les hôtes. Le formulaire d'élément affiché est `net.dns[192.0.2.1,example.c` à titre d'exemple d'une clé possible.

Notez que les caractères génériques doivent correspondre au nombre de paramètres dans la clé (ici `net.dns` comporte trois paramètres: ip, name, type).

(11) Nombre total d'éléments non pris en charge pour le groupe

```
sum(last_foreach(/*/zabbix[host,,items_unsupported]?[group="Zabbix servers"]))
```

Calcule le nombre total d'éléments non pris en charge pour `Zabbix servers`.

(12) Statistiques de l'interface pour le groupe

```
sum(last_foreach(/*/net.if.out[eth0,bytes]?[group="video"])) / sum(last_foreach(/*/nginx_stat.sh[active]?[
```

Calcule les statistiques de l'interface pour le groupe `video`.

Exemples de syntaxe correcte/incorrecte

Les expressions (y compris les appels de fonction) ne peuvent pas être utilisées comme paramètres d'historique, de tendance ou de foreach fonction. Cependant, ces fonctions peuvent elles-mêmes être utilisées dans les paramètres d'autres fonctions (non historiques).

Expression	Exemple

Valide	`avg(last(/host/key1),last(/host/key2)*10,last(/host/key1)*100)`
	
	`max(avg(avg_foreach(/*/system.cpu.load?[group="Servers A"],5m)),avg(avg_foreach(/*/system.cpu.load?[group="Servers B"],5m)),avg(avg_foreach(/*/system.cpu.load?[group="Servers C"],5m)))`

Invalide	`sum(/host/key,10+2)`
	
	`sum(/host/key, avg(10,2))`
	
	`sum(/host/key,last(/host/key2))`

Notez que dans une expression telle que :

```
sum(sum_foreach(//resptime[*],5m))/sum(count_foreach(//resptime[*],5m))
```

(où `resptime[*]` est un exemple de modèle de clé d'élément correspondant aux clés `resptime[1]`, `resptime[2]`, etc., utilisé pour obtenir le temps de réponse de plusieurs nœuds actifs d'un cluster), il n'est pas garanti que les deux parties de l'équation disposent toujours du même ensemble de valeurs. Pendant l'évaluation d'une partie de l'expression, une nouvelle valeur pour la période demandée peut arriver ; l'autre partie de l'expression disposera alors d'un ensemble de valeurs différent.

Élément dépendant

Aperçu

Il existe des situations où un élément collecte plusieurs métriques à la fois, ou il est même plus pertinent de collecter simultanément des métriques liées, par exemple :

- l'utilisation du processeur par cœur individuel
- le trafic réseau entrant/sortant/total

Pour permettre la collecte groupée de métriques et leur utilisation simultanée dans plusieurs éléments liés, Zabbix prend en charge les éléments dépendants. Les éléments dépendants dépendent de l'élément maître qui collecte leurs données simultanément, en une seule requête. Une nouvelle valeur pour l'élément maître renseigne automatiquement les valeurs des éléments dépendants. Les éléments dépendants ne peuvent pas avoir un intervalle de mise à jour différent de celui de l'élément maître.

Les options de prétraitement de Zabbix peuvent être utilisées pour extraire la partie nécessaire à l'élément dépendant à partir des données de l'élément maître.

Le prétraitement est géré par un processus `preprocessing manager`, ainsi que par des threads de travail qui exécutent les étapes de prétraitement. Toutes les valeurs avec prétraitement, reçues de différents collecteurs de données, passent par le gestionnaire de prétraitement avant d'être ajoutées au cache d'historique. Une communication IPC basée sur des sockets est utilisée entre les collecteurs de données (pollers, trappers, etc.) et le processus de prétraitement.

Le serveur Zabbix ou le proxy Zabbix (si l'hôte est supervisé par un proxy) effectuent les étapes de prétraitement et traitent les éléments dépendants.

Les éléments de tout type, y compris les éléments dépendants, peuvent être définis comme éléments maîtres. Des niveaux supplémentaires d'éléments dépendants peuvent être utilisés pour extraire des parties plus petites de la valeur d'un élément dépendant existant.

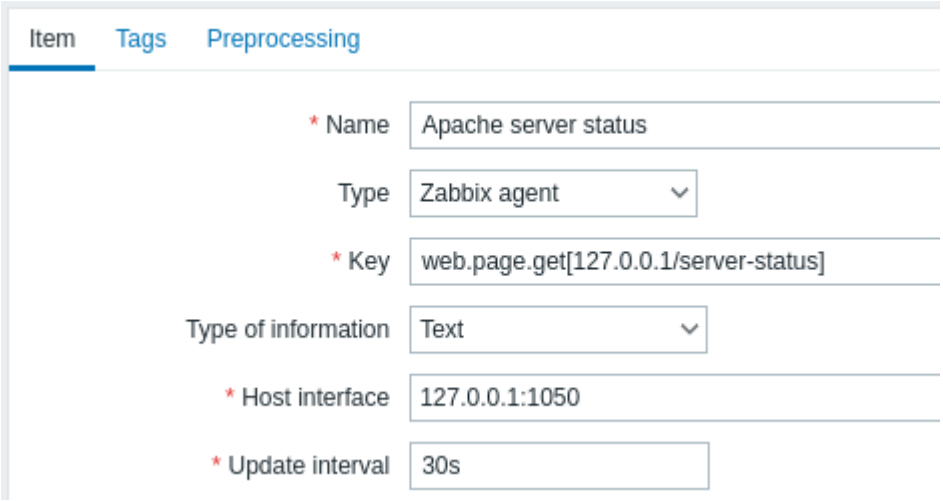
Limitations

- Seules les dépendances sur le même hôte (modèle) sont autorisées
- Un prototype d'élément peut dépendre d'un autre prototype d'élément ou d'un élément standard du même hôte
- Un élément dépendant sur un hôte avec un élément maître provenant d'un modèle ne sera pas exporté en XML

Configuration d'un élément

Un élément dépendant dépend de son élément maître pour les données. C'est pourquoi l'**élément maître** doit être configuré (ou exister) en premier:

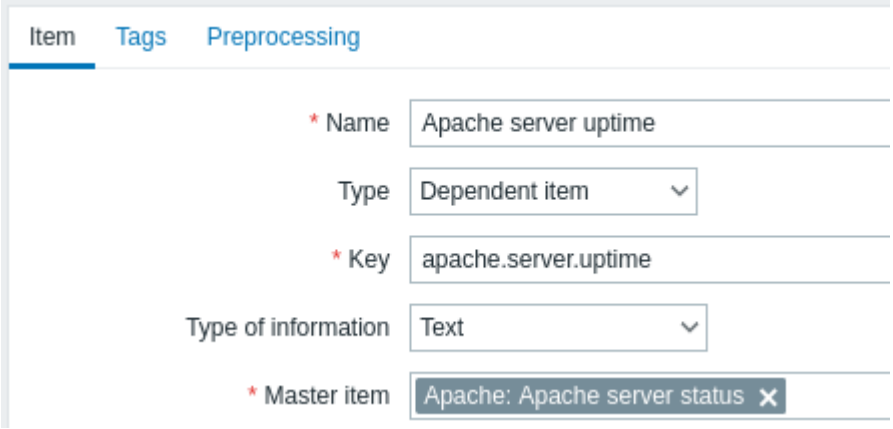
1. Allez dans *Collecte de données* > *Hôtes*.
2. Cliquez sur *Éléments* dans la ligne de l'hôte.
3. Cliquez sur *Créer un élément*.
4. Saisissez les paramètres de l'élément dans le formulaire.



Tous les champs obligatoires sont marqués d'un astérisque rouge.

Cliquez sur *Ajouter* pour enregistrer l'élément maître.

Vous pouvez ensuite configurer un **élément dépendant**.

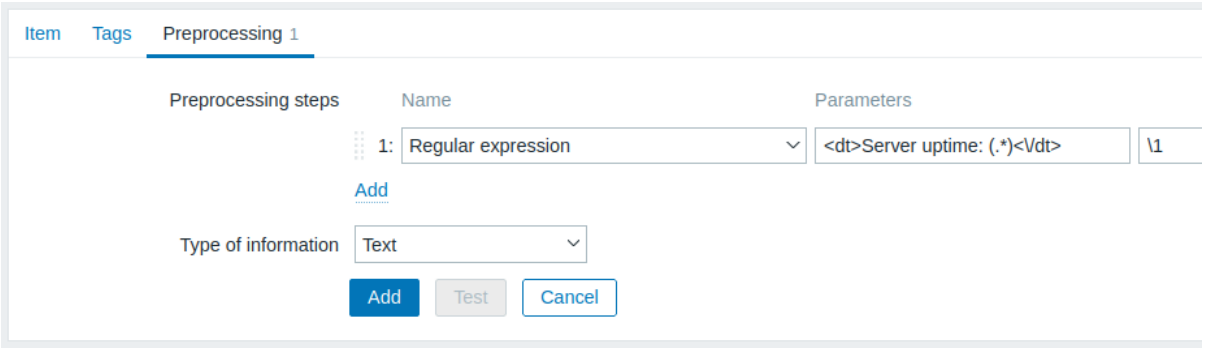


Tous les champs obligatoires sont marqués d'un astérisque rouge.

Les champs qui nécessitent des informations spécifiques pour les éléments dépendants sont:

Type	Sélectionnez **Élément dépendant** ici.
Key	Saisissez une clé qui sera utilisée pour reconnaître l'élément.
Master item	Sélectionnez l'élément maître. La valeur de l'élément maître sera utilisée pour renseigner la valeur de l'élément dépendant.
Type of information	Sélectionnez le type d'information qui correspondra au format des données qui seront stockées.

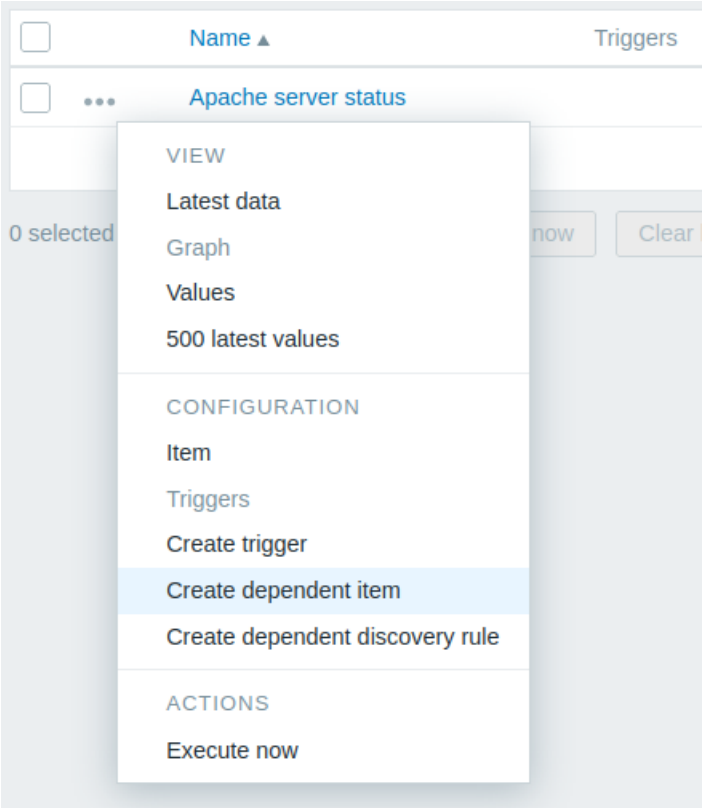
Vous pouvez utiliser le prétraitement de la valeur de l'élément pour extraire la partie requise de la valeur de l'élément maître.



Sans prétraitement, la valeur de l'élément dépendant sera exactement la même que celle de l'élément maître.

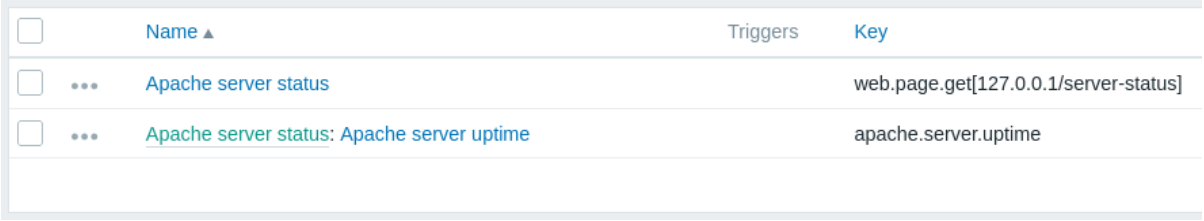
Cliquez sur *Ajouter* pour enregistrer l'élément dépendant.

Un raccourci permettant de créer plus rapidement un élément dépendant est accessible en cliquant sur le bouton ••• dans la liste des éléments, puis en sélectionnant *Créer un élément dépendant*.



Affichage

Dans la liste d'éléments, les éléments dépendants sont affichés avec leur nom d'élément principal comme préfixe.



Si un élément principal est supprimé, tous ses éléments dépendants le sont également.

Élément de script

Aperçu

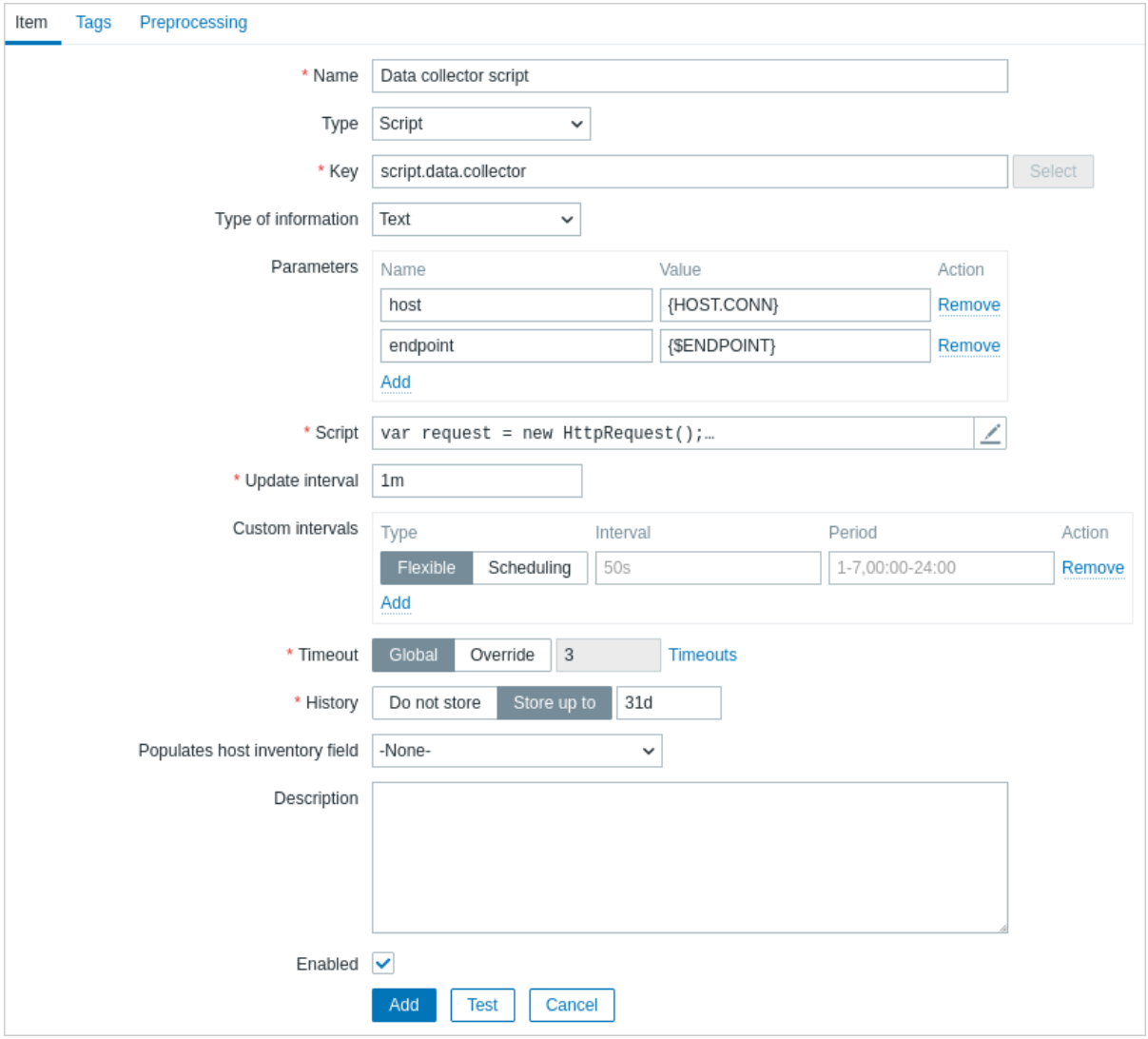
Les éléments de type script peuvent être utilisés pour collecter des données en exécutant un code JavaScript défini par l'utilisateur, avec la possibilité de récupérer des données via HTTP/HTTPS. En plus du script, une liste facultative de paramètres (paires de nom et de valeur) ainsi qu'un délai d'expiration peuvent être spécifiés.

Ce type d'élément peut être utile dans des scénarios de collecte de données nécessitant plusieurs étapes ou une logique complexe. Par exemple, un élément de type script peut être configuré pour effectuer un appel HTTP, puis traiter d'une certaine manière les données reçues lors de la première étape, et transmettre la valeur transformée au second appel HTTP.

Les éléments de type script sont traités par les pollers du serveur ou du proxy Zabbix.

Configuration

Dans le champ *Type* du formulaire de configuration d'un élément, sélectionnez Script, puis renseignez les champs requis.



Tous les champs de saisie obligatoires sont marqués d'un astérisque rouge.

Les champs qui nécessitent des informations spécifiques pour les éléments Script sont les suivants :

Field	Description
Key	Saisissez une clé unique qui sera utilisée pour identifier l'élément.
Parameters	Spécifiez les variables à transmettre au script sous forme de paires attribut-valeur. Les macros utilisateur sont prises en charge. Pour voir quelles macros intégrées sont prises en charge, recherchez « Script-type item » dans le tableau des macros prises en charge.
Script	Saisissez le code JavaScript dans l'éditeur modal qui s'ouvre lorsque vous cliquez dans le champ du paramètre ou sur l'icône en forme de crayon à côté. Ce code doit fournir la logique permettant de renvoyer la valeur de la métrique. Le code a accès à tous les paramètres ainsi qu'à tous les objets JavaScript supplémentaires ajoutés par Zabbix. Voir aussi : Guide JavaScript.
Timeout	Délai d'expiration de l'exécution JavaScript (1-600s ; en cas de dépassement, une erreur sera renvoyée). Notez que, selon le script, le déclenchement du délai d'expiration peut prendre plus de temps. Pour plus d'informations sur le paramètre *Timeout*, consultez les attributs généraux des éléments.

Exemples

Collecte de données simples

Collecter le contenu de *https://www.example.com/release_notes*:

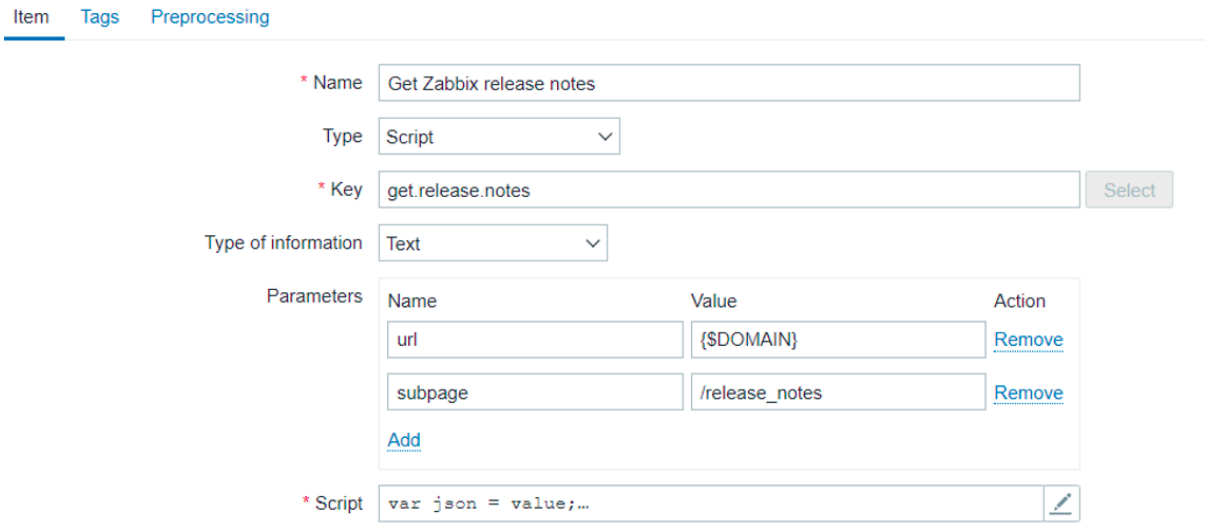
- Créer un élément de type "Script".
- Dans le champ *Script*, saisir :

```
var request = new HttpRequest();
return request.get("https://www.example.com/release_notes");
```

Collecte de données avec des paramètres

Collectez le contenu d'une page spécifique et utilisez des paramètres :

- Créez un élément de type "Script" avec deux paramètres :
  - **url : {$DOMAIN}** (la macro utilisateur {$DOMAIN} doit être définie, de préférence au niveau de l'hôte)
  - **subpage : /release_notes**



- Dans le champ *Script*, saisissez :

```
var obj = JSON.parse(value);
var url = obj.url;
var subpage = obj.subpage;
var request = new HttpRequest();
return request.get(url + subpage);
```

Requêtes HTTP multiples

Collecter le contenu de *https://www.example.com* et *https://www.example.com/release_notes* :

- Créer un élément de type "Script".
- Dans le champ *Script*, saisir :

```
var request = new HttpRequest();
return request.get("https://www.example.com") + request.get("https://www.example.com/release_notes");
```

Journalisation

Ajouter l'entrée "Log test" au fichier de log du serveur Zabbix et recevez la valeur de l'élément "1" en retour :

- Créer un élément de type "Script".
- Dans le champ *Script*, saisir :

```
Zabbix.log(3, 'Log test');
return 1;
```

Élément navigateur

Aperçu

Les éléments de navigateur permettent de surveiller des sites web et des applications web complexes à l'aide d'un navigateur.

> **Attention:**
> La prise en charge des éléments de navigateur est actuellement expérimentale.

Les éléments de navigateur collectent des données en exécutant du code JavaScript défini par l'utilisateur et en récupérant des données via HTTP/HTTPS. Cet élément peut simuler des actions liées au navigateur telles que cliquer, saisir du texte, naviguer dans des pages web, ainsi que d'autres interactions utilisateur avec des sites web ou des applications web.

En plus du script, une liste facultative de paramètres (paires nom et valeur) et un délai d'attente peuvent être spécifiés.

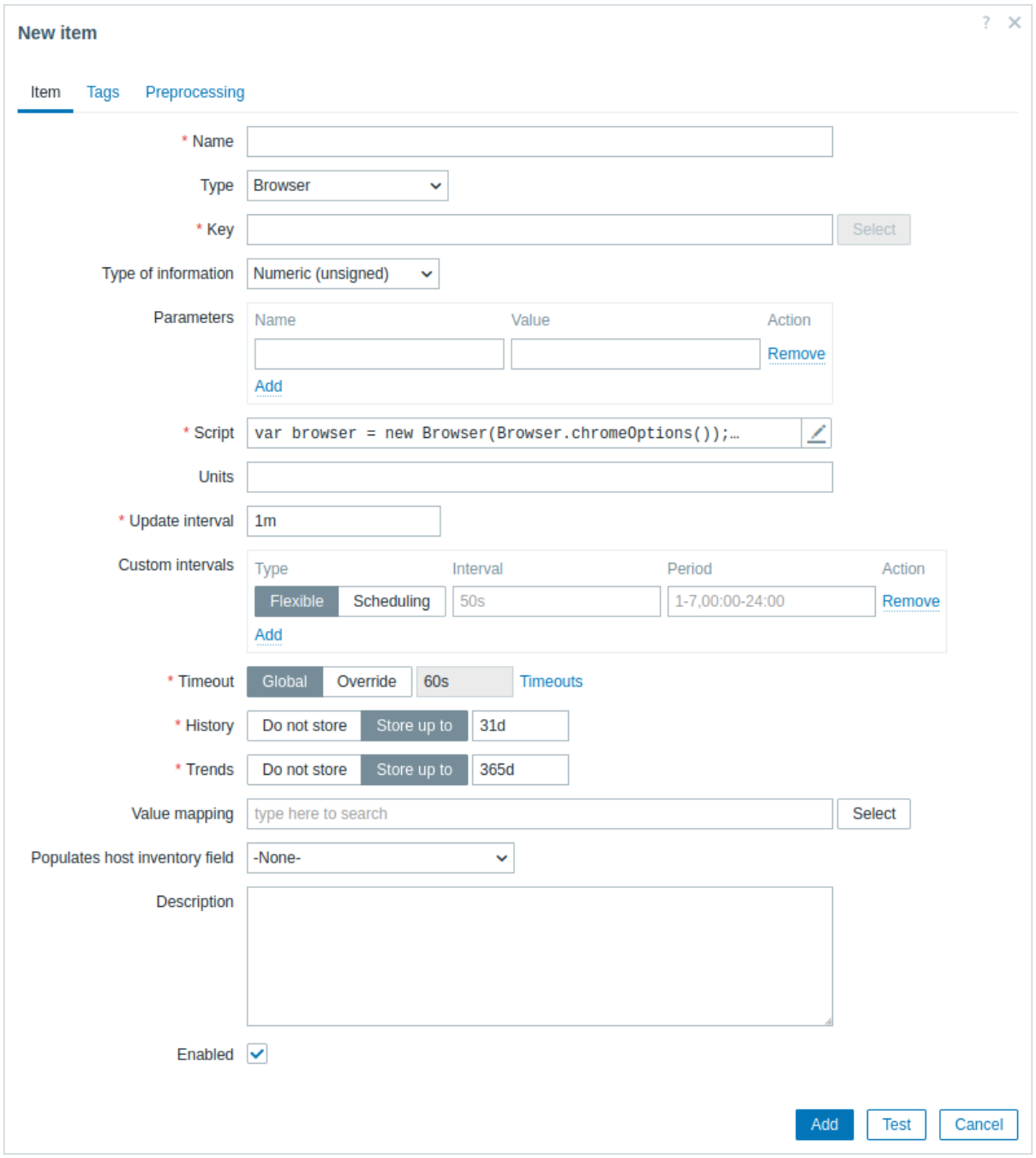
Attention:
L'élément implémente partiellement la norme W3C WebDriver avec Selenium Server ou un WebDriver simple (par exemple, ChromeDriver) comme point de terminaison de test web. Pour que l'élément fonctionne, définissez le paramètre `WebDriverURL` dans le fichier de configuration du serveur/proxy Zabbix (si vous utilisez ChromeDriver, consultez Security Considerations). Pour de meilleures performances, envisagez d'utiliser un serveur dédié pour l'environnement de test web.

Les vérifications des éléments de navigateur sont exécutées et traitées par les collecteurs de navigateur du serveur ou du proxy Zabbix. Si nécessaire, vous pouvez ajuster le nombre d'instances pré-forkées des collecteurs de navigateur dans le paramètre `StartBrowserPollers` du fichier de configuration du serveur/proxy Zabbix.

Pour surveiller des sites web et des applications web complexes, le modèle Website by Browser est disponible en tant que modèle prêt à l'emploi.

Configuration

Dans le champ *Type* du formulaire de configuration d'un élément, sélectionnez Browser, puis renseignez les champs requis.



Tous les champs de saisie obligatoires sont marqués d'un astérisque rouge.

Les champs nécessitant des informations spécifiques pour les éléments Browser sont les suivants :

Field	Description
Key	Saisissez une clé unique qui sera utilisée pour identifier l'élément.
Parameters	Spécifiez les variables à transmettre au script sous forme de paires attribut-valeur. Les macros utilisateur sont prises en charge. Pour voir quelles macros intégrées sont prises en charge, recherchez « Browser-type item » dans le tableau des macros prises en charge.
Script	Saisissez le code JavaScript dans l'éditeur modal qui s'ouvre lorsque vous cliquez dans le champ du paramètre ou sur l'icône en forme de crayon à côté. Ce code doit fournir la logique permettant de renvoyer la valeur de la métrique. Le code a accès à tous les paramètres, à tous les objets JavaScript supplémentaires et aux objets JavaScript des éléments Browser ajoutés par Zabbix. Voir aussi : Guide JavaScript.
Timeout	Délai d'expiration de l'exécution JavaScript (1-600s ; en cas de dépassement, une erreur sera renvoyée). Notez que, selon le script, le déclenchement du délai d'expiration peut prendre plus de temps. Pour plus d'informations sur le paramètre *Timeout*, consultez les attributs généraux des éléments.

Exemples

Pour un exemple de configuration de Zabbix pour la surveillance de sites web à l'aide du modèle *Website by Browser*, consultez Surveiller des sites web avec des éléments Browser.

Script par défaut

Le script suivant :

1. Initialise une session de navigateur.
2. Accède à une URL spécifiée.
3. Collecte les entrées de performance et les statistiques de session, puis les renvoie sous forme de chaîne JSON.

Dans le champ *Script*, saisissez :

```
var browser = new Browser(Browser.chromeOptions());

try {
    browser.navigate("http://example.com");
    browser.collectPerfEntries();
}
finally {
    return JSON.stringify(browser.getResult());
}
```

Initialiser le navigateur avec des capacités personnalisées

Le script suivant :

1. Initialise une session de navigateur pour le navigateur disponible, en se basant sur le premier navigateur correspondant dans l'ordre spécifié dans le script.
2. Définit les capacités du navigateur, y compris la stratégie de chargement des pages et les options propres à chaque navigateur, comme le mode headless pour les navigateurs Chrome, Firefox et Microsoft Edge.

Dans le champ *Script*, saisissez :

```
var browser = new Browser({
    "capabilities":{
        "firstMatch":[
            {
                "browserName":"chrome",
                "pageLoadStrategy":"normal",
                "goog:chromeOptions":{
                    "args":[
                        "--headless=new"
                    ]
```

```
            }
        },
        {
            "browserName":"firefox",
            "pageLoadStrategy":"normal",
            "moz:firefoxOptions":{
                "args":[
                    "--headless"
                ]
            }
        },
        {
            "browserName":"MicrosoftEdge",
            "pageLoadStrategy":"normal",
            "ms:edgeOptions":{
                "args":[
                    "--headless=new"
                ]
            }
        },
        {
            "browserName":"safari",
            "pageLoadStrategy":"normal"
        }
    ]
  }
});
```

Initialiser le navigateur avec l'interface graphique

Par défaut, les sessions de navigateur (à l'exception de Safari) sont initialisées en mode headless, ce qui signifie que l'interface graphique utilisateur (GUI) du navigateur n'est pas affichée.

Le script suivant initialise une session de navigateur avec l'interface graphique activée.

Notez que si le WebDriver ne parvient pas à localiser le binaire du navigateur, vous pouvez spécifier le chemin manuellement.

```
var opts = Browser.chromeOptions();
opts.capabilities.alwaysMatch['goog:chromeOptions'].args = [];

// Pour initialiser une session Firefox avec interface graphique, décommentez les lignes suivantes :
// var opts = Browser.firefoxOptions();
// opts.capabilities.alwaysMatch['moz:firefoxOptions'].binary = 'usr/bin/firefox';
// opts.capabilities.alwaysMatch['moz:firefoxOptions'].args = [];

// Pour initialiser une session Microsoft Edge avec interface graphique, décommentez les lignes suivantes
// var opts = Browser.edgeOptions();
// opts.capabilities.alwaysMatch['ms:edgeOptions'].binary = 'usr/bin/microsoft-edge';
// opts.capabilities.alwaysMatch['ms:edgeOptions'].args = [];

var browser = new Browser(opts);
```

> **Note:**
> Si vos tests s'exécutent sur un serveur distant ou dans un conteneur, vous pouvez utiliser un client Virtual Network Computing (VNC) pour vous connecter au serveur VNC de la machine. Cela vous permet d'afficher et d'utiliser l'interface graphique du navigateur à distance.

Prendre des captures d'écran

Le script suivant :

1. Initialise une session de navigateur.
2. Définit la taille de la zone d'affichage du navigateur afin de déterminer la taille de la capture d'écran (spécifiée en tant que paramètres, voir ci-dessous).
3. Accède à une URL (spécifiée en tant que paramètre, voir ci-dessous).

4. Collecte les statistiques de session, capture une capture d'écran et l'ajoute aux statistiques collectées.
5. Gère les erreurs en capturant les messages d'erreur et une capture d'écran.
6. Renvoie les résultats collectés sous forme de chaîne JSON.

Le script utilise également des paramètres du formulaire de configuration de l'élément :

- webURL - http://example.com
- width - 1920
- height - 1080

Dans le champ *Script*, saisissez :

```
var browser, result;

var browser = new Browser(Browser.chromeOptions());

try {
    var params = JSON.parse(value); // Parse the JSON string containing parameters passed from Zabbix.

    browser.setScreenSize(Number(params.width), Number(params.height))

    browser.navigate(params.webURL);

    result = browser.getResult();
    result.screenshot = browser.getScreenshot();
}
catch (err) {
    if (!(err instanceof BrowserError)) {
        browser.setError(err.message);
    }
    result = browser.getResult();
    result.error.screenshot = browser.getScreenshot();
}
finally {
    return JSON.stringify(result);
}
```

Vérifier la connexion à Zabbix

Le script suivant :

1. Initialise une session de navigateur.
2. Accède à une page (spécifiée comme paramètre, voir ci-dessous).
3. Saisit le nom d'utilisateur et le mot de passe (spécifiés comme paramètres, voir ci-dessous).
4. Recherche et clique sur le bouton de connexion.
5. Recherche et clique sur le bouton de déconnexion.
6. Collecte les données de performance avant et après la connexion, ainsi qu'après la déconnexion.
7. Gère les erreurs en capturant les messages d'erreur et une capture d'écran.
8. Renvoie les résultats collectés sous forme de chaîne JSON.

Le script utilise également des paramètres du formulaire de configuration de l'élément :

- webURL - http://{HOST.CONN}/index.php
- username - {$USERNAME}
- password - {$PASSWORD}

Dans le champ *Script*, saisissez :

```
var browser, result;

browser = new Browser(Browser.chromeOptions());

try {
    var params = JSON.parse(value); // Analyse la chaîne JSON contenant les paramètres transmis par Zabbix

    browser.navigate(params.webURL);
    browser.collectPerfEntries("open page");
```

```
    var el = browser.findElement("xpath", "//input[@id='name']");
    if (el === null) {
        throw Error("cannot find name input field");
    }
    el.sendKeys(params.username);

    el = browser.findElement("xpath", "//input[@id='password']");
    if (el === null) {
        throw Error("cannot find password input field");
    }
    el.sendKeys(params.password);

    el = browser.findElement("xpath", "//button[@id='enter']");
    if (el === null) {
        throw Error("cannot find login button");
    }
    el.click();

    browser.collectPerfEntries("login");

    el = browser.findElement("link text", "Sign out");
    if (el === null) {
        throw Error("cannot find logout button");
    }
    el.click();

    browser.collectPerfEntries("logout");

    result = browser.getResult();
}
catch (err) {
    if (!(err instanceof BrowserError)) {
        browser.setError(err.message);
    }
    result = browser.getResult();
    result.error.screenshot = browser.getScreenshot();
}
finally {
    return JSON.stringify(result);
}
```

Trouver des liens

Le script suivant :

1. Initialise une session de navigateur.
2. Définit une fonction pour supprimer les éléments en double d'un tableau (voir l'étape 5).
3. Accède à une page (spécifiée comme paramètres, voir ci-dessous).
4. Trouve les liens sur la page.
5. Supprime les liens en double pour garantir leur unicité.
6. Extrait uniquement les liens qui commencent par "http".
7. Formate les liens extraits selon une structure spécifique.
8. Gère les erreurs en capturant les messages d'erreur et une capture d'écran.
9. Renvoie les résultats collectés sous forme de chaîne JSON.

Le script utilise également des paramètres du formulaire de configuration de l'élément :

- scheme - {$WEBSITE.SCHEME}
- domain - {$WEBSITE.DOMAIN}
- path - {$WEBSITE.PATH}

Dans le champ *Script*, saisissez :

```
var browser, result;

browser = new Browser(Browser.chromeOptions());
```

```
try {
    var params = JSON.parse(value);  // Parse the JSON string containing parameters passed from Zabbix.

    function uniq(a) {
        return a.sort().filter(function (item, pos, ary) {
            return !pos || item != ary[pos - 1];
        });
    }

    browser.navigate(params.scheme + '://' + params.domain + params.path);

    var el = browser.findElements("link text", "");
    var links = [];
    for (var n = 0; n < el.length; n++) {
        links.push(el[n].getAttribute('href'));
    }

    links = uniq(links);

    result = [];
    for (i = 0; i < links.length; i++) {
        if (links[i].match(/^http.*/)) {
            var row = {};
            row["{#URL}"] = links[i];
            result.push(row);
        }
    }
}
catch (err) {
    if (!(err instanceof BrowserError)) {
        browser.setError(err.message);
    }
    result = browser.getResult();
    result.error.screenshot = browser.getScreenshot();
}
finally {
    return JSON.stringify(result);
}
```

Requête de télémétrie

Ce type d'élément vous permet de récupérer ou d'agréger des données de télémétrie. Les données de télémétrie sont utilisées pour la surveillance des performances des applications (APM), ce qui vous aide à voir comment les requêtes circulent dans vos applications et services, à mesurer leurs performances et à trouver la cause des opérations lentes ou ayant échoué.

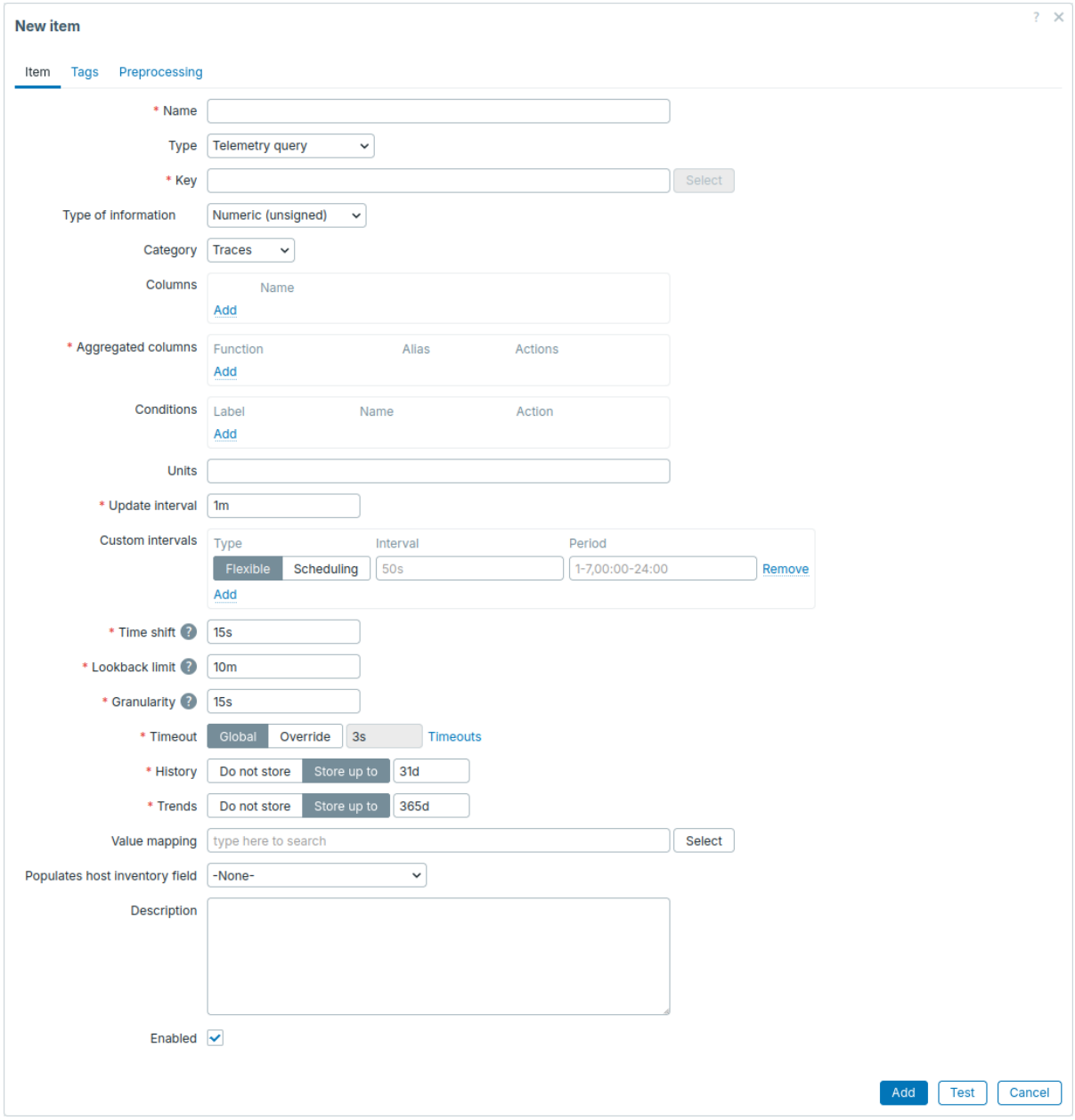
Les données de télémétrie sont disponibles sous trois formes : les traces (le chemin suivi par une requête dans vos applications et services), les métriques (des mesures telles que le temps de réponse) et les journaux (des enregistrements textuels d'événements). En utilisant l'élément de requête de télémétrie, vous pouvez, par exemple, trouver toutes les traces dans lesquelles un service a renvoyé une erreur, obtenir le temps de réponse moyen à partir de vos métriques ou rechercher un message d'erreur spécifique dans vos journaux.

Lorsque vous configurez un élément de requête de télémétrie, vous définissez un ensemble d'instructions (une requête) qui indique à Zabbix quelles données de télémétrie demander, comment les filtrer et comment les agréger. Lorsque la vérification de l'élément est exécutée, Zabbix envoie cette requête à la source de données de télémétrie et récupère le résultat en fonction de la requête.

Pour utiliser cet élément, spécifiez la source de données de télémétrie et définissez le paramètre `StartTelemetryQueryPollers` dans la configuration du serveur Zabbix ou du proxy Zabbix, selon celui qui surveille le hôte de l'élément.

Configuration

Lors de la création d'un élément, définissez son type sur *Requête de télémétrie* :



Paramètre	Description
Nom	Voir les paramètres communs à tous les éléments.
Type	Sélectionnez *Requête de télémétrie*.
Clé	Voir les paramètres communs à tous les éléments.
Type d'information	
Catégorie	Sélectionnez le type de signal à interroger : *Traces*, *Métriques* ou *Journaux*.
Points de métrique	Sélectionnez le type de point de métrique à interroger : *Somme*, *Jauge*, *Histogramme* ou *Histogramme exponentiel*. Ce paramètre est disponible si *Catégorie* est définie sur *Métriques*.
Colonnes	Ajoutez les colonnes de données de télémétrie que vous souhaitez interroger. Pour les colonnes contenant des paires clé-valeur, saisissez également le nom de la clé. Les enregistrements de ces colonnes seront inclus tels quels dans la valeur de l'élément.

Paramètre	Description
Colonnes agrégées	Ajoutez les colonnes de données de télémétrie que vous souhaitez interroger et indiquez comment calculer une valeur à partir de leurs enregistrements : **Fonction** - sélectionnez la fonction à appliquer aux enregistrements de la colonne : *min*, *max*, *avg*, *count*, *sum* ou *percentile*. **Pourcentage** - saisissez le percentile à calculer à partir des enregistrements de la colonne (si *Fonction* est définie sur *percentile*). **Colonne** - sélectionnez la colonne à partir de laquelle effectuer le calcul (si *Fonction* n'est pas définie sur *count*). **Alias** - saisissez un nom unique pour la valeur calculée, utilisé pour l'identifier dans la valeur de l'élément.
Type de calcul	Sélectionnez la manière d'évaluer les conditions définies dans le paramètre *Conditions* ci-dessous : **Et/Ou** - toutes les conditions doivent être remplies ; les conditions ayant le même *Type* et la même *Clé d'attribut* sont combinées avec *Ou*. **Et** - toutes les conditions doivent être remplies. **Ou** - au moins une condition doit être remplie. **Expression personnalisée** - les conditions doivent être remplies comme indiqué dans la formule, par exemple `A and (B or C)`. Ce paramètre est disponible si vous avez défini au moins deux conditions.
Conditions	Ajoutez des conditions pour filtrer les enregistrements des colonnes de données de télémétrie : **Type** - sélectionnez la colonne de données de télémétrie à laquelle la condition s'appliquera. **Clé d'attribut** - saisissez la clé à laquelle la condition s'appliquera (si *Type* est défini sur une colonne contenant des paires clé-valeur). **Opérateur** - sélectionnez la comparaison à utiliser (voir ci-dessous) entre la valeur de l'enregistrement et la *Valeur d'attribut*. **Valeur d'attribut** - saisissez la valeur à laquelle comparer la valeur de l'enregistrement (si *Opérateur* n'est pas défini sur *Existe*). Si *Type* est défini sur une colonne à valeur unique, la condition s'appliquera à l'enregistrement de la colonne ; si la colonne contient un tableau de valeurs, la condition correspondra à l'enregistrement si au moins une valeur du tableau satisfait la condition. Si *Type* est défini sur une colonne contenant des paires clé-valeur, la condition s'appliquera à la valeur de la clé indiquée dans *Clé d'attribut* ; si la colonne contient un tableau de paires clé-valeur, la condition correspondra à l'enregistrement si au moins une paire contenant la *Clé d'attribut* indiquée satisfait la condition. Pour les colonnes à valeur unique, vous pouvez utiliser les opérateurs suivants : **Est égal à** - sélectionne les enregistrements égaux à la *Valeur d'attribut* (sensible à la casse). **N'est pas égal à** - sélectionne les enregistrements qui ne sont pas égaux à la *Valeur d'attribut* (sensible à la casse). **Contient** - sélectionne les enregistrements qui contiennent la *Valeur d'attribut* (sensible à la casse). **Ne contient pas** - sélectionne les enregistrements qui ne contiennent pas la *Valeur d'attribut* (sensible à la casse). Pour les colonnes contenant des paires clé-valeur, vous pouvez utiliser les opérateurs suivants : **Est égal à** - sélectionne les enregistrements où la *Clé d'attribut* existe et où sa valeur est égale à la *Valeur d'attribut* (sensible à la casse). **N'est pas égal à** - sélectionne les enregistrements où la *Clé d'attribut* existe et où sa valeur n'est pas égale à la *Valeur d'attribut* (sensible à la casse). **Existe** - sélectionne les enregistrements où la *Clé d'attribut* existe (sensible à la casse).
Unités *Intervalle de mise à jour* *Intervalles personnalisés*	Voir les paramètres communs à tous les éléments.

Paramètre	Description
Décalage temporel	Saisissez le recul dans le temps (avant l'heure actuelle) auquel doit se terminer la plage temporelle interrogée par l'élément. Accepte un nombre de secondes (par exemple 1, 60, 300), une valeur avec un suffixe temporel (par exemple 30s, 1m, 2h, 1d) ou des macros utilisateur. Chaque requête demande des données pour une plage temporelle. La plage commence généralement à l'endroit où s'est terminée la plage temporelle de la requête précédente et se termine au point défini par *Décalage temporel*. Si la plage temporelle de la requête précédente s'est terminée plus tard que l'heure actuelle moins la *Granularité*, la plage temporelle de la requête actuelle commence à l'heure actuelle moins la *Granularité*, ce qui crée un chevauchement avec la plage temporelle de la requête précédente. En l'absence de requête précédente à poursuivre, la plage commence au point défini par la *Limite de recherche rétrospective*. Si le hôte de cet élément est surveillé par une autre entité (par exemple, un autre proxy d'un groupe de proxies), l'élément peut ne plus savoir quand la requête précédente s'est terminée. Cela peut entraîner des valeurs en double.
Limite de recherche rétrospective	Saisissez le recul dans le temps (avant l'heure actuelle) sur lequel l'élément doit rechercher des données lorsqu'il n'existe aucune requête précédente à poursuivre (par exemple, lors de la première exécution de l'élément), ou lorsque la requête précédente est antérieure à l'*Intervalle de mise à jour* (par exemple, parce que le serveur Zabbix était arrêté). Accepte un nombre de secondes (par exemple 1, 60, 300), une valeur avec un suffixe temporel (par exemple 30s, 1m, 2h, 1d) ou des macros utilisateur. Si la source de données de télémétrie contient une grande quantité de données (par exemple, plus d'un million d'enregistrements par heure), une *Limite de recherche rétrospective* élevée peut réduire considérablement les performances lorsque l'élément l'utilise.
Granularité	Saisissez la durée d'une portion en laquelle la plage temporelle complète de la requête est divisée. Accepte un nombre de secondes (par exemple 1, 60, 300), une valeur avec un suffixe temporel (par exemple 30s, 1m, 2h, 1d) ou des macros utilisateur.
Délai d'expiration *Historique* *Tendances* *Correspondance de valeurs* *Alimente le champ d'inventaire du hôte* *Description* *Activé*	Voir les paramètres communs à tous les éléments.

Colonnes de données de télémétrie prises en charge

Cette section répertorie les colonnes prises en charge dans les paramètres de ces éléments : *Colonnes*, *Colonnes agrégées* et *Conditions*. Il s'agit des colonnes utilisées par ClickHouse pour stocker les données de télémétrie, conformément au schéma de l'exportateur ClickHouse OpenTelemetry.

Sélectionnez la *Catégorie* :

- Traces
- Métriques : somme
- Métriques : jauge
- Métriques : histogramme
- Métriques : histogramme exponentiel
- Journaux

Traces

Colonne	*Colonnes*	*Colonnes agrégées*	*Conditions*
Durée	✓	✓	–
Events.Attributes	–	–	✓
Events.Name	–	–	✓
ParentSpanId	✓	–	✓

Colonne	*Colonnes*	*Colonnes agrégées*	*Conditions*
ResourceAttributes	✓	–	✓
ScopeName	✓	–	✓
ScopeVersion	✓	–	✓
ServiceName	✓	–	✓
SpanAttributes	✓	–	✓
SpanId	✓	–	✓
SpanKind	✓	–	✓
SpanName	✓	–	✓
StatusCode	✓	–	✓
StatusMessage	✓	–	✓
Horodatage	✓	✓	–
TraceId	✓	–	✓
TraceState	✓	–	✓

Métriques : somme

Colonne	*Colonnes*	*Colonnes agrégées*	*Conditions*
AggregationTemporality	✓	–	–
Attributes	✓	–	✓
Exemplars.FilteredAttributes	–	–	✓
Flags	✓	–	–
IsMonotonic	✓	–	–
MetricDescription	✓	–	✓
MetricName	✓	–	✓
MetricUnit	✓	–	✓
ResourceAttributes	✓	–	✓
ResourceSchemaUrl	✓	–	✓
ScopeAttributes	✓	–	✓
ScopeDroppedAttrCount	✓	✓	–
ScopeName	✓	–	✓
ScopeSchemaUrl	✓	–	✓
ScopeVersion	✓	–	✓
ServiceName	✓	–	✓
StartTimeUnix	✓	✓	–
TimeUnix	✓	✓	–
Value	✓	✓	–

Métriques : Gauge

Colonne	*Colonnes*	*Colonnes agrégées*	*Conditions*
Attributs	✓	–	✓
Exemplars.FilteredAttributes	–	–	✓
Flags	✓	–	–
MetricDescription	✓	–	✓
MetricName	✓	–	✓
MetricUnit	✓	–	✓
ResourceAttributes	✓	–	✓
ResourceSchemaUrl	✓	–	✓
ScopeAttributes	✓	–	✓
ScopeDroppedAttrCount	✓	✓	–
ScopeName	✓	–	✓
ScopeSchemaUrl	✓	–	✓
ScopeVersion	✓	–	✓
ServiceName	✓	–	✓
StartTimeUnix	✓	✓	–
TimeUnix	✓	✓	–
Value	✓	✓	–

Métriques : Histogramme

Column	*Columns*	*Aggregated columns*	*Conditions*
AggregationTemporality	✓	–	–
Attributes	✓	–	✓
Count	✓	✓	–
Exemplars.FilteredAttributes	–	–	✓
Flags	✓	–	–
Max	✓	✓	–
MetricDescription	✓	–	✓
MetricName	✓	–	✓
MetricUnit	✓	–	✓
Min	✓	✓	–
ResourceAttributes	✓	–	✓
ResourceSchemaUrl	✓	–	✓
ScopeAttributes	✓	–	✓
ScopeDroppedAttrCount	✓	✓	–
ScopeName	✓	–	✓
ScopeSchemaUrl	✓	–	✓
ScopeVersion	✓	–	–
ServiceName	✓	–	✓
StartTimeUnix	✓	✓	–
Sum	✓	✓	–
TimeUnix	✓	✓	–

Métriques : histogramme exponentiel

Colonne	*Colonnes*	*Colonnes agrégées*	*Conditions*
AggregationTemporality	✓	–	–
Attributes	✓	–	✓
Count	✓	✓	–
Exemplars.FilteredAttributes	–	–	✓
Flags	✓	–	–
Max	✓	✓	–
MetricDescription	✓	–	✓
MetricName	✓	–	✓
MetricUnit	✓	–	✓
Min	✓	✓	–
NegativeOffset	✓	✓	–
PositiveOffset	✓	✓	–
ResourceAttributes	✓	–	✓
ResourceSchemaUrl	✓	–	✓
Scale	✓	✓	–
ScopeAttributes	✓	–	✓
ScopeDroppedAttrCount	✓	✓	–
ScopeName	✓	–	✓
ScopeSchemaUrl	✓	–	✓
ScopeVersion	✓	–	✓
ServiceName	✓	–	✓
StartTimeUnix	✓	✓	–
Sum	✓	✓	–
TimeUnix	✓	✓	–
ZeroCount	✓	✓	–

Journaux

Colonne	*Colonnes*	*Colonnes agrégées*	*Conditions*
Corps	✓	–	✓
EventName	✓	–	✓
LogAttributes	✓	–	✓
ResourceAttributes	✓	–	✓

Colonne	*Colonnes*	*Colonnes agrégées*	*Conditions*
ResourceSchemaUrl	✓	–	✓
ScopeAttributes	✓	–	✓
ScopeName	✓	–	✓
ScopeSchemaUrl	✓	–	✓
ScopeVersion	✓	–	✓
SpanId	✓	–	✓
ServiceName	✓	–	✓
SeverityNumber	✓	✓	–
SeverityText	✓	–	✓
Timestamp	✓	✓	–
TraceFlags	✓	–	–
TraceId	✓	–	✓

Historique et tendances

Aperçu

L'historique et les tendances sont les deux façons de stocker les données collectées dans Zabbix.

Alors que l'historique conserve chaque valeur collectée, les tendances conservent des informations moyennées sur une base horaire et sont donc moins gourmandes en ressources.

Conservation de l'historique

Vous pouvez définir pendant combien de jours l'historique sera conservé :

- dans le formulaire des propriétés de l'élément
- lors de la mise à jour en masse des éléments
- lors de la configuration des tâches de nettoyage automatique

Toutes les données plus anciennes seront supprimées par le processus de nettoyage automatique.

Il est fortement recommandé, de manière générale, de conserver l'historique pendant le plus petit nombre de jours possible afin de ne pas surcharger la base de données avec un grand nombre de valeurs historiques.

Au lieu de conserver un long historique, vous pouvez conserver plus longtemps les données de tendances. Par exemple, vous pouvez conserver l'historique pendant 14 jours et les tendances pendant 5 ans.

Vous pouvez obtenir une bonne estimation de l'espace requis par les données d'historique par rapport aux données de tendances en consultant la page de dimensionnement de la base de données.

Même en conservant un historique plus court, vous pourrez toujours consulter les anciennes données dans les graphiques, car les graphiques utiliseront les valeurs de tendance pour afficher les données plus anciennes.

Attention:
Si l'historique est défini sur '0', l'élément mettra à jour uniquement les éléments dépendants et l'inventaire. Aucune fonction de déclencheur ne sera évaluée, car l'évaluation des déclencheurs repose uniquement sur les données d'historique.

Note:
Comme autre moyen de préserver l'historique, vous pouvez envisager d'utiliser la fonctionnalité d'export de l'historique des modules chargeables.

Conservation des tendances

Les tendances constituent un mécanisme intégré de réduction des données historiques qui stocke, pour chaque heure, les valeurs minimale, maximale, moyenne ainsi que le nombre total de valeurs pour les types de données numériques.

Vous pouvez définir pendant combien de jours les tendances seront conservées :

- dans le formulaire des propriétés de l'élément
- lors de la mise à jour en masse des éléments
- lors de la configuration des tâches du Housekeeper

Les tendances peuvent généralement être conservées beaucoup plus longtemps que l'historique. Toute donnée plus ancienne sera supprimée par le housekeeper.

Le serveur Zabbix accumule les données de tendance à l'exécution dans le cache des tendances, au fur et à mesure de l'arrivée des données. Le serveur écrit dans la base de données les tendances de **l'heure précédente** pour chaque élément (où le frontend peut les trouver) dans les situations suivantes :

- le serveur reçoit la première valeur de l'heure en cours pour l'élément
- il reste 5 minutes ou moins dans l'heure en cours et aucune valeur de l'heure en cours n'a encore été reçue pour l'élément
- le serveur s'arrête

Pour voir les tendances sur un graphique, vous devez attendre au minimum le début de l'heure suivante (si l'élément est mis à jour fréquemment) et au maximum la fin de l'heure suivante (si l'élément est mis à jour rarement), soit 2 heures au maximum.

Lorsque le serveur vide le cache des tendances et qu'il existe déjà des tendances dans la base de données pour cette heure (par exemple, si le serveur a été redémarré au milieu de l'heure), le serveur doit utiliser des instructions de mise à jour au lieu de simples insertions. Par conséquent, sur une installation plus importante, si un redémarrage est nécessaire, il est préférable d'arrêter le serveur à la fin d'une heure et de le démarrer au début de l'heure suivante afin d'éviter le chevauchement des données de tendance.

Les tables d'historique ne participent en aucune façon à la génération des tendances.

Attention:
Si les tendances sont définies sur « 0 », le serveur Zabbix ne calcule ni ne stocke de tendances.

Note:
Les tendances sont calculées et stockées avec le même type de données que les valeurs d'origine. Par conséquent, les calculs de valeur moyenne pour les valeurs de type de données non signé sont arrondis, et plus l'intervalle de valeurs est faible, moins le résultat sera précis. Par exemple, si un élément a les valeurs 0 et 1, la valeur moyenne sera 0, et non 0,5. De plus, le redémarrage du serveur peut entraîner une perte de précision dans les calculs de valeur moyenne du type de données non signé pour l'heure en cours.

Paramètres utilisateur

Overview

A user parameter lets you define a custom item key and the command used to retrieve its value when no built-in Zabbix agent item provides the data you need. A user parameter is a user-defined command referenced by a user-defined item key and executed by the Zabbix agent when the corresponding item is checked.

Note:
On UNIX systems, the agent executes user parameter commands using `/bin/sh`, under the operating system account running the agent (usually `zabbix`). That account must have sufficient permissions to execute the command and access any files, sockets, or credentials it uses.

There are two types of user parameters:

- **Simple user parameters** do not accept arguments; they define a fixed item key and a command to execute.
- **Flexible user parameters** accept arguments (specified as parameters in the item key) and pass them to the command, allowing a single definition to serve multiple items.

User parameter configuration

A user parameter is defined using the `UserParameter` configuration parameter.

A user parameter has the following syntax:

`UserParameter=<key>,<shell command>`

where:

- `<key>` - a unique item key that identifies the user parameter.
 When configuring an item, specify this key to reference the command defined by the user parameter.
 For a simple user parameter, the key is fixed and does not accept arguments. For a flexible user parameter, append `[*]` to the key (for example, `mysql.status[*]`) to indicate that it accepts arguments. Specify these arguments as parameters in the item key when configuring the item. The command can refer to the parameters using positional references $1–$9.
- `<shell command>` - a shell command or a path to a custom script that the agent executes to obtain the item value.
 For the flexible user parameter, the command can use positional references $1–$9 to refer to the corresponding parameters in the item key (for example, `UserParameter=mysql.status[*],mysqladmin status --user=$1 --password=$2`). Zabbix parses the parameters enclosed in `[]` in the item key (i.e., `mysql.status[zabbix,{$MYSQL.PASSWORD}]`) and

substitutes the positional references in the command accordingly.
 $0 is substituted with the original command before positional references are expanded. Positional references are interpreted regardless of whether they are enclosed in double (") or single (') quotes. To use positional references without substitution, specify a double dollar sign. For example, `awk '{print $$2}'` passes $2 to the command unchanged.
 Note that positional references prefixed with $ are searched for and replaced by the Zabbix agent only in flexible user parameters. For simple user parameters, positional references are not processed, so escaping $ is unnecessary.

Attention:
Certain characters cannot be passed as arguments by default. For the complete list of characters, see UnsafeUserParameters.

`UserParameter` must be configured on each Zabbix agent that uses it: - Directly in the Zabbix agent configuration file (`zabbix_agentd.conf` or `zabbix_agent2.conf`). - In a separate `*.conf` file, provided that the file is included by the agent configuration.

More than one `UserParameter` can be defined per agent, as multiple files can be included with different sets of parameters. Each user parameter must have a unique item key within the monitored host.

The `UserParameterDir` configuration parameter lets you specify the working directory for executing `UserParameter` commands. If configured, the agent changes its working directory to the specified directory before executing a command. Commands can be specified using a relative path with the `./` prefix, for example, `UserParameter=mykey,./myscript.sh`.

To configure a user parameter:

1. Write a custom script or command to retrieve the required data.
2. Add the command and its unique item key to the `UserParameter` configuration parameter in the Zabbix agent configuration file (`zabbix_agentd.conf`).
3. Reload user parameters from the configuration file by running the runtime control command:

   zabbix_agentd -R userparameter_reload

   Alternatively, restart the agent to apply the configuration changes.
4. Create a new item on the monitored host with the item key defined in `UserParameter`.
 Set the item's type to *Zabbix Agent* or *Zabbix Agent (active)*.
 Make sure that the item's *Type of information* is configured correctly; otherwise, Zabbix will reject the returned value.

See Extending Zabbix agents for an example of configuring and using user parameters.

User parameters are subject to the Zabbix agent check timeout. If the timeout is reached, the process executing the user parameter is terminated. See Command execution for more information.

Note that long-running scripts may consume system resources and delay other checks.

Résultat de la commande

La valeur de retour de la commande correspond à la sortie standard ainsi qu'à la sortie d'erreur standard produites par la commande.

Attention:
Un élément qui renvoie du texte (informations de type caractère, journal ou texte) ne deviendra pas non pris en charge en cas de sortie d'erreur standard.

La valeur de retour est limitée à 16 Mo (y compris les espaces de fin qui sont tronqués) ; les limites de la base de données s'appliquent également.

Les paramètres utilisateur qui renvoient du texte (informations de type caractère, journal ou texte) peuvent également renvoyer des espaces. En cas de résultat non valide, l'élément deviendra non pris en charge.

Exemples de paramètres utilisateur simples

Une commande simple :

UserParameter=ping,echo 1

L'agent renverra toujours '1' pour une élément avec la clé 'ping'.

Un exemple plus complexe :

UserParameter=mysql.ping,mysqladmin -uroot ping | grep -c alive

L'agent renverra '1', si le serveur MySQL est actif, '0' - sinon.

Exemple 1

Quelque chose de très simple :

```
UserParameter=ping[*],echo $1
```

Nous pouvons définir un nombre illimité d'éléments pour la surveillance de tous les formats ping[quelque chose].

- ping[0] - retournera toujours '0'
- ping[aaa] - retournera toujours 'aaa'

Exemple 2

Ajoutons un peu de sens !

```
UserParameter=mysql.ping[*],mysqladmin -u$1 -p$2 ping | grep -c alive
```

Ce paramètre peut être utilisé pour superviser la disponibilité de la base de données MySQL. Nous pouvons passer le nom d'utilisateur et le mot de passe :

```
mysql.ping[zabbix,our_password]
```

Exemple 3

Combien de lignes correspondent à une expression régulière dans un fichier ?

```
UserParameter=wc[*],grep -c "$2" $1
```

Ce paramètre peut être utilisé pour calculer le nombre de ligne dans un fichier.

```
wc[/etc/passwd,root]
wc[/etc/services,zabbix]
```

Extension des agents Zabbix

Ce tutoriel fournit des instructions étape par étape pour étendre les fonctionnalités de l'agent Zabbix à l'aide d'un paramètre utilisateur.

Étape 1

Écrivez un script ou une commande en ligne de commande pour récupérer le paramètre requis.

Par exemple, nous pouvons écrire la commande suivante afin d'obtenir le nombre total de requêtes exécutées par un serveur MySQL :

```
mysqladmin -uroot status | cut -f4 -d":" | cut -f1 -d"S"
```

Lorsqu'elle est exécutée, la commande renvoie le nombre total de requêtes SQL.

Étape 2

Ajoutez la commande au fichier de configuration de l'agent Zabbix (`zabbix_agentd.conf`) :

```
UserParameter=mysql.questions,mysqladmin -uroot status | cut -f4 -d":" | cut -f1 -d"S"
```

mysql.questions est un identifiant unique. Il peut s'agir de n'importe quel identifiant de clé valide, par exemple *queries*.

Testez ce paramètre en utilisant l'agent Zabbix avec l'option `-t` (si l'agent est exécuté avec les privilèges root, notez toutefois qu'il peut disposer d'autorisations différentes lorsqu'il est lancé en tant que démon) :

```
zabbix_agentd -t mysql.questions
```

Étape 3

Rechargez les paramètres utilisateur depuis le fichier de configuration en exécutant :

```
zabbix_agentd -R userparameter_reload
```

Vous pouvez également redémarrer l'agent au lieu d'utiliser la commande de contrôle à chaud.

Testez le paramètre à l'aide de l'utilitaire `zabbix_get`.

Étape 4

Ajoutez un nouvel élément avec `Key=mysql.questions` à l'hôte surveillé. Le type de l'élément doit être soit *Zabbix Agent* soit *Zabbix Agent (active)*.

Veillez à ce que le type des valeurs renvoyées soit correctement défini sur le serveur Zabbix. Sinon, Zabbix ne les acceptera pas.

Compteurs de performance Windows

Vue d'ensemble

Vous pouvez surveiller efficacement les compteurs de performances Windows à l'aide de la clé `perf_counter[]`.

Par exemple :

```
perf_counter["\Processor(0)\Interrupts/sec"]
```

ou

```
perf_counter["\Processor(0)\Interrupts/sec", 10]
```

Pour plus d'informations sur l'utilisation de cette clé ou de son équivalent en anglais uniquement `perf_counter_en`, consultez Clés d'éléments spécifiques à Windows.

Pour obtenir la liste complète des compteurs de performances disponibles pour la surveillance, vous pouvez exécuter :

```
typeperf -qx
```

Vous pouvez également utiliser la découverte de bas niveau pour découvrir plusieurs instances d'objets des compteurs de performances Windows et automatiser la création d'éléments `perf_counter` pour plusieurs objets d'instance.

Représentation numérique

Windows conserve des représentations numériques (index) pour les noms des objets et des compteurs de performance. Zabbix prend en charge ces représentations numériques en tant que paramètres des clés d'élément `perf_counter`, `perf_counter_en` et des paramètres de configuration `PerfCounter`, `PerfCounterEn`.

Cependant, il n'est pas recommandé de les utiliser, sauf si vous pouvez garantir que vos index numériques correspondent aux chaînes correctes sur des hôtes spécifiques. Si vous devez créer des éléments portables fonctionnant sur différents hôtes équipés de différentes versions localisées de Windows, vous pouvez utiliser la clé `perf_counter_en` ou le paramètre de configuration `PerfCounterEn`, qui vous permettent d'utiliser les noms anglais quelle que soit la langue du système.

Pour connaître les équivalents numériques, exécutez `regedit`, puis recherchez *Counter* dans `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Perflib\009`.

L'entrée du registre contient des informations telles que celles-ci :

```
1
1847
2
System
4
Memory
6
% Processor Time
10
File Read Operations/sec
12
File Write Operations/sec
14
File Control Operations/sec
16
File Read Bytes/sec
18
File Write Bytes/sec
....
```

Vous trouverez ici les nombres correspondants à chaque partie textuelle du compteur de performance, comme dans `\System\% Processor Time` :

```
System → 2
% Processor Time → 6
```

Vous pouvez ensuite utiliser ces nombres pour représenter le chemin sous forme numérique :

```
\2\6
```

Paramètres des compteurs de performance

Vous pouvez déployer certains paramètres PerfCounter pour surveiller les compteurs de performance Windows.

Par exemple, vous pouvez les ajouter au fichier de configuration de l'agent Zabbix :

```
PerfCounter=UserPerfCounter1,"\Memory\Page Reads/sec",30
ou
PerfCounter=UserPerfCounter2,"\4\24",30
```

Une fois ces paramètres configurés, il vous suffit d'utiliser `UserPerfCounter1` ou `UserPerfCounter2` comme clés pour créer les éléments correspondants.

N'oubliez pas de redémarrer l'agent Zabbix après avoir modifié le fichier de configuration.

Mise à jour en masse

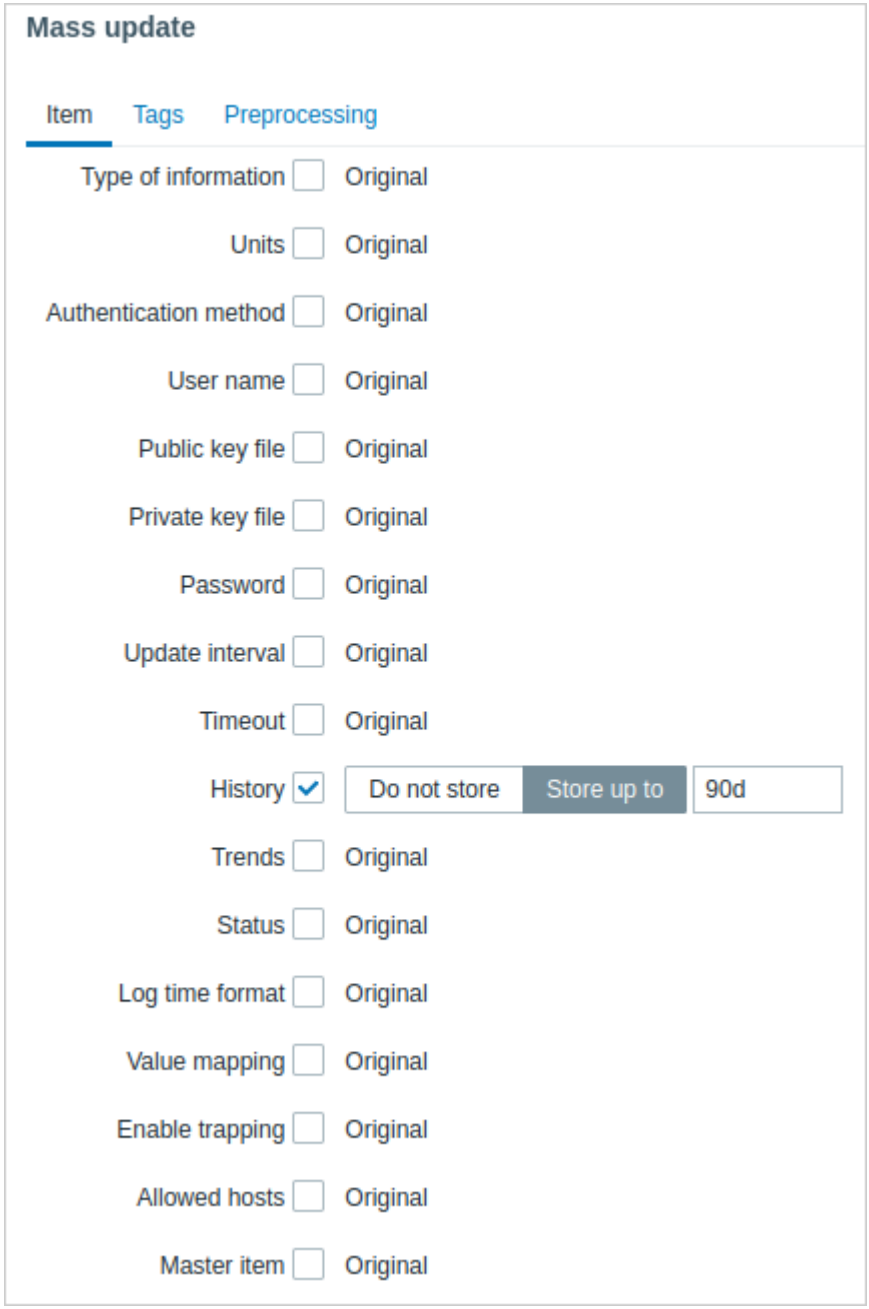
Aperçu

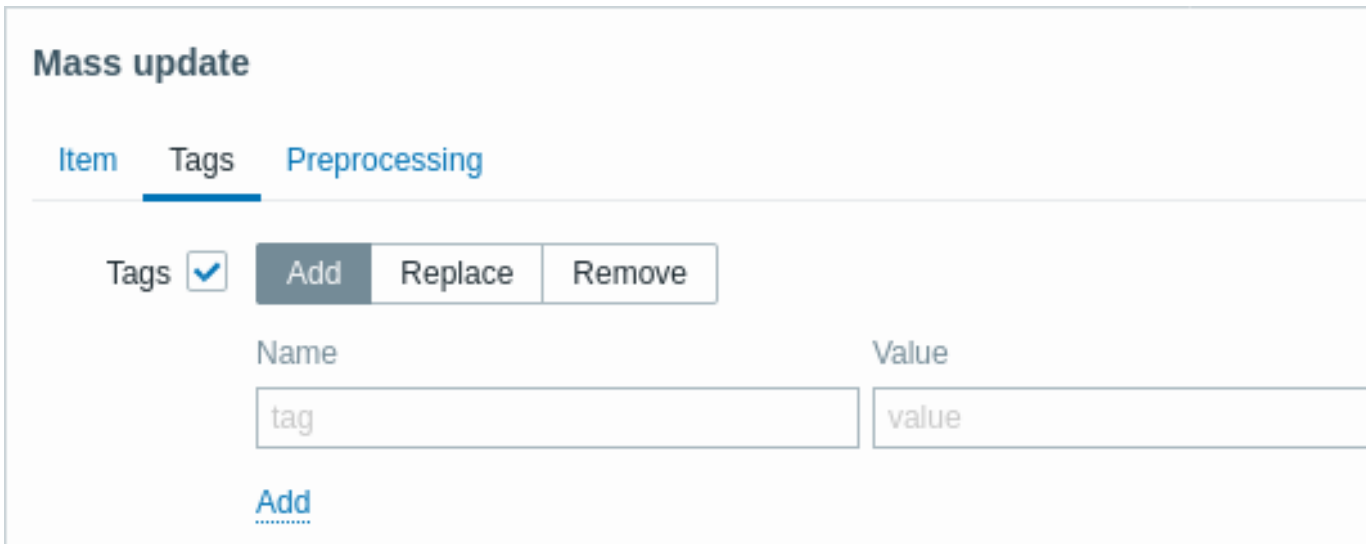
Parfois, vous voudrez peut-être modifier un attribut pour un certain nombre d'éléments à la fois. Au lieu d'ouvrir chaque élément individuel pour le modifier, vous pouvez utiliser la fonction de mise à jour en masse pour cela.

Utilisation de la mise à jour en masse

Pour mettre à jour plusieurs éléments en masse, procédez comme suit :

- Cochez les cases des éléments à mettre à jour dans la liste
- Cliquez sur *Mise à jour en masse* sous la liste
- Accédez à l'onglet contenant les attributs requis (*Élément*, *Étiquettes* ou *Prétraitement*)
- Cochez les cases des attributs à mettre à jour
- Saisissez les nouvelles valeurs des attributs



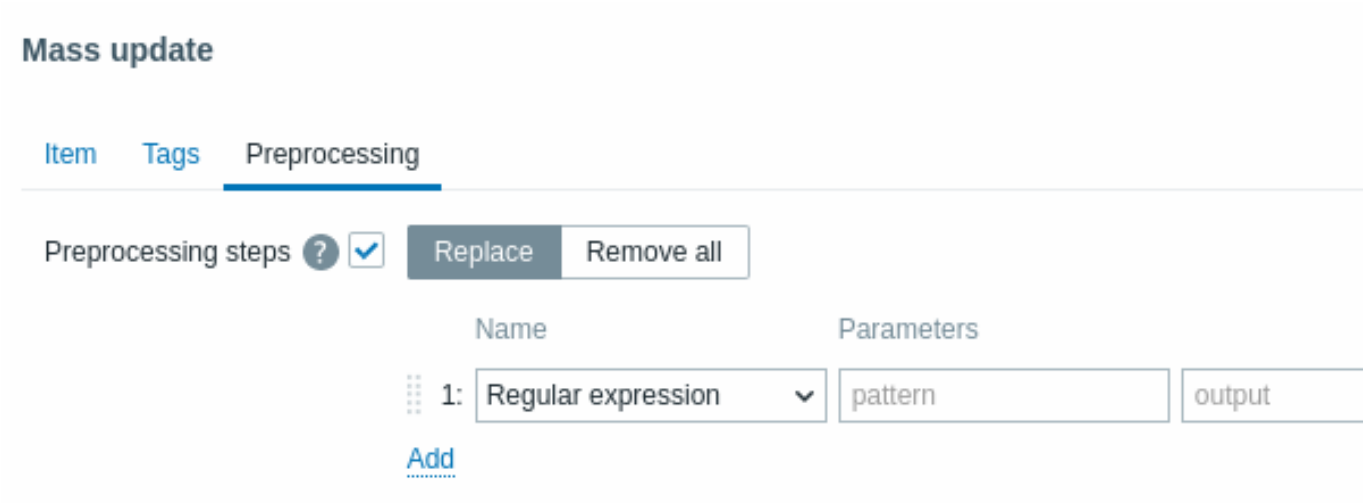


L'option *Étiquettes* vous permet de :

- *Ajouter* - ajouter les étiquettes spécifiées aux éléments (les étiquettes portant le même nom, mais ayant des valeurs différentes, ne sont pas considérées comme des « doublons » et peuvent être ajoutées au même élément).
- *Remplacer* - supprimer les étiquettes spécifiées et ajouter des étiquettes avec de nouvelles valeurs

- *Supprimer* - supprimer les étiquettes spécifiées des éléments

Les macros utilisateur, les macros {INVENTORY.*}, {HOST.HOST}, {HOST.NAME}, {HOST.CONN}, {HOST.DNS}, {HOST.IP}, {HOST.PORT} et {HOST.ID} sont prises en charge dans les étiquettes.



Cochez la case située à côté de *Étapes de prétraitement* pour activer la mise à jour en masse des étapes de prétraitement.

La mise à jour en masse du *Prétraitement* vous permet de :

- *Remplacer* - remplacer les étapes de prétraitement existantes sur les éléments par les étapes de prétraitement spécifiées ci-dessous
- *Tout supprimer* - supprimer toutes les étapes de prétraitement des éléments

Lorsque vous avez terminé, cliquez sur *Mettre à jour*.

Mappage de valeurs

Présentation

Le mappage de valeurs permet de configurer une représentation plus conviviale des valeurs reçues à l'aide de mappages entre des valeurs numériques/de chaîne et des représentations de chaîne.

Par exemple, lorsque la valeur d'un élément est « 0 » ou « 1 », des mappages de valeurs peuvent être utilisés pour représenter ces valeurs de manière plus conviviale :

- 0 → Non disponible
- 1 → Disponible

Les mappages de valeurs pour les types de sauvegarde de données peuvent être configurés comme suit :

- F → Complet
- D → Différentiel
- I → Incrémentiel

Les plages de valeurs pour la tension peuvent être configurées comme suit :

- <=209 → Faible
- 210-230 → OK
- >=231 → Élevé

Le mappage de valeurs est utilisé dans le frontend Zabbix et les notifications envoyées par les types de média.

Attention:
La substitution de la valeur reçue par la représentation configurée est effectuée à la fois dans le frontend Zabbix et sur le serveur ; cependant, le serveur gère la substitution uniquement dans les cas suivants :
- lors du remplissage de l'inventaire de l'hôte ;
- lors du développement de macros prises en charge {ITEM.VALUE}, {ITEM.LASTVALUE}, {EVENT.OPDATA} et {EVENT.CAUSE.OPDATA}.

Les mappages de valeurs sont configurés sur des modèles ou des hôtes. Une fois configurés, ils sont disponibles pour tous les éléments du modèle ou de l'hôte respectif. Lors de la configuration des éléments, spécifiez le nom d'un mappage de valeurs précédemment configuré dans le paramètre *Mappage de valeurs*.

Note:
Il n'y a pas d'héritage de mappage de valeurs - les hôtes et les modèles n'héritent pas des mappages de valeurs des modèles liés. Les éléments de modèle sur un hôte continueront d'utiliser les mappages de valeurs configurés sur le modèle.

Note:
Les mappages de valeurs peuvent être utilisés avec des éléments ayant des types d'informations *Numérique (non signé)*, *Numérique (float)* et *Caractère*.

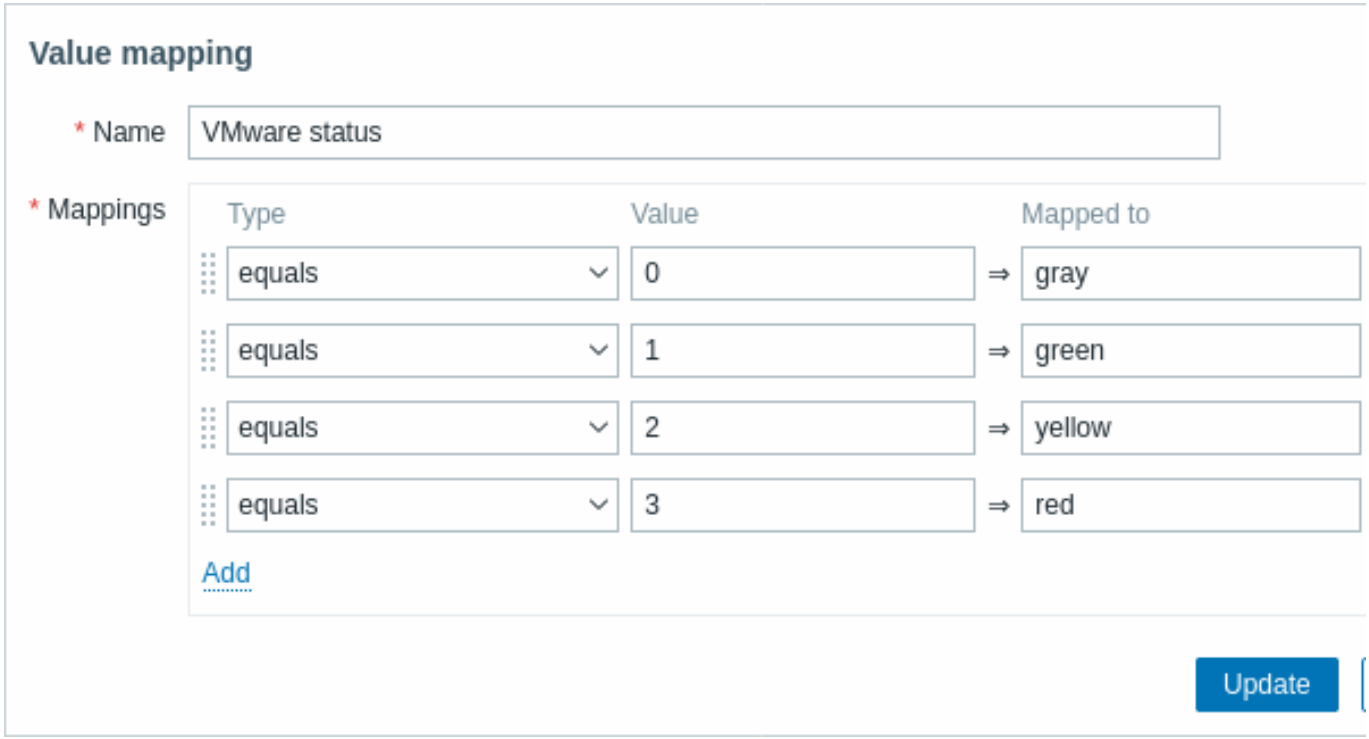
Les mappages de valeurs sont exportés/importés avec le modèle ou l'hôte respectif. Ils peuvent également être mis à jour en masse à l'aide des formulaires de mise à jour en masse hôte et modèle.

Configuration

Pour configurer une correspondance de valeurs, procédez comme suit :

1. Ouvrez le formulaire de configuration de l'hôte ou du modèle.

2. Dans l'onglet *Correspondance des valeurs*, cliquez sur *Ajouter* pour ajouter une nouvelle correspondance de valeurs, ou cliquez sur le nom d'une correspondance existante pour la modifier.



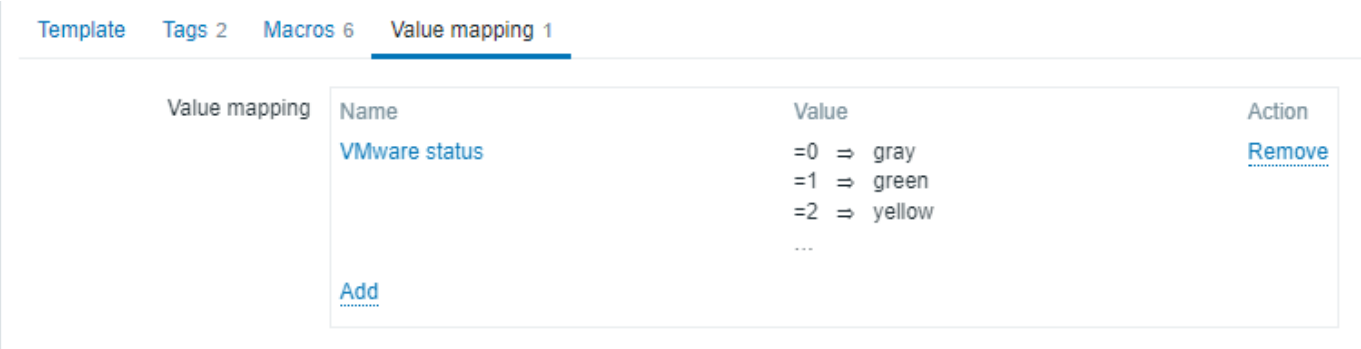
Paramètres d'une correspondance de valeurs :

Paramètre	Description
Nom	Nom unique de l'ensemble des correspondances de valeurs.
Correspondances	Règles individuelles permettant de faire correspondre des valeurs numériques ou textuelles à des représentations textuelles. La correspondance est appliquée dans l'ordre des règles, qui peuvent être réorganisées par glisser-déposer.

Paramètre	Description
Type	Type de correspondance : **equals** - les valeurs égales seront mises en correspondance ; **is greater than or equals** - les valeurs égales ou supérieures seront mises en correspondance ; **is less than or equals** - les valeurs égales ou inférieures seront mises en correspondance ; **in range** - les valeurs comprises dans la plage seront mises en correspondance ; la plage est exprimée sous la forme <number1>-<number2> ou <number> ; plusieurs plages sont prises en charge (par exemple, 1-10,101-110,201) ; **regexp** - les valeurs correspondant à l'expression régulière seront mises en correspondance (les expressions régulières globales ne sont pas prises en charge) ; **default** - toutes les valeurs restantes seront mises en correspondance, à l'exception de celles ayant des correspondances spécifiques. Pour les correspondances de plages, seuls les types de valeurs numériques (*is greater than or equals*, *is less than or equals*, *in range*) sont pris en charge.
Valeur	Valeur reçue (peut contenir une plage ou une expression régulière, selon le type de correspondance).
Correspond à	Représentation textuelle (jusqu'à 64 caractères) de la valeur reçue.

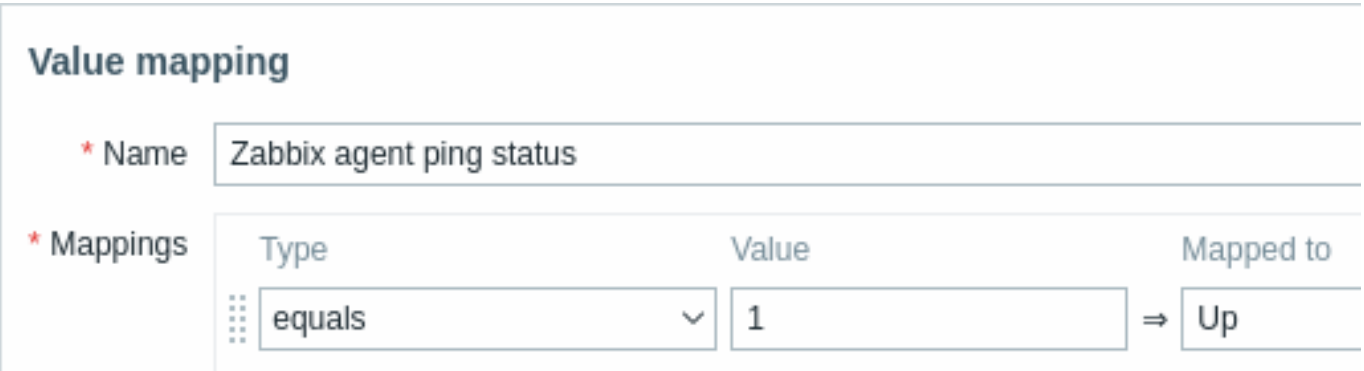
Tous les champs de saisie obligatoires sont marqués d'un astérisque rouge.

Lors de l'affichage de la correspondance de valeurs dans la liste, seules les trois premières correspondances sont visibles ; trois points indiquent que d'autres correspondances existent.



Exemple de correspondance de valeurs

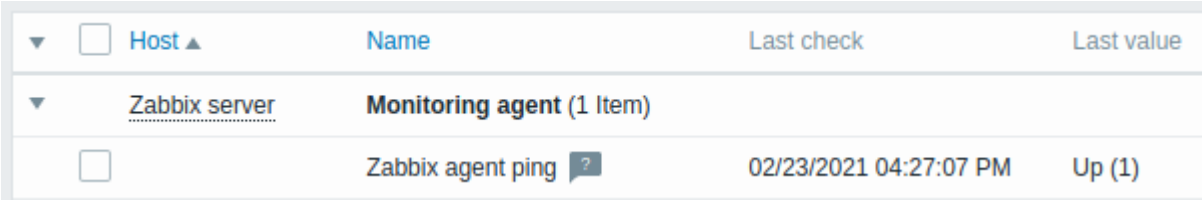
L'un des éléments prédéfinis de l'agent, *Zabbix agent ping*, utilise une correspondance de valeurs au niveau du modèle, « Zabbix agent ping status », pour afficher ses valeurs.



Dans le formulaire de configuration de l'élément, vous trouverez une référence à cette correspondance de valeurs dans le champ *Correspondance de valeurs* :

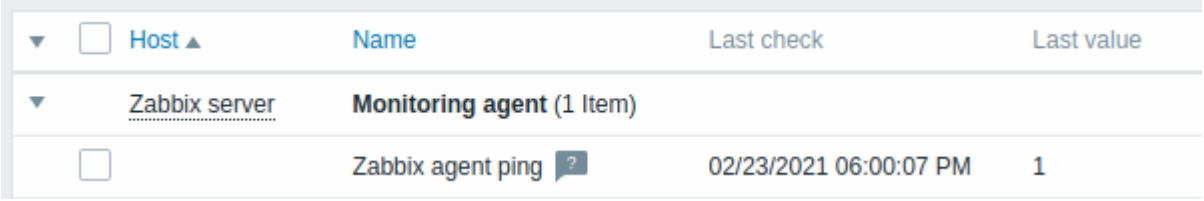


Cette correspondance est utilisée dans la section *Surveillance* > *Dernières données* pour afficher « Up » (avec la valeur brute entre parenthèses).



Note:
Dans la section *Dernières données*, les valeurs affichées sont limitées à 20 caractères. Si une correspondance de valeurs est utilisée, cette limitation ne s'applique pas à la valeur correspondante, mais uniquement à la valeur brute (affichée entre parenthèses).

Sans correspondance de valeurs prédéfinie, vous ne verriez que « 1 », ce qui pourrait être difficile à comprendre.



File d'attente

Aperçu

La file d'attente affiche les éléments en attente d'un rafraîchissement. La file d'attente n'est qu'une représentation **logique** des données. Il n'existe aucune file IPC ni aucun autre mécanisme de file d'attente dans Zabbix.

Les éléments surveillés par des proxies sont également inclus dans la file d'attente ; ils seront comptabilisés comme étant en file d'attente pendant la période de mise à jour des données d'historique du proxy.

Seuls les éléments ayant des heures de rafraîchissement planifiées sont affichés dans la file d'attente. Cela signifie que les types d'éléments suivants sont exclus de la file d'attente :

- les éléments log, logrt et event log de l'agent Zabbix actif
- les éléments de trap SNMP
- les éléments trapper
- les éléments de supervision web
- les éléments dépendants

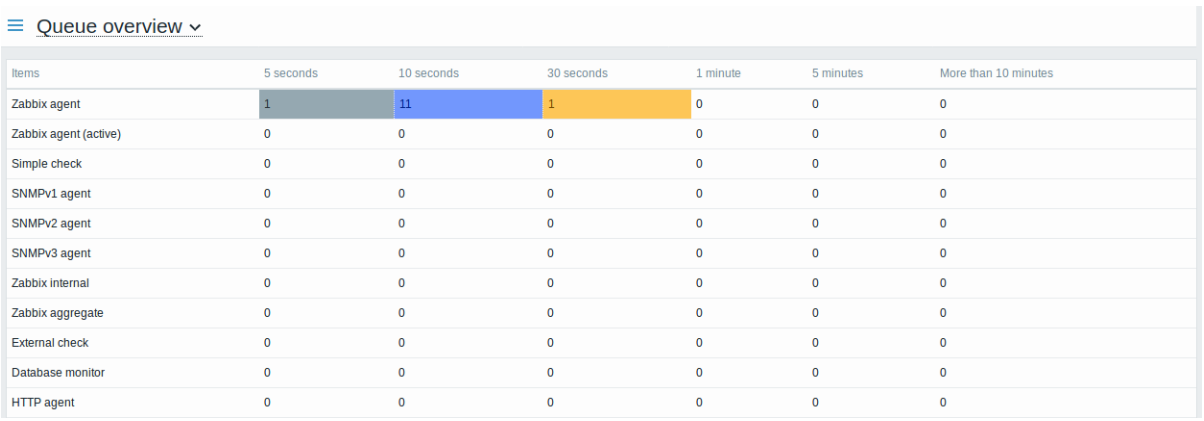
Les statistiques affichées par la file d'attente constituent un bon indicateur des performances du serveur Zabbix.

La file d'attente est récupérée directement depuis le serveur Zabbix à l'aide du protocole JSON. Les informations ne sont disponibles que si le serveur Zabbix est en cours d'exécution.

Attention:
Les éléments ne sont pas comptabilisés dans la file d'attente si l'interface de l'élément devient indisponible en raison de problèmes de connexion ou d'un agent ne fonctionnant pas correctement.

Lecture de la file d'attente

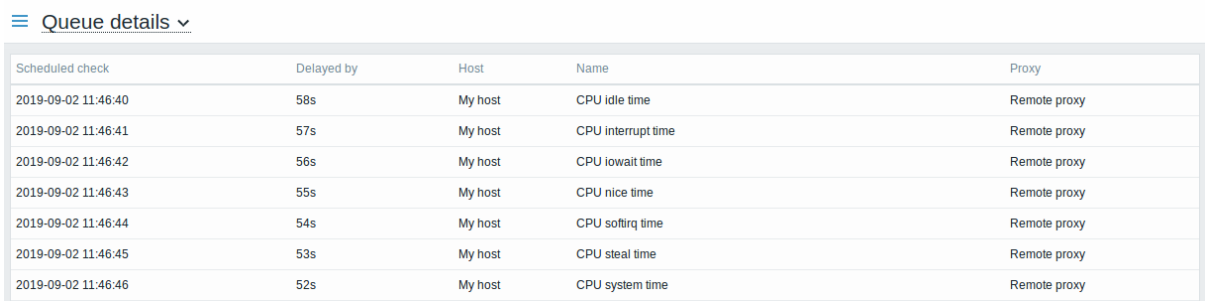
Pour lire la file d'attente, accédez à *Administration* > *File d'attente* > *Vue d'ensemble de la file d'attente*.



L'image présentée ici est généralement « correcte » ; nous pouvons donc supposer que le serveur fonctionne correctement.

La file d'attente indique que certains éléments attendent depuis 30 secondes au maximum. Il serait utile de savoir de quels éléments il s'agit.

Pour cela, sélectionnez *Détails de la file d'attente* dans la liste déroulante du titre. Vous pouvez maintenant voir la liste de ces éléments retardés.

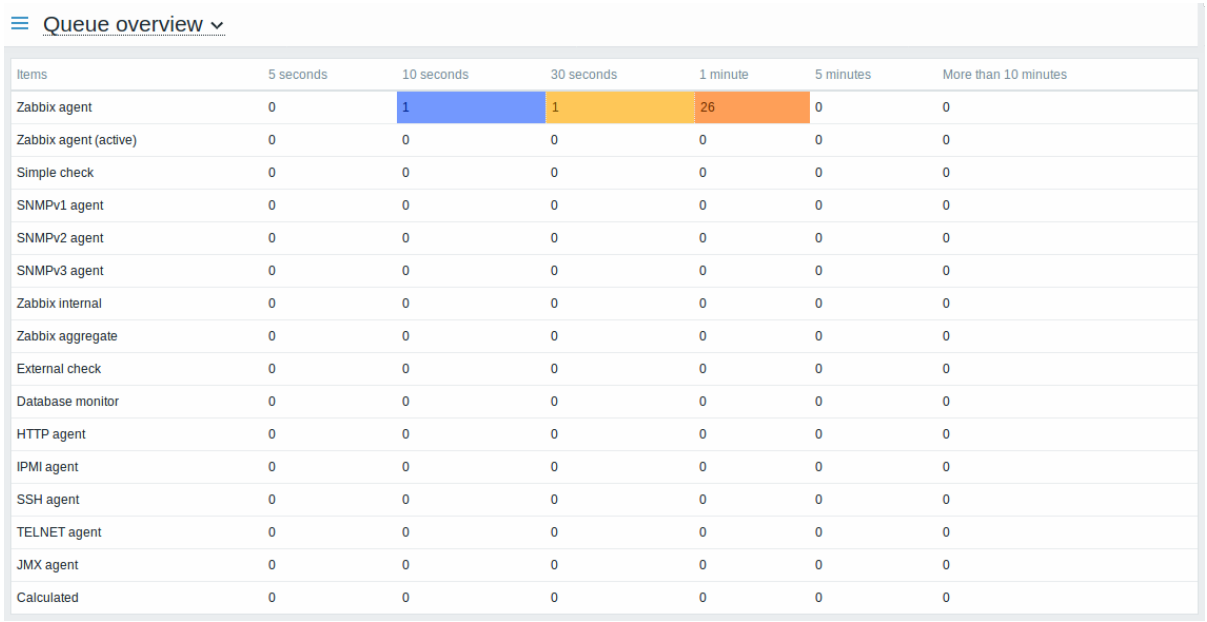

Queue details

Scheduled check	Delayed by	Host	Name	Proxy
2019-09-02 11:46:40	58s	My host	CPU idle time	Remote proxy
2019-09-02 11:46:41	57s	My host	CPU interrupt time	Remote proxy
2019-09-02 11:46:42	56s	My host	CPU iowait time	Remote proxy
2019-09-02 11:46:43	55s	My host	CPU nice time	Remote proxy
2019-09-02 11:46:44	54s	My host	CPU softirq time	Remote proxy
2019-09-02 11:46:45	53s	My host	CPU steal time	Remote proxy
2019-09-02 11:46:46	52s	My host	CPU system time	Remote proxy

Grâce à ces informations détaillées, il peut être possible de déterminer pourquoi ces éléments sont retardés.

Un ou deux éléments retardés ne sont peut-être pas préoccupants. Ils peuvent être mis à jour dans la seconde qui suit. Toutefois, si vous constatez qu'un grand nombre d'éléments sont retardés pendant trop longtemps, il peut s'agir d'un problème plus sérieux.

Voir également : Alignement des fuseaux horaires lors de l'utilisation des intervalles de planification.


Queue overview

Items	5 seconds	10 seconds	30 seconds	1 minute	5 minutes	More than 10 minutes
Zabbix agent	0	1	1	26	0	0
Zabbix agent (active)	0	0	0	0	0	0
Simple check	0	0	0	0	0	0
SNMPv1 agent	0	0	0	0	0	0
SNMPv2 agent	0	0	0	0	0	0
SNMPv3 agent	0	0	0	0	0	0
Zabbix internal	0	0	0	0	0	0
Zabbix aggregate	0	0	0	0	0	0
External check	0	0	0	0	0	0
Database monitor	0	0	0	0	0	0
HTTP agent	0	0	0	0	0	0
IPMI agent	0	0	0	0	0	0
SSH agent	0	0	0	0	0	0
TELNET agent	0	0	0	0	0	0
JMX agent	0	0	0	0	0	0
Calculated	0	0	0	0	0	0

Élément de file d'attente

Un élément interne spécial `zabbix[queue,<from>,<to>]` peut être utilisé pour surveiller l'état de la file d'attente dans Zabbix. Il renvoie le nombre d'éléments retardés de la durée définie. Pour plus d'informations, consultez la page Éléments internes.

Cache de valeurs

Vue d'ensemble

Pour accélérer considérablement le calcul des expressions de déclencheur, des éléments calculés et de certaines macros, une option de cache des valeurs est prise en charge par le serveur Zabbix.

Ce cache en mémoire peut être utilisé pour accéder aux données historiques, au lieu d'effectuer des appels SQL directs à la base de données. Si les valeurs historiques ne sont pas présentes dans le cache, les valeurs manquantes sont demandées à la base de données et le cache est mis à jour en conséquence.

Les valeurs d'élément restent dans le cache des valeurs jusqu'à ce que :

- l'élément soit supprimé (les valeurs mises en cache sont supprimées après la prochaine synchronisation de la configuration) ;
- la valeur de l'élément soit en dehors de la plage de temps ou du nombre de valeurs spécifiés dans l'expression du déclencheur/de l'élément calculé (la valeur mise en cache est supprimée lorsqu'une nouvelle valeur est reçue) ;

- la plage de temps ou le nombre de valeurs spécifiés dans l'expression du déclencheur/de l'élément calculé soient modifiés de sorte que moins de données soient nécessaires pour le calcul (les valeurs mises en cache inutiles sont supprimées après 24 heures).

Note:
L'état du cache des valeurs peut être observé à l'aide de l'option runtime control du serveur `diaginfo` (ou `diaginfo=valuecache`) et en examinant la section contenant les informations de diagnostic du cache des valeurs. Cela peut être utile pour déterminer les déclencheurs ou les éléments calculés mal configurés.

Pour activer la fonctionnalité de cache des valeurs, le paramètre facultatif `ValueCacheSize` peut être spécifié dans la configuration du serveur Zabbix.

Deux éléments internes sont pris en charge pour surveiller le cache des valeurs : `zabbix[vcache,buffer,<mode>]` et `zabbix[vcache,cache,<parameter>]`. Pour plus de détails, consultez les éléments internes.

Exécuter maintenant

Aperçu

L'option *Exécuter maintenant* permet d'exécuter immédiatement les contrôles passifs.

La collecte des valeurs d'élément dans Zabbix suit les intervalles de mise à jour configurés. Certains éléments (par exemple, les règles de découverte de bas niveau) ont des intervalles longs, et il peut être nécessaire de récupérer immédiatement une nouvelle valeur, par exemple pour détecter des changements dans les ressources découvrables.

L'option *Exécuter maintenant* est prise en charge pour les types d'élément suivants :

- Agent Zabbix (passif)
- Contrôle simple
- Agent SNMP (v1/v2/v3)
- Interne Zabbix
- Contrôle externe
- Moniteur de base de données
- Agent HTTP
- Agent IPMI
- Agent SSH
- Agent TELNET
- Agent JMX
- Calculé
- Élément dépendant
- Script
- Navigateur

Si l'élément est de type « Élément dépendant », son élément maître doit également être de l'un des types ci-dessus.

Configuration

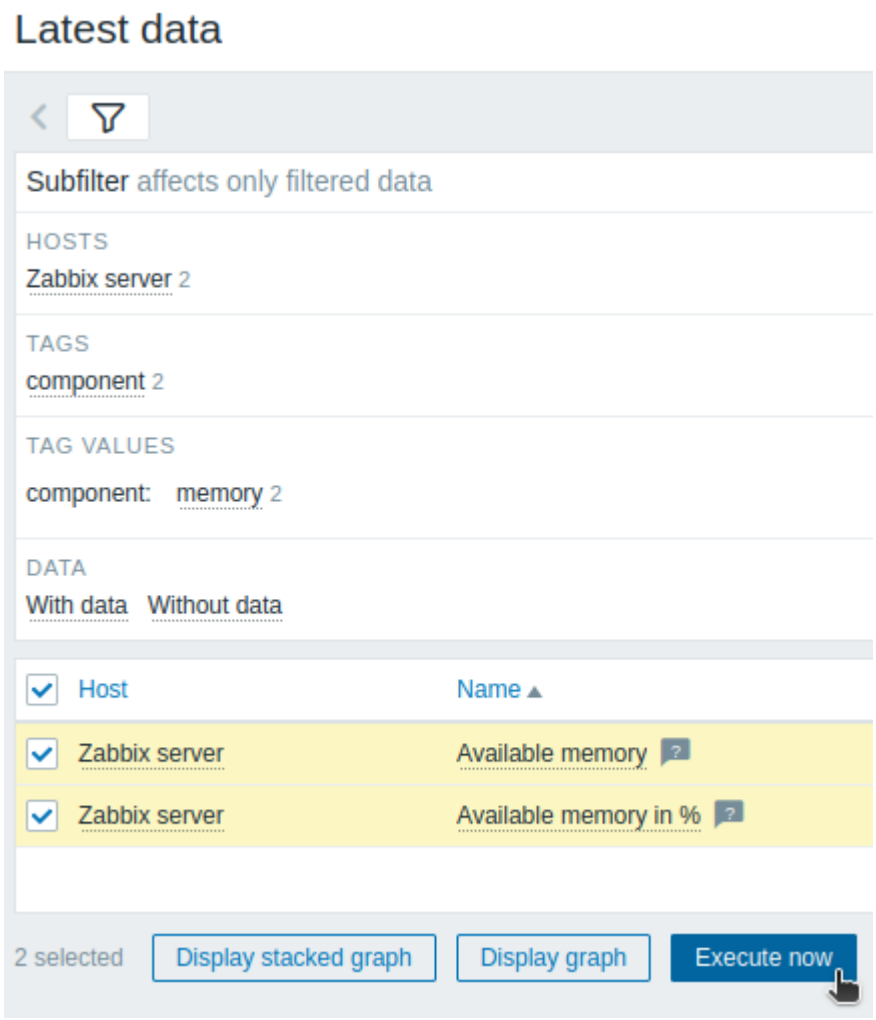
Vous pouvez exécuter des vérifications passives depuis :

- *Monitoring* > *Latest data*
- *Data collection* > *Hosts* > *Items* ou *Discovery*
- Le menu d'élément
- Lors de la modification d'éléments ou de règles de découverte de bas niveau (dans leurs formulaires de configuration)

Attention:
La vérification doit exister dans le cache de configuration pour être exécutée ; voir aussi CacheUpdateFrequency. Le cache n'est **pas mis à jour** avant l'exécution de la vérification, donc les modifications récentes de la configuration d'un élément ou d'une règle de découverte de bas niveau ne seront pas prises en compte. Pour tester des éléments ou des règles LLD en cours de création ou nouvellement créés, vous pouvez utiliser l'option Test à la place.

Par exemple, pour exécuter une vérification passive dans *Monitoring* > *Latest data* :

1. Sélectionnez des éléments ; vous pouvez exécuter la vérification pour plusieurs éléments à la fois.
2. Cliquez sur *Execute now*.



Note:
Dans *Latest data*, les utilisateurs peuvent *Execute now* sur des éléments des hôtes pour lesquels ils disposent d'une permission *read-write*, ou d'une permission *read* et de l'action *Invoke "Execute now" on read-only hosts* activée pour leur rôle.

Limitation des vérifications de l'agent

Aperçu

Vous pouvez contrôler quelles clés d'élément Zabbix agent ou agent 2 est autorisé ou interdit d'utiliser lors de l'exécution des vérifications d'élément, des commandes distantes ou des scripts.

Pour ce faire, utilisez ces paramètres de configuration de l'agent pour définir des règles d'autorisation/interdiction :

- `AllowKey=<pattern>`
- `DenyKey=<pattern>`
- `AllowKeyRegexp=<pattern>`
- `DenyKeyRegexp=<pattern>`

Le <pattern> doit contenir une seule clé d'élément.
 Pour `AllowKey` et `DenyKey`, le <pattern> prend en charge les caractères génériques (*). Le caractère générique correspond à n'importe quel nombre de caractères à sa position et peut être utilisé pour faire correspondre des clés d'élément ou des paramètres (par exemple, `vfs.file.*[*]`).
 Pour `AllowKeyRegexp` et `DenyKeyRegexp`, le <pattern> prend en charge les expressions régulières. Le modèle d'expression régulière peut être utilisé pour faire correspondre des clés d'élément et leurs chaînes de paramètres. Voir aussi des exemples d'échappement correct.
 Veuillez noter que Zabbix agent et Zabbix agent 2 peuvent prendre en charge une syntaxe d'expressions régulières différente.

Attention:
Pour améliorer la sécurité, il est recommandé d'utiliser des clés d'élément exactes plutôt que des caractères génériques pour AllowKey et DenyKey. Il est également recommandé d'utiliser les modèles d'expressions régulières les plus spécifiques pour AllowKeyRegexp et DenyKeyRegexp. Pour plus de détails, voir Sécurisation des règles d'autorisation/interdiction.

Contrairement aux autres paramètres de configuration de l'agent, vous pouvez spécifier un nombre illimité de paramètres AllowKey, DenyKey, AllowKeyRegexp ou DenyKeyRegexp.

Notes importantes

- Tous les éléments system.run sont désactivés par défaut (même lorsque DenyKey et DenyKeyRegexp sont vides), comme si DenyKey=system.run[*] ou DenyKeyRegexp=^system\.run\[.*\]$ était défini comme dernière règle. De ce fait, vous pouvez autoriser des éléments system.run spécifiques sans interdire explicitement les autres éléments system.run.
- Si possible, utilisez AllowKey/AllowKeyRegexp pour autoriser uniquement les éléments requis et refuser tout le reste. Certaines clés peuvent être exploitées pour lire des fichiers non prévus via une traversée de chemin (par exemple, vfs.file.contents["../../../../etc/passwd"]), et les nouvelles versions de l'agent Zabbix peuvent introduire des clés non couvertes par vos règles DenyKey/DenyKeyRegexp.
- La configuration AllowKey, DenyKey, AllowKeyRegexp et DenyKeyRegexp n'affecte pas les paramètres d'agent HostnameItem, HostMetadataItem ou HostInterfaceItem.
- Les éléments refusés deviennent non pris en charge sans indication ni message d'erreur ; par exemple :
  - La commande en ligne de commande de l'agent Zabbix --print (-p) n'affichera pas les clés d'éléments refusées.
  - La commande en ligne de commande de l'agent Zabbix --test (-t) renverra "Unsupported item key." pour les clés d'éléments refusées.
  - Lorsque la journalisation est activée (LogRemoteCommands=1), le fichier journal de l'agent Zabbix n'enregistrera pas les commandes à distance refusées.

Ordre des règles allow/deny

Vous pouvez spécifier un nombre illimité de règles AllowKey, DenyKey, AllowKeyRegexp ou DenyKeyRegexp, bien que leur ordre soit important.

- Les règles sont évaluées une par une, de haut en bas.
- Lorsqu'une clé d'élément correspond à une règle, elle est soit autorisée, soit refusée, et l'évaluation des règles s'arrête.

Par exemple, lors de l'évaluation de vfs.file.contents[/etc/passwd], les règles sont traitées comme suit :

```
AllowKey=vfs.file.contents[/tmp/app.log]    # Le modèle de clé d'élément ne correspond pas ; l'agent passe
AllowKey=vfs.file.contents[/etc/passwd]     # Le modèle de clé d'élément correspond ; l'agent autorise la
DenyKey=vfs.file.*[*]                       # L'agent ignore la règle, car l'évaluation s'est arrêtée.
```

L'ordre de règles suivant refusera la vérification de l'élément :

```
DenyKey=vfs.file.*[*]                       # Le modèle de clé d'élément correspond ; l'agent refuse la vé
AllowKey=vfs.file.contents[/etc/passwd]     # L'agent ignore la règle, car l'évaluation s'est arrêtée.
AllowKey=vfs.file.contents[/tmp/app.log]    # L'agent ignore la règle, car l'évaluation s'est arrêtée.
```

Exemples

Les exemples suivants montrent des modèles de configuration courants pour AllowKey, DenyKey, AllowKeyRegexp et DenyKeyRegexp.

Autoriser des vérifications et des commandes spécifiques

Autorisez uniquement deux vérifications d'élément vfs.file et deux commandes system.run :

en utilisant AllowKey et DenyKey :

```
AllowKey=vfs.file.contents[/tmp/app.log]
AllowKey=vfs.file.size[/tmp/app.log]
AllowKey=system.run[/usr/bin/uptime]
AllowKey=system.run[/usr/bin/df -h /]
DenyKey=vfs.file.*[*]
```

en utilisant AllowKeyRegexp et DenyKeyRegexp :

```
AllowKeyRegexp=^vfs\.file\.(contents|size)\[/tmp/app\.log\]$
AllowKeyRegexp=^system\.run\[/usr/bin/(uptime|df -h /)\]$
DenyKeyRegexp=^vfs\.file\..*\[.*\]$
```

Note:
Il est inutile de définir `DenyKey=system.run[*]` ou `DenyKeyRegexp=^system\.run\[.*\]$`, car toutes les autres commandes `system.run` sont refusées par défaut.

Autoriser les scripts

Autorisez l'agent Zabbix à exécuter des scripts sur les hôtes via toutes les méthodes disponibles :

- Scripts globaux pouvant être exécutés dans l'interface ou via l'API (cette méthode utilise toujours la clé `system.run[myscript.sh]`)
- Commandes à distance issues des opérations d'action (cette méthode utilise toujours la clé `system.run[myscript.sh,nowait]`)
- Éléments `system.run` de l'agent Zabbix avec le script, par exemple :
  - `system.run[myscript.sh]`
  - `system.run[myscript.sh,wait]`
  - `system.run[myscript.sh,nowait]`

```
AllowKey=system.run[myscript.sh,*]
```

ou

```
AllowKeyRegexp=^system\.run\[myscript\.sh,.*\]$
```

Pour contrôler le paramètre `wait/nowait`, vous devez définir une règle différente. Par exemple, vous pouvez autoriser uniquement les éléments `system.run[myscript.sh,wait]`, excluant ainsi les autres méthodes :

```
AllowKey=system.run[myscript.sh,wait]
```

ou

```
AllowKeyRegexp=^system\.run\[myscript\.sh,wait\]$
```

Sécurisation des règles allow/deny

Ces exemples montrent comment sécuriser des règles `AllowKey/AllowKeyRegexp` ou `DenyKey/DenyKeyRegexp` trop permissives.

Considérez les règles suivantes :

en utilisant `AllowKey` et `DenyKey` :

```
AllowKey=system.run["C:\Program^ Files\Zabbix^ Agent^ 2\scripts\test.bat*"]
DenyKey=vfs.file.*
DenyKey=system.cpu.load[*]
```

en utilisant `AllowKeyRegexp` et `DenyKeyRegexp` :

```
AllowKeyRegexp=^system\.run\["C:\\Program\sFiles\\Zabbix\sAgent\s2\\scripts\\test\.bat.*"\]$
DenyKeyRegexp=^vfs\.file\..*$
DenyKeyRegexp=^system\.cpu\.load\[.*\]$
```

Note:
Sous Windows, vous devez échapper les espaces dans le chemin à l'aide d'un caret (^) pour `AllowKey` ou `DenyKey`, et de `\s` pour `AllowKeyRegexp` ou `DenyKeyRegexp`.

Ces règles contiennent un caractère générique (*) (ou `.*` pour les motifs basés sur des expressions régulières), qui peut être exploité de manière abusive :

- Le script `test.bat` peut être exécuté avec n'importe quels arguments, y compris des arguments non souhaités.
- Le motif `vfs.file.*` (`^vfs\.file\..*$`) correspond à des clés d'élément avec ou sans paramètres ; cependant, tous les éléments `vfs.file` nécessitent des paramètres.
- Le motif `system.cpu.load[*]` (`^system\.cpu\.load\[.*\]$`) correspond uniquement aux clés d'élément avec paramètres ; cependant, les éléments `system.cpu.load` ne nécessitent pas de paramètres.

Pour sécuriser ces règles, autorisez explicitement l'exécution de `test.bat` uniquement avec des arguments spécifiques, et refusez les motifs de clés d'élément corrects ; par exemple :

en utilisant `AllowKey` et `DenyKey` :

```
AllowKey=system.run["C:\Program^ Files\Zabbix^ Agent^ 2\scripts\test.bat status"]
AllowKey=system.run["C:\Program^ Files\Zabbix^ Agent^ 2\scripts\test.bat version"]
DenyKey=vfs.file.*[*]
DenyKey=system.cpu.load
DenyKey=system.cpu.load[*]
```

en utilisant AllowKeyRegexp et DenyKeyRegexp :

```
AllowKeyRegexp=^system\.run\["C:\\Program\sFiles\\Zabbix\sAgent\s2\\scripts\\test\.bat (status|version)"\]
DenyKeyRegexp=^vfs\.file\..+\[.*\]$
DenyKeyRegexp=^system\.cpu\.load(\[.*\])?$
```

Vous pouvez tester les règles en exécutant les commandes suivantes, qui renverront ZBX_NOTSUPPORTED.

```
cd "C:\Program Files\Zabbix Agent 2"
zabbix_agent2.exe -t system.run["C:\Program^ Files\Zabbix^ Agent^ 2\scripts\test.bat debug"]
zabbix_agent2.exe -t vfs.file.size["C:\ProgramData\MyApp\config.ini"]
zabbix_agent2.exe -t vfs.file.contents["C:\Windows\System32\drivers\etc\hosts"]
zabbix_agent2.exe -t system.cpu.load
zabbix_agent2.exe -t system.cpu.load[all,avg1]
```

Exemples de motifs

Les tableaux suivants montrent comment les motifs de clé d'élément sont mis en correspondance :

- Une clé correspond au motif uniquement si elle satisfait à **toutes** les conditions de la colonne *Matches*.
- Les paramètres doivent être entièrement encadrés par des crochets (par exemple, `vfs.file.contents[*` et `vfs.file.contents*utf8]` sont des motifs invalides).
- Les exemples illustrent uniquement la correspondance des motifs ; les éléments réels nécessitent des paramètres.

Pour AllowKey et DenyKey :

Pattern	Matches	Examples
`*`	Toute clé avec ou sans paramètres	

`vfs.file.*`	La clé commence par `vfs.file.`
	Avec ou sans paramètres
		Correspond à :
		`vfs.file.size`
		`vfs.file.contents`
		`vfs.file.contents[]`
		`vfs.file.size[/var/log/app.log]`

`vfs.*.contents`	La clé commence par `vfs.`
	La clé se termine par `.contents`
	Aucun paramètre
		Correspond à :
		`vfs..contents`
		`vfs.mount.point.file.contents`
		
		Ne correspond pas à :
		`vfs.contents`
		`vfs.file.contents[]`

`vfs.file.*[*]`	La clé commence par `vfs.file.`
	Paramètres quelconques ou vides
		Correspond à
		`vfs.file.get.custom[]`
		`vfs.file.size[/var/log/app.log, utf8]`
		
		Ne correspond pas à :
		`vfs.file.get.custom`

`vfs.file.contents`	La clé est `vfs.file.contents`
	Aucun paramètre
		Correspond à :
		`vfs.file.contents`
		
		Ne correspond pas à :
		`vfs.file.contents[/etc/passwd]`

`vfs.file.contents[]`	La clé est `vfs.file.contents[]`
	Paramètres vides
		Correspond à :
		`vfs.file.contents[]`
		
		Ne correspond pas à :
		`vfs.file.contents`

Pattern	Matches	Examples
vfs.file.contents[*]	La clé est vfs.file.contents Paramètres quelconques ou vides	Correspond à : vfs.file.contents[/path/to/file] Ne correspond pas à : vfs.file.contents
vfs.file.contents[/etc/passwd,*]	La clé est vfs.file.contents Le premier paramètre est /etc/passwd Le deuxième paramètre est quelconque ou vide	Correspond à : vfs.file.contents[/etc/passwd,] vfs.file.contents[/etc/passwd,utf8] Ne correspond pas à : vfs.file.contents[] vfs.file.contents[/etc/passwd]
vfs.file.contents[*passwd*]	La clé est vfs.file.contents Paramètres quelconques, dont au moins un contient passwd	Correspond à : vfs.file.contents[/etc/passwd] vfs.file.contents[/etc/passwd,] vfs.file.contents[/etc/passwd,utf8]
vfs.file.contents[*passwd*,*]	La clé est vfs.file.contents Le premier paramètre contient passwd Le deuxième paramètre est quelconque ou vide	Correspond à : vfs.file.contents[/etc/passwd,] vfs.file.contents[/etc/passwd,utf8] Ne correspond pas à : vfs.file.contents[/etc/passwd] vfs.file.contents[/tmp/test]
vfs.file.contents[/etc/passwd,utf8]	La clé est vfs.file.contents Le premier paramètre est /etc/passwd Le deuxième paramètre est utf8	Correspond à : vfs.file.contents[/etc/passwd,utf8] Ne correspond pas à : vfs.file.contents[/etc/passwd,] vfs.file.contents[/etc/passwd,utf16]

Pour AllowKeyRegexp et DenyKeyRegexp :

Pattern	Matches	Examples
^.*$	Toute clé avec ou sans paramètres	
^vfs\.file\..*$	La clé commence par vfs.file. Avec ou sans paramètres	Correspond à : vfs.file.size vfs.file.contents vfs.file.contents[] vfs.file.size[/var/log/app.log]
^vfs\..*\.contents$	La clé commence par vfs. La clé se termine par .contents Aucun paramètre	Correspond à : vfs..contents vfs.mount.point.file.contents Ne correspond pas à : vfs.contents vfs.file.contents[]
^vfs\.file\..*\[.*\]$	La clé commence par vfs.file. Paramètres quelconques ou vides	Correspond à vfs.file.get.custom[] vfs.file.size[/var/log/app.log, utf8] Ne correspond pas à : vfs.file.get.custom
^vfs\.file\.contents$	La clé est vfs.file.contents Aucun paramètre	Correspond à : vfs.file.contents Ne correspond pas à : vfs.file.contents[/etc/passwd]

Pattern	Matches	Examples

`^vfs\.file\.contents\[\]$`	La clé est `vfs.file.contents[]`
	Paramètres vides
		Correspond à :
		`vfs.file.contents[]`
		
		Ne correspond pas à :
		`vfs.file.contents`

`^vfs\.file\.contents\[.*\]$`	La clé est `vfs.file.contents`
	Paramètres quelconques ou vides
		Correspond à :
		`vfs.file.contents[/path/to/file]`
		
		Ne correspond pas à :
		`vfs.file.contents`

`^vfs\.file\.contents\[/etc/passwd,.*\]$`	La clé est `vfs.file.contents`
	Le premier paramètre est `/etc/passwd`
	Le deuxième paramètre est quelconque ou vide
		Correspond à :
		`vfs.file.contents[/etc/passwd,]`
		`vfs.file.contents[/etc/passwd,utf8]`
		
		Ne correspond pas à :
		`vfs.file.contents[]`
		`vfs.file.contents[/etc/passwd]`

`^vfs\.file\.contents\[.*passwd.*\]$`	La clé est `vfs.file.contents`
	Paramètres quelconques, dont au moins un contient `passwd`
		Correspond à :
		`vfs.file.contents[/etc/passwd]`
		`vfs.file.contents[/etc/passwd,]`
		`vfs.file.contents[/etc/passwd,utf8]`

`^vfs\.file\.contents\[.*passwd.*,.*\]$`	La clé est `vfs.file.contents`
	Le premier paramètre contient `passwd`
	Le deuxième paramètre est quelconque ou vide
		Correspond à :
		`vfs.file.contents[/etc/passwd,]`
		`vfs.file.contents[/etc/passwd,utf8]`
		
		Ne correspond pas à :
		`vfs.file.contents[/etc/passwd]`
		`vfs.file.contents[/tmp/test]`

`^vfs\.file\.contents\[/etc/passwd,(utf8|windows-1252)\]$`	La clé est `vfs.file.contents`
	Le premier paramètre est `/etc/passwd`
	Le deuxième paramètre est `utf8` ou `windows-1252`
		Correspond à :
		`vfs.file.contents[/etc/passwd,utf8]`
		`vfs.file.contents[/etc/passwd,windows-1252]`
		
		Ne correspond pas à :
		`vfs.file.contents[/etc/passwd,]`
		`vfs.file.contents[/etc/passwd,utf16]`

Détection des problèmes avec les déclencheurs

La détection des problèmes dans Zabbix repose sur les **déclencheurs**.

Un déclencheur décrit la condition du problème, ce qui permet à Zabbix de réagir si le problème se produit.

Le flux de travail complet de la détection des problèmes est le suivant :

1. Assurez-vous de disposer d'un élément qui collecte des données.
2. Définissez le déclencheur pour cet élément. Sans déclencheur, Zabbix se contente de collecter des données, mais n'y réagit jamais.
3. Observez les problèmes détectés :
   - Dans *Monitoring* > *Problems*
   - Dans le widget *Problems*
4. Recevez des notifications, exécutez des commandes à distance (si elles sont définies).

Notez que les modèles officiels Zabbix contiennent des déclencheurs prédéfinis.
Consultez le guide pour appliquer un modèle afin de monitor Linux.

Qu'est-ce qu'un « déclencheur » ?

Un déclencheur décrit la *condition de problème*, au moyen de l'expression de déclencheur.

La condition de problème, lorsqu'elle se produit en situation réelle, correspond au moment où le déclencheur « se déclenche » (un problème est créé dans Zabbix).

La condition de problème peut être basée sur la dernière valeur, la valeur moyenne, une chaîne découverte et de nombreuses autres fonctions. Les déclencheurs ne peuvent pas être créés pour des éléments renvoyant des valeurs avec le type de données binaire ou JSON.

Une expression de déclencheur simple calcule la fonction (par exemple, `max()` pour la valeur maximale) pour les données spécifiées entre parenthèses (généralement l'élément de données et la fenêtre temporelle), puis compare ce calcul à un seuil.

Par exemple :

```
max(/host/vfs.fs.size[/,free],5m)<10G
```

Ce déclencheur se déclenche si l'espace disque libre, mesuré sur 5 minutes, est resté constamment inférieur à 10 Go.

Un déclencheur peut avoir les états suivants :

État du déclencheur	Description	Dans Zabbix
OK	L'expression de déclencheur a été évaluée à « 0 » (ou FALSE)	Aucun problème créé Le problème est résolu (s'il existait)
Problème	L'expression de déclencheur a été évaluée à « 1 » (ou TRUE)	Le problème est créé

Il arrive parfois que l'évaluation de l'expression de déclencheur ne soit pas possible pour une raison quelconque. Voir État d'expression inconnu.

Autres ressources :

- Exemples de déclencheurs
- Configuration d'un déclencheur
- Bonnes pratiques

Fréquence de calcul

Un déclencheur est recalculé chaque fois que le serveur Zabbix reçoit une **nouvelle valeur** faisant partie de l'expression. Lorsqu'une nouvelle valeur est reçue, chaque fonction incluse dans l'expression est recalculée (et pas seulement celle qui a reçu la nouvelle valeur).

De plus, un déclencheur est recalculé toutes les 30 secondes (par le history syncer) si l'expression contient l'un des éléments suivants :

- les fonctions de date et heure
- la fonction nodata()

Les déclencheurs contenant uniquement des fonctions de tendance sont évalués une fois par la plus petite période de temps de l'expression. Bien que de nombreuses fonctions de déclencheur utilisent les données d'historique des éléments, les fonctions de tendance utilisent des données de tendance.

Exemples de déclencheur

Cette page est une collection d'exemples de déclencheur.

Les exemples sont triés par cas d'utilisation :

- Hôte indisponible
- Proxy indisponible
- Cluster SMTP indisponible
- Hôte injoignable
- Redémarrage inattendu
- Modifications dans des fichiers importants
- Modifications dans une requête DNS
- Logiciel non correspondant sur différents hôtes
- Horloges désynchronisées
- Agent obsolète
- Interface instable
- Trafic entrant élevé

- Espace disque faible
- Espace disque faible (seuil dynamique)
- Charge CPU élevée
- Comparaison de la charge CPU
- Comparaison de la charge CPU à long terme
- Chaîne contenant des caractères spéciaux

Détecter un hôte indisponible

```
max(/host/zabbix[host,agent,available],5m)=0
```

Ce déclencheur se déclenche si l'agent Zabbix sur l'hôte est indisponible depuis 5 minutes.
 Fonction : max
 Élément de données : zabbix[host,agent,available]

Alternative :

```
nodata(/host/agent.ping,5m)=1
```

Ce déclencheur se déclenche si aucune donnée n'a été reçue de l'agent Zabbix pendant 5 minutes.
 Fonction : nodata
 Élément de données : agent.ping

Détecter un proxy indisponible

```
fuzzytime(/host/zabbix[proxy,{$PROXY_NAME},lastaccess],1m)=0
```

Ce déclencheur se déclenche si les données du proxy Zabbix ont 1 minute de retard par rapport à l'heure du serveur Zabbix.
 Fonction : fuzzytime
 Élément de données : zabbix[proxy,{$PROXY_NAME},lastaccess]

Détecter un cluster SMTP indisponible

```
last(/smtp1.example.com/net.tcp.service[smtp])=0 and last(/smtp2.example.com/net.tcp.service[smtp])=0
```

Ce déclencheur se déclenche si les deux serveurs SMTP sont indisponibles.
 Fonction : last
 Élément de données : net.tcp.service

Détecter un hôte injoignable

```
count(/host/icmpping,30m,,"0")>5
```

Ce déclencheur se déclenche si l'hôte est injoignable par ping plus de 5 fois au cours des 30 dernières minutes.
 Fonction : count
 Élément de données : icmpping

Détecter un redémarrage inattendu

```
change(/host/system.uptime)<0
```

Ce déclencheur se déclenche si une variation négative de la valeur du temps de fonctionnement du système est détectée (indiquant un redémarrage).
 Fonction : change
 Élément de données : system.uptime

Détecter les modifications dans des fichiers importants

```
last(/host/vfs.file.cksum[/etc/passwd],#1)<>last(/host/vfs.file.cksum[/etc/passwd],#2)
```

Ce déclencheur se déclenche si /etc/passwd a été modifié. L'expression est vraie lorsque la somme de contrôle précédente de /etc/passwd diffère de la plus récente. Des expressions similaires peuvent être utiles pour surveiller les modifications dans des fichiers importants, tels que /etc/passwd, /etc/inetd.conf, /kernel, etc.
 Fonction : last
 Élément de données : vfs.file.cksum

Détecter les changements dans une requête DNS

```
last(/Zabbix server/net.dns.record[192.0.2.1,{$WEBSITE_NAME},{$DNS_RESOURCE_RECORD_TYPE},2,1])<>"{$WEBSITE
```

Remarquez les guillemets autour du second opérande.

Ce déclencheur se déclenche si le résultat de la requête est différent de ce qu'elle renvoie normalement :

```
example.com             MX       0 mail.example.com
```

Fonction: last
 Élément de données: net.dns.record[192.0.2.1,{$WEBSITE_NAME},{$DNS_RESOURCE_RECORD_TYPE},2,1], avec les macros définies comme suit :

```
{$WEBSITE_NAME} = example.com
{$DNS_RESOURCE_RECORD_TYPE} = MX
```

Détecter les logiciels non correspondants sur différents hôtes

```
last(/host/vfs.file.contents[/etc/os-release])<>last(/host2/vfs.file.contents[/etc/os-release])
```

Ce déclencheur se déclenche si la version d'Ubuntu est différente sur différents hôtes. Notez que les opérandes ici sont des fonctions qui renvoient des chaînes.
 Fonction : `last`
 Élément de données : `vfs.file.contents`

Détecter les horloges désynchronisées

```
fuzzytime(/host/system.localtime,10s)=0
```

Le déclencheur se déclenche si l'heure locale du client et l'heure du serveur Zabbix diffèrent de plus de 10 secondes.
 Fonction: `fuzzytime`
 Élément de données: `system.localtime`

Notez que `system.localtime` doit être configuré comme une vérification passive pour l'agent Zabbix; sur Zabbix agent 2, il peut être configuré comme une vérification active.

Détecter un agent obsolète

```
find(/host/agent.version,,"like","beta")=1
```

Ce déclencheur se déclenche si l'agent Zabbix a une version bêta. L'agent Zabbix doit être mis à niveau.
 Fonction : `find`
 Élément de données : `agent.version`

Détecter une interface instable

```
changecount(/host/vfs.file.contents["/sys/class/net/eth0/operstate"],1h)>5
```

Ce déclencheur se déclenche si l'état opérationnel (up/down/unknown) de eth0 a changé plus de 5 fois en une heure.
 Fonction : `changecount`
 Élément de données : `vfs.file.contents`

Détecter un trafic entrant élevé

```
min(/host/net.if.in[eth0,bytes],5m)>100K
```

Ce déclencheur se déclenche si le nombre d'octets reçus sur eth0 au cours des cinq dernières minutes a toujours été supérieur à 100 kilo-octets. Quelqu'un est probablement en train de télécharger un fichier volumineux.
 Fonction : `min`
 Élément de données : `net.if.in[eth0,bytes]`

Détecter un espace disque faible

```
max(/host/vfs.fs.size[/,free],5m)<10G
```

Le déclencheur se déclenche si l'espace disque libre reste constamment (pendant 5 minutes) inférieur à 10 Go.

Vous pouvez également définir une expression de récupération :

```
min(/host/vfs.fs.size[/,free],10m)>40G
```

Le problème est **résolu** uniquement lorsque l'espace disque libre reste constamment (pendant 10 minutes) supérieur à 40 Go.
 Fonction : `min`
 Élément de données : `vfs.fs.size`

Détecter un espace disque faible (seuil dynamique)

```
last(/template/hrStorageFree[{#SNMPVALUE}])<last(/template/hrStorageSize[{#SNMPVALUE}])*0.1
```

Le déclencheur se déclenche si l'espace de stockage libre (en unités d'allocation) descend en dessous de 10 pour cent. Notez que la valeur d'un autre élément est utilisée pour obtenir un seuil de déclenchement adaptatif, applicable au stockage découvert de différentes tailles. Fonction : `last`

Détecter une charge CPU élevée

```
last(/host/system.cpu.load[all,avg1])>5
```

Le déclencheur se déclenche lorsque la charge moyenne du processeur est supérieure à 5 pendant une minute.

Variantes :

```
min(/host/system.cpu.load[all,avg1],5m)>2 and time()<060000
```

```
min(/host/system.cpu.load[all,avg1],5m)>2 and not (dayofweek()=7 and time()>230000) and not (dayofweek()=1
```

Ces déclencheurs analysent 5 minutes de données et se déclenchent uniquement si la charge CPU ne descend jamais en dessous de 2. De plus, ces déclencheurs se déclenchent :

- uniquement pendant la nuit (00:00-06:00)
- à tout moment sauf pendant les 2 heures du changement de semaine (dimanche, 23:00 - lundi, 01:00)

```
(last(/host/system.cpu.load[all,avg1])>5) + (last(/host2/system.cpu.load[all,avg1])>5) + (last(/host3/syst
```

Ce déclencheur se déclenche si la charge du processeur est trop élevée sur au moins deux des trois hôtes.
 Fonctions : `last`, `min`, `dayofweek`, `time`
 Élément de données : `system.cpu.load`

Comparer les charges CPU

```
avg(/Zabbix server/system.cpu.load,1h)/avg(/Zabbix server/system.cpu.load,1h:now-1d)>2
```

Le déclencheur se déclenche si la charge moyenne d'aujourd'hui dépasse de plus de deux fois la charge moyenne de la même heure hier (en utilisant le décalage temporel `now-1d`).
 Fonction : `avg`
 Élément de données : `system.cpu.load`

Comparer les charges CPU à long terme

```
trendavg(/host/system.cpu.load,1M:now/M)>1.1*trendavg(/host/system.cpu.load,1M:now/M-1M)
```

Ce déclencheur se déclenche si la charge CPU sur l'hôte a augmenté de plus de 10 % le mois dernier
 Fonction : `trendavg`
 Élément de données : `system.cpu.load`

Vous pouvez également utiliser le champ Event name dans la configuration du déclencheur pour créer un message d'alerte explicite, par exemple pour recevoir quelque chose comme

```
"La charge du serveur Exchange a augmenté de 24 % en juillet (0.69) par rapport à juin (0.56)"
```

le nom de l'événement doit être défini comme suit :

```
Load of {HOST.HOST} server increased by {{?100*trendavg(//system.cpu.load,1M:now/M)/trendavg(//system.cpu.
```

Il est également utile d'autoriser la fermeture manuelle dans la configuration du déclencheur pour ce type de problème.

Détecter une chaîne contenant des caractères spéciaux

```
last(/host/vfs.file.contents[/tmp/hello])={$HELLO_MACRO}
```

Le déclencheur se déclenche si le contenu de `/tmp/hello` est égal à la chaîne définie dans `{$HELLO_MACRO}` :

```
{$HELLO_MACRO} = \" //hello ?\"
```

Vous pouvez également comparer directement à la chaîne :

```
last(/Zabbix server/vfs.file.contents[/tmp/hello])="\\\" //hello ?\\\""
```

Notez que les caractères spéciaux (\\ et ") sont échappés lorsque la chaîne est comparée directement.

Fonction : `last`
 Élément de données : `vfs.file.contents`

Configuration d'un déclencheur

Cette page décrit comment configurer un déclencheur dans l'interface Zabbix.

Notez que les modèles officiels Zabbix contiennent des déclencheurs prédéfinis. Consultez le guide pour appliquer un modèle à monitor Linux. Il est également possible d'effectuer une mise à jour en masse des déclencheurs configurés.

Autres ressources :

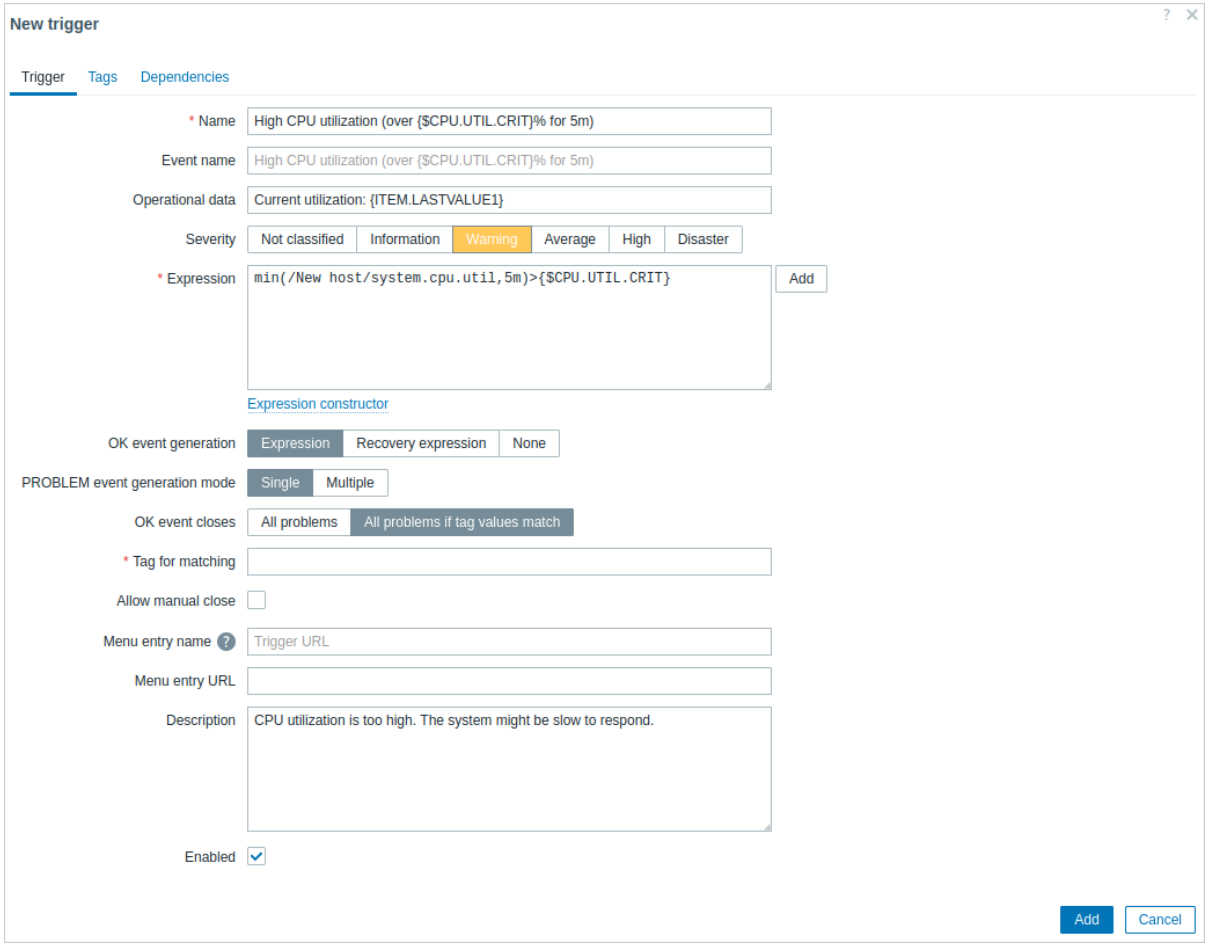
- Détection des problèmes avec les déclencheurs (introduction générale)
- Exemples de déclencheurs (par cas d'utilisation)
- Expression et fonctions des déclencheurs (détails de syntaxe)
- Bonnes pratiques

Configuration

Pour configurer un déclencheur, procédez comme suit :

1. Allez dans *Collecte de données* > *Hôtes*.
2. Cliquez sur *Déclencheurs* dans la ligne de l'hôte.
3. Cliquez sur *Créer un déclencheur* à droite (ou sur le nom du déclencheur pour modifier un déclencheur existant).
4. Saisissez les paramètres du déclencheur dans le formulaire.

L'onglet **Déclencheur** contient tous les attributs essentiels du déclencheur.



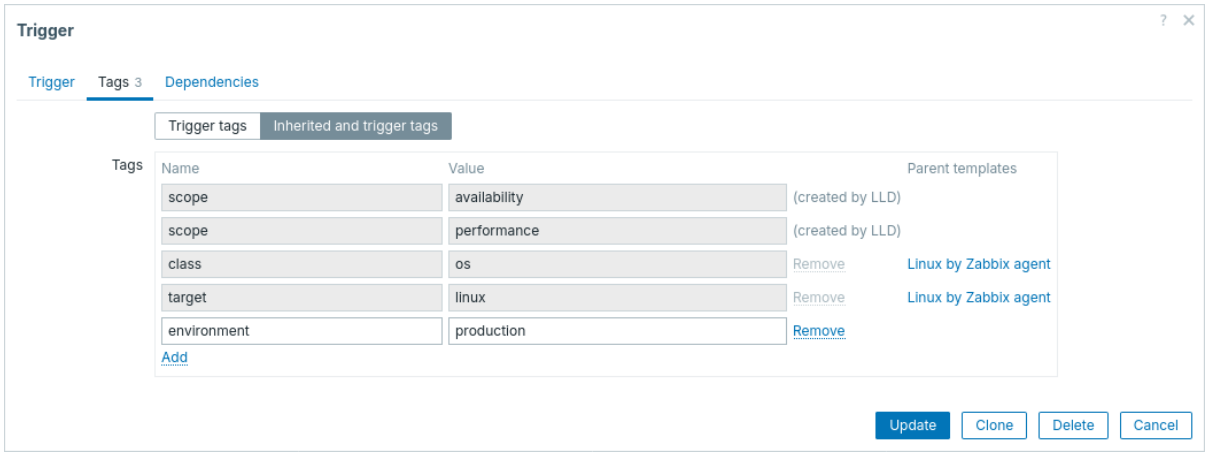
Tous les champs de saisie obligatoires sont marqués d'un astérisque rouge.

Parameter	Description
Name	Nom du déclencheur. Les macros prises en charge sont : {HOST.HOST}, {HOST.NAME}, {HOST.PORT}, {HOST.CONN}, {HOST.DNS}, {HOST.IP}, {ITEM.VALUE}, {ITEM.VALUE.AGE}, {ITEM.VALUE.DATE}, {ITEM.VALUE.TIME}, {ITEM.VALUE.TIMESTAMP}, {ITEM.LASTVALUE}, {ITEM.LASTVALUE.AGE}, {ITEM.LASTVALUE.DATE}, {ITEM.LASTVALUE.TIME}, {ITEM.LASTVALUE.TIMESTAMP}, {ITEM.LOG.*}, et les macros utilisateur {$MACRO}. Les macros **$1, $2...$9** peuvent être utilisées pour faire référence à la première, deuxième... neuvième constante de l'expression. *Remarque* : les macros $1-$9 seront résolues correctement si elles font référence à des constantes dans des expressions relativement simples et directes. Par exemple, le nom "Processor load above $1 on {HOST.NAME}" sera automatiquement remplacé par "Processor load above 5 on New host" si l'expression est last(/New host/system.cpu.load[percpu,avg1])>5.
Event name	Lorsqu'il est défini, ce nom sera utilisé pour créer le nom de l'événement de problème. Par défaut, le nom de l'événement est identique au nom du déclencheur. Le nom de l'événement peut être utilisé pour construire des alertes pertinentes contenant des données de problème (voir exemple). Le même ensemble de macros est pris en charge que pour le nom du déclencheur, ainsi que les macros d'expression {TIME}, {TIMESTAMP} et {?EXPRESSION}.

Parameter	Description
Operational data	Saisissez une chaîne avec des macros pour afficher des données dynamiques en temps réel dans *Surveillance* > *Problèmes*. Vous pouvez également laisser ce champ vide pour afficher les dernières valeurs de tous les éléments de l'expression du déclencheur. Le même ensemble de macros est pris en charge que pour le nom du déclencheur, avec la possibilité d'une résolution dynamique. Par exemple : {ITEM.VALUE<1-9>} se résout en valeurs d'élément au moment où l'état du déclencheur change (problème créé, résolu, fermé manuellement ou fermé par corrélation). {ITEM.LASTVALUE<1-9>} se résout en dernières valeurs d'élément. Notez que la fermeture manuelle d'un problème ne produit pas de nouvelle valeur, donc les deux macros afficheront toujours la valeur du moment du problème. Notez également que les deux macros se résolvent en *UNKNOWN* si la dernière valeur est plus ancienne que la *Période maximale d'affichage de l'historique* (voir *Administration* > *Général*).
Severity	Définissez la gravité requise du déclencheur en cliquant sur les boutons.
Expression	Expression logique utilisée pour définir les conditions d'un problème. Les suffixes de temps et les suffixes de taille de mémoire sont pris en charge. Un problème est créé une fois que toutes les conditions incluses dans l'expression sont remplies, c'est-à-dire lorsque l'expression est évaluée à TRUE. Le problème sera résolu dès que l'expression sera évaluée à FALSE, sauf si des conditions de rétablissement supplémentaires sont spécifiées dans *Recovery expression*.
OK event generation	Options de génération d'événements OK : **Expression** - les événements OK sont générés sur la base de la même expression que les événements de problème ; **Recovery expression** - les événements OK sont générés si l'expression de problème est évaluée à FALSE et que l'expression de rétablissement est évaluée à TRUE ; **None** - dans ce cas, le déclencheur ne reviendra jamais de lui-même à l'état OK.
Recovery expression	Expression logique (facultative) définissant des conditions supplémentaires qui doivent être remplies avant que le problème ne soit résolu, après que l'expression de problème d'origine a déjà été évaluée à FALSE. L'expression de rétablissement est utile pour l'hystérésis des déclencheurs. Il n'est **pas** possible de résoudre un problème uniquement par l'expression de rétablissement si l'expression de problème est toujours TRUE. Ce champ est disponible uniquement si 'Recovery expression' est sélectionné pour *OK event generation*.
PROBLEM event generation mode	Mode de génération des événements de problème : **Single** - un seul événement est généré lorsqu'un déclencheur passe pour la première fois à l'état 'Problem' ; **Multiple** - un événement est généré à chaque évaluation 'Problem' du déclencheur.
OK event closes	Sélectionnez si l'événement OK ferme : **All problems** - tous les problèmes de ce déclencheur ; **All problems if tag values match** - uniquement les problèmes de déclencheur dont les valeurs de balise d'événement correspondent.
Tag for matching	Saisissez le nom de la balise d'événement à utiliser pour la corrélation des événements. Ce champ est affiché si 'All problems if tag values match' est sélectionné pour la propriété *OK event closes* et est obligatoire dans ce cas.
Allow manual close	Cochez cette case pour autoriser la fermeture manuelle des événements de problème générés par ce déclencheur. La fermeture manuelle est possible lors de l'acquittement des événements de problème.
Menu entry name	Si ce champ n'est pas vide, le nom saisi ici (jusqu'à 64 caractères) est utilisé dans plusieurs emplacements de l'interface comme libellé pour l'URL du déclencheur spécifiée dans le paramètre *Menu entry URL*. S'il est vide, le nom par défaut *Trigger URL* est utilisé. Le même ensemble de macros est pris en charge que pour le nom du déclencheur, ainsi que {EVENT.ID}, {HOST.ID} et {TRIGGER.ID}.
Menu entry URL	Si ce champ n'est pas vide, l'URL saisie ici (jusqu'à 2048 caractères) est disponible sous forme de lien dans le menu d'événement dans plusieurs emplacements de l'interface, par exemple lors d'un clic sur le nom du problème dans le widget de tableau de bord *Surveillance* > *Problèmes* ou *Problèmes*. Le même ensemble de macros est pris en charge que pour le nom du déclencheur, ainsi que {EVENT.ID}, {HOST.ID} et {TRIGGER.ID}. Remarque : les macros utilisateur avec des valeurs secrètes ne seront pas résolues dans l'URL.

Parameter	Description
Description	Champ de texte utilisé pour fournir davantage d'informations sur ce déclencheur. Peut contenir des instructions pour corriger un problème spécifique, les coordonnées de la personne responsable, etc. Le même ensemble de macros est pris en charge que pour le nom du déclencheur.
Enabled	Décochez cette case pour désactiver le déclencheur si nécessaire. Les problèmes d'un déclencheur désactivé ne sont plus affichés dans l'interface, mais ne sont pas supprimés.

L'onglet **Tags** vous permet de définir des balises au niveau du déclencheur. Tous les problèmes de ce déclencheur seront balisés avec les valeurs saisies ici.



En outre, l'option *Inherited and trigger tags* vous permet d'afficher les balises définies au niveau du modèle si le déclencheur provient de ce modèle. S'il existe plusieurs modèles avec la même balise, ces balises sont affichées une seule fois et les noms des modèles sont séparés par des virgules. Un déclencheur n'hérite pas des balises au niveau de l'hôte et ne les affiche pas.

Parameter	Description
Name/Value	Définissez des balises personnalisées pour marquer les événements de déclencheur. Les balises sont une paire composée d'un nom de balise et d'une valeur. Vous pouvez utiliser uniquement le nom ou l'associer à une valeur. Un déclencheur peut avoir plusieurs balises portant le même nom, mais avec des valeurs différentes. Les macros utilisateur, les macros utilisateur avec contexte, les macros de découverte de bas niveau et les fonctions de macros avec `{{ITEM.VALUE}}`, `{{ITEM.LASTVALUE}}` sont prises en charge dans les balises d'événement. Les macros de découverte de bas niveau peuvent être utilisées à l'intérieur du contexte de macro. La macro {TRIGGER.ID} est prise en charge dans les valeurs des balises de déclencheur. Elle peut être utile pour identifier les déclencheurs créés à partir de prototypes de déclencheur et, par exemple, pour supprimer les problèmes de ces déclencheurs pendant la maintenance. Si la longueur totale de la valeur développée dépasse 255, elle sera tronquée à 255 caractères. Voir toutes les macros prises en charge pour les balises d'événement. Les balises d'événement peuvent être utilisées pour la corrélation des événements, dans les conditions d'action et seront également visibles dans *Surveillance* > *Problèmes* ou dans le widget *Problèmes*.

L'onglet **Dependencies** contient toutes les dépendances du déclencheur.

Cliquez sur *Add* pour ajouter une nouvelle dépendance.

> **Note:**
> Vous pouvez également configurer un déclencheur en en ouvrant un existant, en cliquant sur le bouton *Clone*, puis en l'enregistrant sous un autre nom.

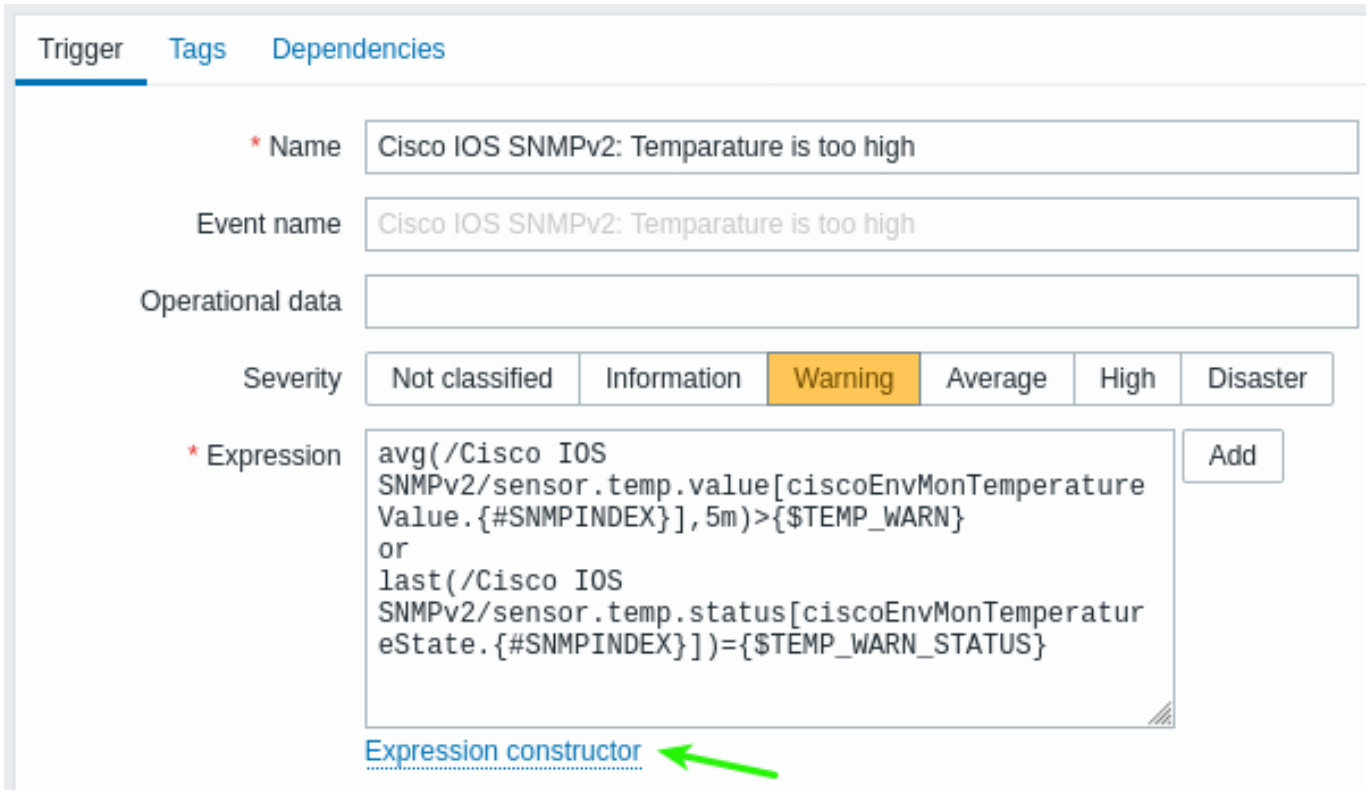
Test d'expression de déclencheur

Il est possible de tester quel serait le résultat de l'expression en fonction de la valeur reçue.

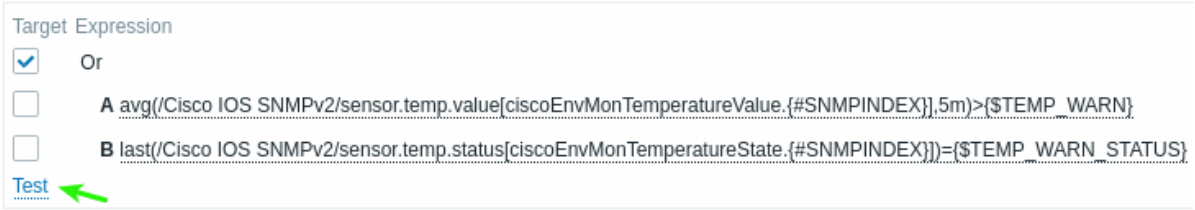
L'expression suivante, issue d'un modèle officiel, est prise comme exemple :

```
avg(/Cisco IOS SNMPv2/sensor.temp.value[ciscoEnvMonTemperatureValue.{#SNMPINDEX}],5m)>{$TEMP_WARN}
or
last(/Cisco IOS SNMPv2/sensor.temp.status[ciscoEnvMonTemperatureState.{#SNMPINDEX}])={$TEMP_WARN_STATUS}
```

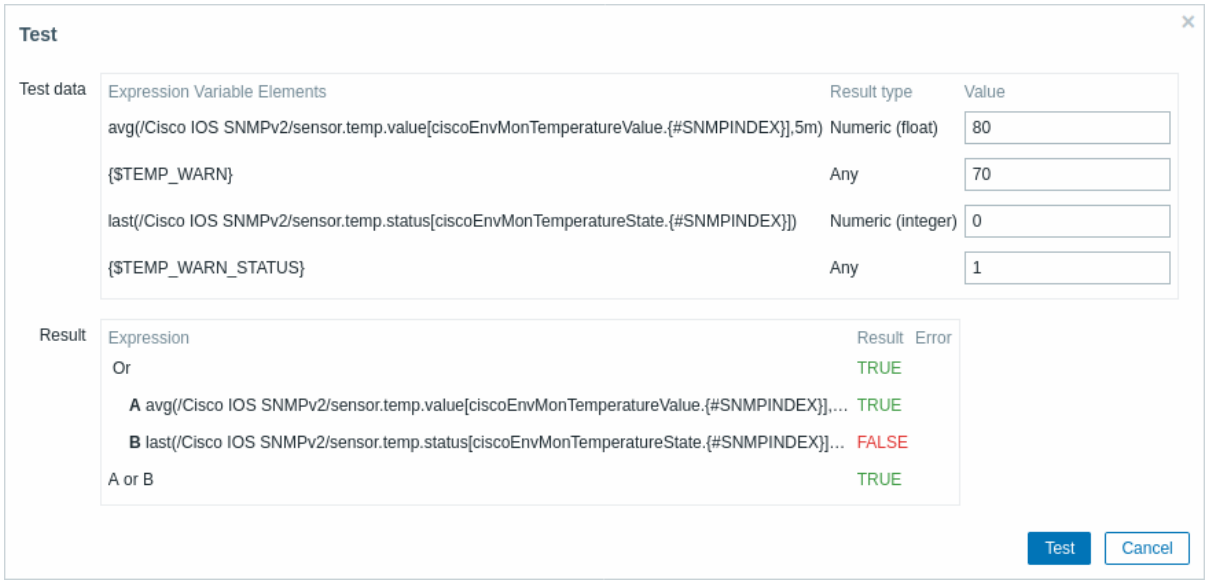
Pour tester l'expression, cliquez sur *Constructeur d'expression* sous le champ de l'expression.



Dans le Constructeur d'expression, toutes les expressions individuelles sont listées. Pour ouvrir la fenêtre de test, cliquez sur *Test* sous la liste des expressions.



Dans la fenêtre de test, vous pouvez saisir des valeurs d'exemple ('80', '70', '0', '1' dans cet exemple), puis voir le résultat de l'expression en cliquant sur le bouton *Test*.



Le résultat des expressions individuelles ainsi que de l'expression complète peut être consulté.

"TRUE" signifie que l'expression spécifiée est correcte. Dans ce cas A, "80" est supérieur à la valeur spécifiée par {$TEMP_WARN}, "70" dans cet exemple. Comme prévu, un résultat "TRUE" apparaît.

"FALSE" signifie que l'expression spécifiée est incorrecte. Dans ce cas B, {$TEMP_WARN_STATUS} "1" doit être égal à la valeur spécifiée, "0" dans cet exemple. Comme prévu, un résultat "FALSE" apparaît.

Le type d'expression choisi est "OR". Si au moins une des conditions spécifiées (A ou B dans ce cas) est TRUE, le résultat global sera également TRUE. Cela signifie que la valeur actuelle dépasse la valeur d'avertissement et qu'un problème s'est produit.

Expression de déclencheur et fonctions

Aperçu

Cette page décrit la syntaxe des expressions de déclencheur et les détails de leur évaluation.

La syntaxe d'une expression simple est la suivante :

function(/host/item,time_period)**<operator><constant>**

Dans cette expression, le premier opérande (à gauche de l'opérateur) est une fonction avec ses paramètres entre parenthèses (généralement l'élément de données et la période).

La fonction est utilisée pour analyser les données reçues pendant la période définie, ce qui produit une valeur calculée.

Cette valeur est ensuite comparée au second opérande à l'aide de l'opérateur. Dans cet exemple, le second opérande est une constante, mais il peut également s'agir d'une autre fonction.

Par exemple :

```
min(/Zabbix server/net.if.in[eth0,bytes],5m)>100K
```

Ce déclencheur se déclenche si le nombre d'octets reçus sur eth0 au cours des cinq dernières minutes a toujours été supérieur à 100 kilo-octets. Dans ce cas, l'expression est vraie et un problème est créé.

Les expressions de déclencheur sont extrêmement flexibles. Dans des expressions plus complexes, plusieurs fonctions, opérateurs et constantes peuvent être combinés.

Voir aussi :

- Exemples de déclencheurs
- Détection des problèmes avec les déclencheurs (introduction générale)

Fonctions

Les fonctions permettent d'analyser les valeurs collectées, par exemple en calculant la moyenne ou en recherchant une chaîne spécifique.

Cliquez sur le groupe de fonctions correspondant pour afficher plus de détails.

Groupe de fonctions	Fonctions
Fonctions d'agrégation	avg, bucket_percentile, count, histogram_quantile, item_count, kurtosis, mad, max, min, skewness, stddevpop, stddevsamp, sum, sumofsquares, varpop, varsamp
Fonctions Foreach	avg_foreach, bucket_rate_foreach, count_foreach, exists_foreach, last_foreach, max_foreach, min_foreach, sum_foreach
Fonctions bit à bit	bitand, bitlshift, bitnot, bitor, bitrshift, bitxor
Fonctions de date et d'heure	date, dayofmonth, dayofweek, now, time
Fonctions d'historique	change, changecount, count, countunique, find, first, firstclock, fuzzytime, jsonpath, last, lastclock, logeventid, logseverity, logsource, logtimestamp, monodec, monoinc, nodata, percentile, rate, xmlxpath

Groupe de fonctions	Fonctions
Fonctions de tendance	baselinedev, baselinewma, trendavg, trendcount, trendmax, trendmin, trendstl, trendsum
Fonctions mathématiques	abs, acos, asin, atan, atan2, avg, cbrt, ceil, cos, cosh, cot, degrees, e, exp, expm1, floor, log, log10, max, min, mod, pi, power, radians, rand, round, signum, sin, sinh, sqrt, sum, tan, truncate
Fonctions d'opérateur	between, in
Fonctions prédictives	forecast, timeleft
Fonctions de chaîne	ascii, bitlength, bytelength, char, concat, contains, insert, jsonpath, left, length, ltrim, mid, repeat, replace, right, rtrim, substring, trim, xmlxpath

Sauf indication contraire, ces fonctions sont prises en charge dans :

- Les expressions de déclencheur
- Les éléments calculés
- Les macros d'expression

Les fonctions Foreach sont prises en charge uniquement pour les calculs d'agrégation.

En règle générale, les fonctions renvoient des valeurs numériques à des fins de comparaison. Lorsqu'elles renvoient des chaînes, la comparaison est possible avec les opérateurs **=** et **<>** (voir l'exemple Détecter les logiciels qui ne correspondent pas sur différents hôtes).

Paramètres des fonctions

Les paramètres des fonctions vous permettent de spécifier :

- la clé de l'élément (au format `/host/key`) pour les fonctions faisant référence à l'historique d'un élément d'un hôte
- la période (ainsi que d'autres paramètres spécifiques à la fonction)
- d'autres expressions

Clé d'élément

L'élément référencé doit être dans un état pris en charge (à l'exception de la fonction **nodata()**, qui est également calculée pour les éléments non pris en charge).

L'omission du nom d'hôte dans le premier paramètre (c'est-à-dire comme dans `function(//key,parameter,...)`) n'est prise en charge que dans certains contextes :

- Dans la formule des éléments calculés
- Dans les macros d'expression, qui peuvent être utilisées dans :
  - le champ *Nom de l'événement*
  - le nom du graphique
  - l'étiquette des éléments de carte « Host » et « Trigger »

Dans ces contextes, vous pouvez également utiliser la macro `{HOST.HOST}`. `{HOST.HOST<1-9>}` peut être utilisée dans le cas du champ *Nom de l'événement* et de l'élément de carte « Trigger » pour faire référence à un élément spécifique dans l'expression du déclencheur. Lorsque le nom d'hôte est omis ou remplacé par `{HOST.HOST}` dans ces contextes, la référence pointe vers le premier élément de l'expression du déclencheur ou vers le premier élément du graphique. En dehors de ces contextes pris en charge, l'omission du nom d'hôte dans les expressions de déclencheur entraînera une erreur. Voir l'exemple Comparer les charges CPU à long terme pour une illustration de l'utilisation de la double barre oblique dans les macros de nom d'événement.

Période

Les paramètres spécifiques à la fonction sont placés après la clé de l'élément et séparés de celle-ci par une virgule.

La plupart des fonctions numériques acceptent une **période** comme paramètre. Elle permet de spécifier l'intervalle qui nous intéresse. Elle peut être spécifiée sous forme de période (30s, 10m, 1h) ou de plage de valeurs (#5 - pour les cinq dernières valeurs).

Vous pouvez utiliser des secondes ou des suffixes de temps pour indiquer la période. Précédé d'un symbole dièse, le paramètre a une signification différente :

Expression	Description
sum(/host/key,**10m)**	Somme des valeurs des 10 dernières minutes.
sum(/host/key,**#10)**	Somme des dix dernières valeurs.

Les paramètres précédés d'un symbole dièse ont une signification différente avec la fonction **last** : ils désignent la N-ième valeur précédente. Ainsi, pour les valeurs 30, 70, 20, 60, 50 (de la plus récente à la moins récente) :

- `last(/host/key,#2)` renverrait « 70 »
- `last(/host/key,#5)` renverrait « 50 »

La période est mesurée jusqu'à « maintenant » - où « maintenant » correspond à l'heure du dernier recalcul du déclencheur (voir Fréquence de calcul) ; « maintenant » ne correspond pas à l'heure actuelle du serveur.

La période spécifie soit :

- Toutes les valeurs comprises entre « maintenant - période » et « maintenant » (ou, lorsqu'un décalage temporel est appliqué, entre « maintenant - décalage temporel - période » et « maintenant - décalage temporel »).
- Le nombre spécifié de valeurs passées, jusqu'à « maintenant ».

Si aucune donnée n'existe pour la période d'évaluation spécifiée, le déclencheur ou l'élément calculé utilisant cette fonction devient non pris en charge.

Notez que :

- Si une seule fonction (référençant l'historique des données) est utilisée dans le déclencheur, « maintenant » correspond toujours à la dernière valeur reçue. Par exemple, si la dernière valeur a été reçue il y a une heure, la période sera considérée comme allant jusqu'à la dernière valeur reçue il y a une heure.
- Un nouveau déclencheur est calculé dès que la première valeur est reçue (fonctions d'historique) ; il sera calculé dans les 30 secondes pour les fonctions date et heure et nodata(). Ainsi, le déclencheur sera calculé même si la période définie (par exemple, une heure) ne s'est pas encore écoulée depuis la création du déclencheur. Le déclencheur sera également calculé après la première valeur, même si la période a été définie, par exemple, sur les dix dernières valeurs.

Décalage temporel

Un décalage temporel facultatif est pris en charge avec une durée ou un nombre de valeurs comme paramètre de fonction. Ce paramètre vous permet de faire référence aux données d'une période passée.

Le décalage temporel commence par `now` - qui indique l'heure actuelle, suivi de `+N<time unit>` ou `-N<time unit>` - pour ajouter ou soustraire N unités de temps.

Par exemple, `avg(/host/key,1h:now-1d)` renvoie la valeur moyenne sur une heure, il y a un jour.

> **Attention:**
> Le décalage temporel spécifié en mois (M) et en années (y) est pris en charge uniquement pour les fonctions de tendance.
> Les autres fonctions prennent en charge les secondes (s), les minutes (m), les heures (h), les jours (d) et les semaines (w).

Décalage temporel avec des périodes absolues

Les périodes absolues sont prises en charge dans le paramètre de décalage temporel, par exemple, de minuit à minuit pour un jour, du lundi au dimanche pour une semaine, du premier au dernier jour du mois pour un mois.

Le décalage temporel pour les périodes absolues commence par `now` - qui indique l'heure actuelle, suivi d'un nombre quelconque d'opérations temporelles : `/<time unit>` - définit le début et la fin de l'unité de temps, par exemple, de minuit à minuit pour un jour, `+N<time unit>` ou `-N<time unit>` - pour ajouter ou soustraire N unités de temps.

Veuillez noter que la valeur du décalage temporel peut être supérieure ou égale à 0, tandis que la valeur minimale de la période est de 1.

Parameter	Description
1d:now/d	Hier
1d:now/d+1d	Aujourd'hui

Parameter	Description
2d:now/d+1d	Les 2 derniers jours
1w:now/w	La semaine dernière
1w:now/w+1w	Cette semaine

Autres expressions

Les paramètres de fonction peuvent contenir d'autres expressions, comme dans la syntaxe suivante :

```
min(min(/host/key,1h),min(/host2/key2,1h)*10)
```

Notez que d'autres expressions ne peuvent pas être utilisées si la fonction référence l'historique d'un élément. Par exemple, la syntaxe suivante n'est **pas** autorisée :

~~`min(/host/key,#5*10)`~~

Bien que les autres expressions de déclencheur en tant que paramètres de fonction soient limitées aux fonctions sans historique dans les déclencheurs, cette limitation ne s'applique pas aux éléments calculés.

Opérateurs

Les opérateurs suivants sont pris en charge pour les déclencheurs **(par ordre décroissant de priorité d'exécution)** :

Priorité	Opérateur	Définition	**Remarques pour les valeurs inconnues**	Forcer la conversion de l'opérande en float [1]
1	**-**	Moins unaire	**-**Unknown → Unknown	Oui
2	**not**	NON logique	**not** Unknown → Unknown	Oui
3	*	Multiplication	0 * Unknown → Unknown (oui, Unknown, et non 0 - afin de ne pas perdre Unknown dans les opérations arithmétiques) 1.2 * Unknown → Unknown	Oui
	/	Division	Unknown / 0 → error Unknown / 1.2 → Unknown 0.0 / Unknown → Unknown	Oui
4	+	Plus arithmétique	1.2 + Unknown → Unknown	Oui
	-	Moins arithmétique	1.2 - Unknown → Unknown	Oui
5	<	Inférieur à. L'opérateur est défini comme suit : A<B ⇔ (A<B-0.000001)	1.2 < Unknown → Unknown	Oui
	<=	Inférieur ou égal à. L'opérateur est défini comme suit : A<=B ⇔ (A≤B+0.000001)	Unknown <= Unknown → Unknown	Oui
	>	Supérieur à. L'opérateur est défini comme suit : A>B ⇔ (A>B+0.000001)		Oui

Priorité	Opérateur	Définition	**Remarques pour les valeurs inconnues**	Forcer la conversion de l'opérande en float [1]
	>=	Supérieur ou égal à. L'opérateur est défini comme suit : A>=B ⇔ (A≥B-0.000001)		Oui
6	**=**	Égal à. L'opérateur est défini comme suit : A=B ⇔ (A≥B-0.000001) and (A≤B+0.000001)		Non [1]
	<>	Différent de. L'opérateur est défini comme suit : A<>B ⇔ (A<B-0.000001) or (A>B+0.000001)		Non [1]
7	**and**	ET logique	0 **and** Unknown → 0 1 **and** Unknown → Unknown Unknown **and** Unknown → Unknown	Oui
8	**or**	OU logique	1 **or** Unknown → 1 0 **or** Unknown → Unknown Unknown **or** Unknown → Unknown	Oui

[1] Un opérande de type chaîne est tout de même converti en valeur numérique si :

- l'autre opérande est numérique
- un opérateur autre que **=** ou **<>** est utilisé sur un opérande

(Si la conversion échoue, l'opérande numérique est converti en chaîne et les deux opérandes sont comparés comme des chaînes.)

Les opérateurs **not**, **and** et **or** sont sensibles à la casse et doivent être en minuscules. Ils doivent également être entourés d'espaces ou de parenthèses.

Tous les opérateurs, à l'exception de **-** unaire et **not**, ont une associativité de gauche à droite. **-** unaire et **not** ne sont pas associatifs (ce qui signifie que **-(-1)** et **not (not 1)** doivent être utilisés à la place de **--1** et **not not 1**).

Résultat de l'évaluation :

- les opérateurs **<**, **<=**, **>**, **>=**, **=**, **<>** doivent produire '1' dans l'expression du déclencheur si la relation spécifiée est vraie, et '0' si elle est fausse. Si au moins un opérande est Unknown, le résultat est Unknown ;
- **and** pour des opérandes connus doit produire '1' si ses deux opérandes sont différents de '0' ; sinon, il produit '0' ; pour des opérandes inconnus, **and** produit '0' uniquement si un opérande est égal à '0' ; sinon, il produit 'Unknown' ;
- **or** pour des opérandes connus doit produire '1' si l'un de ses opérandes est différent de '0' ; sinon, il produit '0' ; pour des opérandes inconnus, **or** produit '1' uniquement si un opérande est différent de '0' ; sinon, il produit 'Unknown' ;
- le résultat de l'opérateur de négation logique **not** pour un opérande connu est '0' si la valeur de son opérande est différente de '0' ; '1' si la valeur de son opérande est égale à '0'. Pour un opérande inconnu, **not** produit 'Unknown'.

État d'expression inconnu

Il est possible qu'un opérande inconnu apparaisse dans une expression de déclencheur lorsque :

- un élément non pris en charge est utilisé
- l'évaluation d'une fonction, pour un élément pris en charge, se termine par une erreur

Dans ce cas, l'expression du déclencheur est généralement évaluée à `Unknown` (car elle ne peut pas être évaluée)

Il est possible d'être notifié des déclencheurs inconnus.

Exceptions

Malgré un opérande inconnu, les expressions de déclencheur peuvent, dans certains cas, être évaluées à un résultat connu (Problem/OK) :

- La fonction `nodata()` est évaluée indépendamment du fait que l'élément référencé soit pris en charge ou non.
- Les expressions avec AND/OR peuvent être évaluées à un résultat connu dans deux cas :
  - **Cas 1** : `"1 or some_function(unsupported_item1) or some_function(unsupported_item2) or ..."` est évaluée à un résultat connu ('1' ou "Problem"),
  - **Cas 2** : `"0 and some_function(unsupported_item1) and some_function(unsupported_item2) and ..."` est évaluée à un résultat connu ('0' ou "OK").
- Si l'évaluation de la fonction pour un élément pris en charge se termine par une erreur, la valeur de la fonction devient `Unknown` et elle intervient comme opérande inconnu dans l'évaluation ultérieure de l'expression.

Les opérandes inconnus peuvent "disparaître" uniquement dans les expressions logiques, comme décrit ci-dessus. Dans les expressions arithmétiques, les opérandes inconnus conduisent toujours à `Unknown` (sauf division par 0).

Attention:
L'état d'expression inconnu ne modifie pas l'état du déclencheur (Problem/OK). Ainsi, si l'état du déclencheur était "Problem" (voir Cas 1), il reste dans l'état de problème même si la partie connue est résolue ('1' devient '0'), car l'expression est alors évaluée à `Unknown` et cela ne modifie pas l'état du déclencheur.

Si une expression de déclencheur avec plusieurs éléments non pris en charge est évaluée à `Unknown`, le message d'erreur dans l'interface fait référence au dernier élément non pris en charge évalué.

Mise en cache des valeurs

Les valeurs requises pour l'évaluation des déclencheurs sont mises en cache par le serveur Zabbix. Pour cette raison, l'évaluation des déclencheurs entraîne une charge plus élevée sur la base de données pendant un certain temps après le redémarrage du serveur.

Le cache des valeurs n'est pas vidé lorsque les valeurs d'historique des éléments sont supprimées (manuellement ou par le housekeeper) ; le serveur utilisera donc les valeurs mises en cache jusqu'à ce qu'elles soient plus anciennes que les périodes définies dans les fonctions de déclencheur ou jusqu'au redémarrage du serveur.

Note:
S'il n'y a pas de données récentes dans le cache et qu'aucune période d'interrogation n'est définie dans la fonction, Zabbix remontera par défaut jusqu'à une semaine dans le passé pour interroger la base de données à la recherche de valeurs historiques.

Fonctions d'agrégation

Sauf indication contraire, toutes les fonctions répertoriées ici sont prises en charge dans :

- Expressions de déclencheur
- Formules d'élément calculé
- Macros d'expression

Les fonctions d'agrégation peuvent fonctionner avec :

- l'historique des éléments, par exemple, `min(/host/key,1h)`
- fonctions foreach comme seul paramètre, par exemple, `min(last_foreach(/*/key))` (uniquement dans les formules d'élément calculé ; pour les utiliser dans les déclencheurs, référencez l'élément calculé par sa clé)

Les fonctions sont listées sans informations supplémentaires. Cliquez sur la fonction pour voir les détails complets.

Function	Description
avg	La valeur moyenne d'un élément sur la période d'évaluation définie.
bucket_percentile	Calcule le percentile à partir des buckets d'un histogramme.
count	Le nombre de valeurs dans un tableau renvoyé par une fonction foreach.

Function	Description
histogram_quantile	Calcule le φ-quantile à partir des buckets d'un histogramme.
item_count	Le nombre d'éléments existants dans la configuration qui correspondent aux critères de filtre.
kurtosis	L'« aplatissement » de la distribution de probabilité dans les valeurs collectées sur la période d'évaluation définie.
mad	L'écart absolu médian dans les valeurs collectées sur la période d'évaluation définie.
max	La valeur la plus élevée d'un élément sur la période d'évaluation définie.
min	La valeur la plus basse d'un élément sur la période d'évaluation définie.
skewness	L'asymétrie de la distribution de probabilité dans les valeurs collectées sur la période d'évaluation définie.
stddevpop	L'écart type de la population dans les valeurs collectées sur la période d'évaluation définie.
stddevsamp	L'écart type de l'échantillon dans les valeurs collectées sur la période d'évaluation définie.
sum	La somme des valeurs collectées sur la période d'évaluation définie.
sumofsquares	La somme des carrés dans les valeurs collectées sur la période d'évaluation définie.
varpop	La variance de la population des valeurs collectées sur la période d'évaluation définie.
varsamp	La variance de l'échantillon des valeurs collectées sur la période d'évaluation définie.

Paramètres courants

- `/host/key` est le premier paramètre obligatoire commun aux fonctions faisant référence à l'historique de l'élément de l'hôte.
- `(sec|#num)<:time shift>` est le deuxième paramètre commun aux fonctions faisant référence à l'historique de l'élément de l'hôte, où :
  - **sec** - période d'évaluation maximale en secondes (des suffixes de temps peuvent être utilisés).
  - **#num** - plage d'évaluation maximale dans les dernières valeurs collectées (si précédé d'un symbole dièse).
  - **time shift** (facultatif) permet de déplacer le point d'évaluation vers le passé. Consultez plus de détails sur la spécification du décalage temporel.

Détails des fonctions

Quelques remarques générales sur les paramètres des fonctions :

- Les paramètres des fonctions sont séparés par une virgule
- Les paramètres de fonction facultatifs (ou parties de paramètres) sont indiqués par < >
- Les paramètres spécifiques à chaque fonction sont décrits avec chaque fonction
- Les paramètres `/host/key` et `(sec|#num)<:time shift>` ne doivent jamais être entre guillemets

avg(/host/key,(sec|#num)<:time shift>)

La valeur moyenne d'un élément sur la période d'évaluation définie.
 Types de valeurs pris en charge : *Float*, *Integer*.
 Fonctions foreach prises en charge : *avg_foreach*, *count_foreach*, *exists_foreach*, *last_foreach*, *max_foreach*, *min_foreach*, *sum_foreach*.

Paramètres : voir paramètres communs.

Le décalage temporel est utile lorsqu'il est nécessaire de comparer la valeur moyenne actuelle avec la valeur moyenne d'il y a un certain temps.

Exemples :

```
avg(/host/key,1h) #calculate the average value for the last hour until now
avg(/host/key,#5) #calculate the average value of the five latest values
avg(/host/key,1h:now-1d) #calculate the average value for an hour from 25 hours ago to 24 hours ago from n
avg(last_foreach(/*/system.cpu.load[,avg1]?[group="MySQL Servers"])) #calculate the average processor load
avg(/host/proc.num,5m)>300 #trigger if the average number of processes in the last 5 minutes has been abov
```

bucket_percentile(item filter,time period,percentage)

Calcule le percentile à partir des buckets d'un histogramme.

Paramètres :

- **item filter** - voir item filter ;
- **time period** - voir time period ;
- **percentage** - pourcentage (0-100).

Commentaires :

- Pris en charge uniquement dans les éléments calculés ;

- Cette fonction est un alias de `histogram_quantile(percentage/100, bucket_rate_foreach(item filter, time period, 1))`.

Exemples :

```
bucket_percentile(/*/http_request_duration_seconds_bucket[*],5m,95) #calculate the 95th percentile respons
bucket_percentile(/*/apiserver_request_duration_seconds_bucket[*],10m,99) #calculate the 99th percentile A
```

count(func_foreach(item filter,<time period>),<operator>,<pattern>)

Le nombre de valeurs dans un tableau renvoyé par une fonction foreach.
 Fonctions foreach prises en charge : *avg_foreach*, *count_foreach*, *exists_foreach*, *last_foreach*, *max_foreach*, *min_foreach*, *sum_foreach*.

Paramètres :

- **func_foreach** - fonction foreach pour laquelle le nombre de valeurs renvoyées doit être compté. Voir fonctions foreach pour plus de détails. Notez que count_foreach et bucket_rate_foreach prennent en charge des paramètres supplémentaires.
- **item filter** - voir syntaxe du filtre d'élément;
- **time period** - voir période de temps;
- **operator** (doit être entre guillemets doubles). `operators` pris en charge :
eq - égal
ne - différent
gt - supérieur
ge - supérieur ou égal
lt - inférieur
le - inférieur ou égal
like - correspond si contient le motif (sensible à la casse)
bitand - ET bit à bit
regexp - correspondance sensible à la casse avec l'expression régulière donnée dans `pattern`
iregexp - correspondance insensible à la casse avec l'expression régulière donnée dans `pattern`
- **pattern** - le motif requis (les arguments de type chaîne doivent être entre guillemets doubles) ; pris en charge si *operator* est spécifié dans le troisième paramètre.

Commentaires :

- L'utilisation de **count()** avec une fonction foreach liée à l'historique (max_foreach, avg_foreach, etc.) peut avoir des implications sur les performances, alors que l'utilisation de **exists_foreach()**, qui fonctionne uniquement avec les données de configuration, n'aura pas cet effet.
- Les paramètres facultatifs *operator* ou *pattern* ne peuvent pas être laissés vides après une virgule, ils doivent être entièrement omis.
- Avec *bitand* comme troisième paramètre, le quatrième paramètre `pattern` peut être spécifié sous la forme de deux nombres séparés par '/': **number_to_compare_with/mask**. count() calcule le "ET bit à bit" entre la valeur et le *mask* et compare le résultat à *number_to_compare_with*. Si le résultat du "ET bit à bit" est égal à *number_to_compare_with*, la valeur est comptée.
Si *number_to_compare_with* et *mask* sont égaux, seul le *mask* doit être spécifié (sans '/').
- Avec *regexp* ou *iregexp* comme troisième paramètre, le quatrième paramètre `pattern` peut être une expression régulière ordinaire ou globale (commençant par '@'). Dans le cas des expressions régulières globales, la sensibilité à la casse est héritée des paramètres de l'expression régulière globale. Pour la correspondance regexp, les valeurs flottantes seront toujours représentées avec 4 chiffres décimaux après '.'. Notez également que pour les grands nombres, la différence entre la représentation décimale (stockée dans la base de données) et binaire (utilisée par le serveur Zabbix) peut affecter le 4e chiffre décimal.

Exemples :

```
count(max_foreach(/*/net.if.in[*],1h)) #calculer le nombre d'éléments net.if.in ayant reçu des données au
count(last_foreach(/*/vfs.fs.size[*,pused]),"gt",95) #calculer le nombre de systèmes de fichiers dont plus
```

histogram_quantile(quantile,bucket1,value1,bucket2,value2,...)

Calcule le φ-quantile à partir des buckets d'un histogramme.
 Prend en charge la fonction foreach : *bucket_rate_foreach*.

Paramètres :

- **quantile** - $0 \leq \varphi \leq 1$;
- **bucketN, valueN** - paires saisies manuellement (>=2) de paramètres ou la réponse de bucket_rate_foreach.

Commentaires :

- Pris en charge uniquement dans les éléments calculés ;
- Correspond fonctionnellement à 'histogram_quantile' de PromQL ;
- Renvoie -1 si les valeurs du dernier bucket 'Infinity' ("+*inf*") sont égales à 0.

Exemples :

```
histogram_quantile(0.75,1.0,last(/host/rate_bucket[1.0]),"+Inf",last(/host/rate_bucket[Inf]))
histogram_quantile(0.5,bucket_rate_foreach(//item_key,30s))
```

item_count(item filter)

Le nombre d'éléments existants dans la configuration qui correspondent aux critères du filtre.
 Type de valeur pris en charge : *Integer*.

Paramètre :

- **item filter** - critères de sélection des éléments, permet de référencer par groupe d'hôtes, hôte, clé d'élément et tags. Les caractères génériques sont pris en charge. Voir item filter pour plus de détails.

Commentaires :

- Pris en charge uniquement dans les éléments calculés ;
- Fonctionne comme un alias de la fonction *count(exists_foreach(item_filter))*.

Exemples :

```
item_count(/*/agent.ping?[group="Host group 1"]) #calculate the number of hosts with the agent.ping item i
```

kurtosis(/host/key,(sec|#num)<:time shift>)

La « queue » de la distribution de probabilité des valeurs collectées au cours de la période d'évaluation définie. Voir aussi : Kurtosis.
 Types de valeurs pris en charge : *Float*, *Integer*.
 fonction foreach prise en charge : *last_foreach*.

Paramètres : voir paramètres communs.

Exemples :

```
kurtosis(/host/system.cpu.util[,user],5m)>5 #trigger on sharp spikes of CPU utilization over 5 minutes
```

mad(/host/key,(sec|#num)<:décalage temporel>)

L'écart absolu médian des valeurs collectées au cours de la période d'évaluation définie. Voir aussi : Écart absolu médian.
 Types de valeurs pris en charge : *Float*, *Integer*.
 Fonction foreach prise en charge : *last_foreach*.

Paramètres : voir paramètres communs.

Exemples :

```
last(/host/system.cpu.util[,user])>avg(/host/system.cpu.util[,user],1h)+3*mad(/host/system.cpu.util[,user]
```

max(/host/key,(sec|#num)<:time shift>)

La valeur la plus élevée d'un élément pendant la période d'évaluation définie.
 Types de valeurs pris en charge : *Float*, *Integer*.
 Fonctions foreach prises en charge : *avg_foreach*, *count_foreach*, *exists_foreach*, *last_foreach*, *max_foreach*, *min_foreach*, *sum_foreach*.

Paramètres : voir paramètres communs.

Exemples :

```
max(/host/key,1h) - min(/host/key,1h) #calcule la différence entre les valeurs maximale et minimale au cou
max(last_foreach(/*/vfs.fs.size[*,pused]?[group="Linux servers"])) #renvoie l'utilisation de disque la plu
```

min(/host/key,(sec|#num)<:time shift>)

La valeur la plus basse d'un élément pendant la période d'évaluation définie.
 Types de valeurs pris en charge : *Float*, *Integer*.
 Fonctions foreach prises en charge : *avg_foreach*, *count_foreach*, *exists_foreach*, *last_foreach*, *max_foreach*, *min_foreach*, *sum_foreach*.

Paramètres : voir paramètres communs.

Exemples :

```
max(/host/key,1h) - min(/host/key,1h) #calcule la différence entre les valeurs maximale et minimale au cou
min(last_foreach(/*/vfs.file.contents["/sys/class/net/enp0s3/operstate"]?[group="Linux servers"])) #renvoi
```

skewness(/host/key,(sec|#num)<:time shift>)

L'asymétrie de la distribution de probabilité des valeurs collectées au cours de la période d'évaluation définie. Voir aussi : Skewness.
 Types de valeurs pris en charge : *Float*, *Integer*.
 fonction foreach prise en charge : *last_foreach*.

Paramètres : voir les paramètres communs.

Exemples :

```
skewness(/host/vm.memory.size[used],5m)<-20 #trigger on sudden negative drops indicating crashes/restarts
```

stddevpop(/host/key,(sec|#num)<:time shift>)

L'écart-type de population des valeurs collectées au cours de la période d'évaluation définie. Voir aussi : Écart-type.
 Types de valeurs pris en charge : *Float*, *Integer*.
 Fonction foreach prise en charge : *last_foreach*.

Paramètres : voir paramètres communs.

Exemples :

```
stddevpop(/host/system.cpu.util[,user],10m)>7 #déclencheur si l'utilisation du CPU fluctue fortement (l'éc
```

stddevsamp(/host/key,(sec|#num)<:time shift>)

L'écart type de l'échantillon des valeurs collectées pendant la période d'évaluation définie. Voir aussi : Écart type.
 Types de valeurs pris en charge : *Float*, *Integer*.
 Fonction foreach prise en charge : *last_foreach*.

Paramètres : voir paramètres communs.

Au moins deux valeurs de données sont requises pour que cette fonction fonctionne.

Exemples :

```
stddevsamp(/host/system.cpu.util[,user],10m)>7 #déclencheur si l'utilisation du CPU fluctue fortement (l'é
```

sum(/hôte/key,(sec|#num)<:time shift>)

La somme des valeurs collectées pendant la période d'évaluation définie.
 Types de valeurs pris en charge : *Float*, *Integer*.
 Fonctions foreach prises en charge : *avg_foreach*, *count_foreach*, *exists_foreach*, *last_foreach*, *max_foreach*, *min_foreach*, *sum_foreach*.

Paramètres : voir paramètres communs.

Exemples :

```
sum(last_foreach(/*/net.if.in[*]?[group="Linux servers"])) #calcule le trafic réseau entrant total pour to
sum(last_foreach(/*/vfs.fs.size[/,total]?[group="MySQL Servers"])) #calcule l'espace disque total pour tou
sum(last_foreach(/*/net.dns[*,*,*])) #calcule le nombre total de vérifications DNS réussies
```

sumofsquares(/host/key,(sec|#num)<:time shift>)

La somme des carrés des valeurs collectées dans la période d'évaluation définie.
 Types de valeurs pris en charge : *Float*, *Integer*.
 Fonction foreach prise en charge : *last_foreach*.

Paramètres : voir paramètres communs.

Exemples :

```
sumofsquares(/host/calculated.net.if.rate,1m) #la somme des carrés du taux de trafic réseau (entrant vs so
sumofsquares(last_foreach(/*/net.if.in[*]?[group="Linux servers"])) #la somme des carrés des dernières val
```

varpop(/host/key,(sec|#num)<:time shift>)

La variance de population des valeurs collectées dans la période d'évaluation définie. Voir aussi : Variance.
 Types de valeurs pris en charge : *Float*, *Integer*.
 Fonction foreach prise en charge : *last_foreach*.

Paramètres : voir paramètres communs.

Exemples :

```
varpop(/host/system.cpu.util[,user],10m)>50 #déclencheur si l'utilisation du CPU fluctue fortement (la var
```

varsamp(/host/key,(sec|#num)<:time shift>)

La variance d'échantillon des valeurs collectées dans la période d'évaluation définie. Voir aussi : Variance.
 Types de valeurs pris en charge : *Float*, *Integer*.
 fonction foreach prise en charge : *last_foreach*.

Paramètres : voir paramètres communs.

Au moins deux valeurs de données sont nécessaires pour que cette fonction fonctionne.

Exemples :

```
varsamp(/host/system.cpu.util[,user],10m)>50 #déclencheur si l'utilisation du CPU fluctue fortement (la va
```

Voir toutes les fonctions prises en charge.

Fonctions foreach

Vue d'ensemble

Les fonctions Foreach sont utilisées dans les calculs agrégés pour renvoyer une valeur agrégée pour chaque élément sélectionné par le **filtre d'élément** utilisé. Un tableau de valeurs est renvoyé.

Par exemple, la fonction `avg_foreach` renverra un tableau de valeurs, où chaque valeur est la valeur d'historique *moyenne* de l'élément sélectionné, pendant l'intervalle de temps spécifié.

> **Attention:**
> Les fonctions Foreach sont prises en charge uniquement dans les formules de élément calculé. Elles ne peuvent pas être invoquées dans des clés d'élément simples ni utilisées directement pour construire des expressions de déclencheur. Dans les expressions de déclencheur, un élément calculé peut être utilisé en référençant sa clé.

Le filtre d'élément fait partie de la syntaxe utilisée par les fonctions foreach. L'utilisation de caractères génériques est prise en charge dans le filtre d'élément, ce qui permet de sélectionner les éléments requis de manière très flexible.

Fonctions prises en charge

Function	Description
avg_foreach	Renvoie la valeur moyenne pour chaque élément.
bucket_rate_foreach	Renvoie des paires (borne supérieure du bucket, valeur du taux) adaptées à l'utilisation dans la fonction `histogram_quantile()`, où "borne supérieure du bucket" est la valeur du paramètre de clé d'élément définie par le paramètre <parameter number>.
count_foreach	Renvoie le nombre de valeurs pour chaque élément.
exists_foreach	Renvoie '1' pour chaque élément activé.
last_foreach	Renvoie la dernière valeur pour chaque élément.
max_foreach	Renvoie la valeur maximale pour chaque élément.
min_foreach	Renvoie la valeur minimale pour chaque élément.
sum_foreach	Renvoie la somme des valeurs pour chaque élément.

Syntaxe des fonctions

Les fonctions foreach prennent en charge deux paramètres communs : `filtre d'élément` (voir les détails ci-dessous) et `période de temps` :

```
foreach_function(item filter,time period)
```

Par exemple :

```
avg_foreach(/*/mysql.qps?[group="MySQL Servers"],5m)
```

renverra la moyenne sur cinq minutes de chaque élément 'mysql.qps' dans le groupe de serveurs MySQL.

Notez que certaines fonctions prennent en charge des paramètres supplémentaires.

Syntaxe du filtre d'élément

Le filtre d'élément :

```
/host/key[parameters]?[conditions]
```

se compose de quatre parties, où :

- host - nom d'hôte
- key - clé d'élément (sans paramètres)
- parameters - paramètres de la clé d'élément
- conditions - conditions basées sur le groupe d'hôtes et/ou les tags d'élément (sous forme d'expression)

Les espaces sont autorisés uniquement à l'intérieur de l'expression de conditions.

Utilisation des caractères génériques

- Un caractère générique peut être utilisé pour remplacer le nom d'hôte, la clé d'élément ou un paramètre individuel de la clé d'élément.
- Soit l'hôte, soit la clé d'élément doit être spécifié sans caractère générique. Ainsi, `/host/*` et `/*/key` sont des filtres valides, mais `/*/*` est invalide.

- Un caractère générique ne peut pas être utilisé pour une *partie* du nom d'hôte, de la clé d'élément ou d'un paramètre de clé d'élément.
- Un caractère générique ne correspond pas à plus d'un seul paramètre de clé d'élément. Ainsi, un caractère générique doit être spécifié séparément pour chaque paramètre (c'est-à-dire `key[abc,*,*]`).

Expression de conditions

L'expression de conditions prend en charge :

- opérandes :
  - group - groupe d'hôtes
  - tag - tag d'élément
  - "<text>" - constante de chaîne, avec le caractère d'échappement \ pour échapper " et \
- opérateurs de comparaison de chaînes sensibles à la casse : =, <>
- opérateurs logiques : `and`, `or`, `not`
- regroupement avec parenthèses : ( )

La mise entre guillemets des constantes de chaîne est obligatoire. Seule la comparaison complète de chaînes sensible à la casse est prise en charge.

Warning:
Lors de la spécification des tags dans le filtre (c'est-à-dire `tag="tagname:value"`), le deux-points ":" est utilisé comme délimiteur. Tout ce qui suit est considéré comme la valeur du tag. Ainsi, il n'est actuellement pas possible de spécifier un nom de tag contenant ":".

Exemples

Un filtre complexe peut être utilisé, en faisant référence à la clé d'élément, au groupe d'hôtes et aux tags, comme l'illustrent les exemples :

Exemple de syntaxe	Description
`/host/key[abc,*]`	Correspond à des éléments similaires sur cet hôte.
`/*/key`	Correspond au même élément de n'importe quel hôte.
`/*/key?[group="ABC" and tag="tagname:value"]`	Correspond au même élément de n'importe quel hôte du groupe ABC ayant les tags 'tagname:value'.
`/*/key[a,*,c]?[(group="ABC" and tag="Tag1") or (group="DEF" and (tag="Tag2" or tag="Tag3:value"))]`	Correspond à des éléments similaires de n'importe quel hôte du groupe ABC ou DEF avec les tags respectifs.

Tous les éléments référencés doivent exister et collecter des données. Seuls les éléments activés sur des hôtes activés sont inclus dans les calculs. Les éléments dans l'état non pris en charge ne sont pas inclus.

Attention:
Si la clé d'élément d'un élément référencé est modifiée, le filtre doit être mis à jour manuellement.

La spécification d'un groupe d'hôtes parent inclut le groupe parent et tous les groupes d'hôtes imbriqués avec leurs éléments.

Période

Le paramètre **second** permet de spécifier la période pour l'agrégation. La période peut uniquement être exprimée en durée ; la quantité de valeurs (préfixée par **#**) n'est pas prise en charge.

Les symboles d'unités pris en charge peuvent être utilisés dans ce paramètre pour plus de commodité, par exemple `5m` (cinq minutes) au lieu de `300s` (300 secondes), ou `1d` (un jour) au lieu de 86400 (86400 secondes).

Pour la fonction `last_foreach`, la période est un paramètre facultatif (pris en charge depuis Zabbix 7.0) et peut être omise :

```
last_foreach(/*/key?[group="host group"])
```

La période n'est pas prise en charge avec la fonction `exists_foreach`.

Paramètres supplémentaires

bucket_rate_foreach

Un troisième paramètre facultatif est pris en charge par la fonction `bucket_rate_foreach` :

```
bucket_rate_foreach(item filter,time period,<parameter number>)
```

où <parameter number> est la position de la valeur "bucket" dans la clé de l'élément. Par exemple, si la valeur "bucket" dans myItem[aaa,0.2] est 0.2, alors sa position est 2.

La valeur par défaut de <parameter number> est 1.

count_foreach

Les troisième et quatrième paramètres facultatifs sont pris en charge par la fonction count_foreach :

```
count_foreach(item filter,time period,<operator>,<pattern>)
```

Où :

- **operator** est l'opérateur conditionnel pour les valeurs de l'élément (doit être entre guillemets doubles). Opérateurs pris en charge :
eq - égal
ne - différent
gt - supérieur
ge - supérieur ou égal
lt - inférieur
le - inférieur ou égal
like - correspond si contient le motif (sensible à la casse)
bitand - ET bit à bit
regexp - correspondance sensible à la casse avec l'expression régulière donnée dans pattern
iregexp - correspondance insensible à la casse avec l'expression régulière donnée dans pattern.
- **pattern** est le motif requis (les arguments de type chaîne doivent être entre guillemets doubles) ; pris en charge si *operator* est spécifié dans le troisième paramètre.

Commentaires :

- Les paramètres facultatifs *operator* ou *pattern* ne peuvent pas être laissés vides après une virgule, ils doivent être entièrement omis.
- Avec *bitand* comme troisième paramètre, le quatrième paramètre pattern peut être spécifié sous la forme de deux nombres, séparés par '/': **number_to_compare_with/mask**. count_foreach() calcule le "ET bit à bit" à partir de la valeur et du *mask* et compare le résultat à *number_to_compare_with*. Si le résultat du "ET bit à bit" est égal à *number_to_compare_with*, la valeur est comptée.
Si *number_to_compare_with* et *mask* sont égaux, seul le *mask* doit être spécifié (sans '/').
- Avec *regexp* ou *iregexp* comme troisième paramètre, le quatrième paramètre pattern peut être une expression régulière ordinaire ou globale (commençant par '@'). Dans le cas des expressions régulières globales, la sensibilité à la casse est héritée des paramètres de l'expression régulière globale. Pour la correspondance regexp, les valeurs flottantes seront toujours représentées avec 4 chiffres décimaux après '.'. Notez également que pour les grands nombres, la différence entre la représentation décimale (stockée dans la base de données) et binaire (utilisée par le serveur Zabbix) peut affecter le 4e chiffre décimal.

Consultez les calculs agrégés pour plus de détails et des exemples d'utilisation des fonctions foreach.

Comportement en fonction de la disponibilité

Le tableau suivant illustre le comportement de chaque fonction en cas de disponibilité limitée de l'hôte/de l'élément et des données d'historique.

Function	Disabled host	Unavailable host with data	Unavailable host without data	Disabled item	Unsupported item	Data retrieval error (SQL)
avg_foreach	ignorer	retourner la moyenne	ignorer	ignorer	ignorer	ignorer
bucket_rate_foreach	ignorer	retourner le taux du compartiment	ignorer	ignorer	ignorer	ignorer
count_foreach	ignorer	retourner le nombre	0	ignorer	ignorer	ignorer
exists_foreach	ignorer	1	1	ignorer	1	n/a
last_foreach	ignorer	retourner la dernière valeur	ignorer	ignorer	ignorer	ignorer
max_foreach	ignorer	retourner le maximum	ignorer	ignorer	ignorer	ignorer
min_foreach	ignorer	retourner le minimum	ignorer	ignorer	ignorer	ignorer
sum_foreach	ignorer	retourner la somme	ignorer	ignorer	ignorer	ignorer

Si l'élément est *ignoré*, rien n'est ajouté à l'agrégation.

Fonctions bit à bit

Toutes les fonctions répertoriées ici sont prises en charge dans :

- Expressions de déclencheur
- Formules d'élément calculé
- Macros d'expression

Les fonctions sont listées sans informations supplémentaires. Cliquez sur la fonction pour voir les détails complets.

Function	Description
bitand	La valeur du "ET bit à bit" d'une valeur d'élément et d'un masque.
bitlshift	Le décalage binaire vers la gauche d'une valeur d'élément.
bitnot	La valeur du "NON bit à bit" d'une valeur d'élément.
bitor	La valeur du "OU bit à bit" d'une valeur d'élément et d'un masque.
bitrshift	Le décalage binaire vers la droite d'une valeur d'élément.
bitxor	La valeur du "OU exclusif bit à bit" d'une valeur d'élément et d'un masque.

Détails de la fonction

Quelques remarques générales sur les paramètres de fonction :

- Les paramètres de fonction sont séparés par une virgule
- Les expressions sont acceptées comme paramètres
- Les paramètres de fonction facultatifs (ou parties de paramètres) sont indiqués par < >

bitand(value,mask)

La valeur du « ET bit à bit » entre la valeur d'un élément et le masque.
 Types de valeurs pris en charge : *Integer*.

Paramètres :

- **value** - la valeur à vérifier ;
- **mask** (obligatoire) - un entier non signé de 64 bits (0 - 18446744073709551615).

Bien que la comparaison soit effectuée de manière bit à bit, toutes les valeurs doivent être fournies et sont renvoyées en décimal.
Par exemple, la vérification du 3e bit se fait en comparant à 4, et non à 100.

Exemples :

```
bitand(last(/host/key),12)=8 or bitand(last(/host/key),12)=4 #3e ou 4e bit défini, mais pas les deux en mê
bitand(last(/host/key),20)=16 #3e bit non défini et 5e bit défini
```

bitlshift(value,bits to shift)

Le décalage binaire à gauche de la valeur d'un élément.
 Types de valeurs pris en charge : *Integer*.

Paramètres :

- **value** - la valeur à vérifier ;
- **bits to shift** (obligatoire) - le nombre de bits à décaler.

Bien que la comparaison soit effectuée de manière binaire, toutes les valeurs doivent être fournies et sont renvoyées en décimal.
Par exemple, la vérification du 3e bit se fait en comparant à 4, et non à 100.

bitnot(value)

La valeur du « NOT bit à bit » d'une valeur d'élément.
 Types de valeurs pris en charge : *Integer*.

Paramètre :

- **value** - la valeur à vérifier.

Bien que la comparaison soit effectuée de manière bit à bit, toutes les valeurs doivent être fournies et sont renvoyées en décimal.
Par exemple, la vérification du 3e bit se fait en comparant à 4, et non à 100.

bitor(value,mask)

La valeur du « bitwise OR » entre la valeur d'un élément et le masque.
 Types de valeurs pris en charge : *Integer*.

Paramètres :

- **value** - la valeur à vérifier ;
- **mask** (obligatoire) - un entier non signé sur 64 bits (0 - 18446744073709551615).

Bien que la comparaison soit effectuée au niveau des bits, toutes les valeurs doivent être fournies et sont renvoyées en décimal. Par exemple, la vérification du 3e bit se fait en comparant à 4, et non à 100.

bitrshift(value,bits to shift)

Le décalage binaire à droite de la valeur d'un élément.
 Types de valeurs pris en charge : *Integer*.

Paramètres :

- **value** - la valeur à vérifier ;
- **bits to shift** (obligatoire) - le nombre de bits à décaler.

Bien que la comparaison soit effectuée de manière binaire, toutes les valeurs doivent être fournies et sont renvoyées en décimal. Par exemple, la vérification du 3e bit se fait en comparant à 4, et non à 100.

bitxor(value,mask)

La valeur du « OU exclusif bit à bit » entre la valeur d'un élément et le masque.
 Types de valeurs pris en charge : *Integer*.

Paramètres :

- **value** - la valeur à vérifier ;
- **mask** (obligatoire) - un entier non signé de 64 bits (0 - 18446744073709551615).

Bien que la comparaison soit effectuée de manière bit à bit, toutes les valeurs doivent être fournies et sont renvoyées en décimal. Par exemple, la vérification du 3e bit se fait en comparant à 4, et non à 100.

Voir toutes les fonctions prises en charge.

Fonctions de date et d'heure

Toutes les fonctions répertoriées ici sont prises en charge dans :

- Expressions de déclencheur
- Formules d'élément calculé
- Macros d'expression

> **Attention:**
> Les fonctions de date et d'heure ne peuvent pas être utilisées seules dans l'expression ; au moins une fonction d'un autre groupe, faisant référence à l'élément de l'hôte, doit être incluse dans l'expression (à l'exception de la fonction nodata()). Pour des informations détaillées sur le fonctionnement des fonctions de date et d'heure dans les expressions, voir Heure de calcul.

Les fonctions sont répertoriées sans informations supplémentaires. Cliquez sur la fonction pour voir tous les détails.

Function	Description
date	La date actuelle au format YYYYMMDD.
dayofmonth	Le jour du mois, dans une plage de 1 à 31.
dayofweek	Le jour de la semaine, dans une plage de 1 à 7.
now	Le nombre de secondes écoulées depuis l'Epoch (00:00:00 UTC, 1er janvier 1970).
time	L'heure actuelle au format HHMMSS.

Détails de la fonction

date

La date actuelle au format YYYYMMDD.

Exemple :

```
date()<20220101
```

dayofmonth

Le jour du mois dans la plage de 1 à 31.

Exemple :

```
dayofmonth()=1
```

dayofweek

Le jour de la semaine dans la plage de 1 à 7 (lun. - 1, dim. - 7).

Exemple (uniquement les jours ouvrés) :

```
dayofweek()<6
```

Exemple (uniquement le week-end) :

```
dayofweek()>5
```

now

Le nombre de secondes écoulées depuis l'époque Unix (00:00:00 UTC, le 1er janvier 1970).

Exemple :

```
now()<1640998800
```

time

L'heure actuelle au format HHMMSS.

Exemple (uniquement la nuit, 00:00-06:00) :

```
time()<060000
```

Voir toutes les fonctions prises en charge.

Fonctions d'historique

Toutes les fonctions répertoriées ici sont prises en charge dans :

- Expressions de déclencheur
- Formules d'éléments calculés
- Macros d'expression

Les fonctions sont répertoriées sans informations supplémentaires. Cliquez sur une fonction pour afficher tous les détails.

Fonction	Description
change	La différence entre la valeur précédente et la dernière valeur.
changecount	Le nombre de changements entre les valeurs adjacentes au cours de la période d'évaluation définie.
count	Le nombre de valeurs au cours de la période d'évaluation définie.
countunique	Le nombre de valeurs uniques au cours de la période d'évaluation définie.
find	Rechercher une correspondance de valeur au cours de la période d'évaluation définie.
first	La première valeur (la plus ancienne) au cours de la période d'évaluation définie.
firstclock	L'horodatage de la première valeur (la plus ancienne) au cours de la période d'évaluation définie.
fuzzytime	Vérifier l'écart entre l'heure de l'agent passif et celle du serveur/proxy Zabbix.
jsonpath	Retourner un tableau des résultats de la requête JSONPath.
last	La valeur la plus récente.
lastclock	L'horodatage de la N-ième valeur la plus récente au cours de la période d'évaluation définie.
logeventid	Vérifier si l'ID d'événement de la dernière entrée de journal correspond à une expression régulière.
logseverity	La gravité du journal de la dernière entrée de journal.
logsource	Vérifier si la source du journal de la dernière entrée de journal correspond à une expression régulière.
logtimestamp	L'horodatage du message de journal de la N-ième valeur la plus récente de l'élément de journal.
monodec	Vérifier si les valeurs ont connu une diminution monotone.
monoinc	Vérifier si les valeurs ont connu une augmentation monotone.
nodata	Vérifier qu'aucune donnée n'a été reçue.
percentile	Le P-ième percentile d'une période, où P (pourcentage) est spécifié par le troisième paramètre.
rate	Le taux moyen par seconde de l'augmentation d'un compteur à croissance monotone au cours de la période définie.
xmlxpath	Retourner un tableau des résultats de la requête XPath XML.

Paramètres courants

- `/host/key` est un premier paramètre obligatoire commun aux fonctions faisant référence à l'historique de l'élément de l'hôte.
- `(sec|#num)<:time shift>` est un deuxième paramètre commun aux fonctions faisant référence à l'historique de l'élément de l'hôte, où :
  - **sec** - période d'évaluation maximale en secondes (les suffixes de temps peuvent être utilisés), ou
  - **#num** - plage d'évaluation maximale en dernières valeurs collectées (si précédée d'un symbole dièse).
  - **time shift** (facultatif) permet de déplacer le point d'évaluation vers le passé. Consultez plus de détails sur la spécification du décalage temporel.

Détails des fonctions

Quelques remarques générales sur les paramètres des fonctions :

- Les paramètres des fonctions sont séparés par une virgule
- Les paramètres de fonction facultatifs (ou parties de paramètres) sont indiqués par < >
- Les paramètres spécifiques à une fonction sont décrits avec chaque fonction
- Les paramètres `/host/key` et `(sec|#num)<:time shift>` ne doivent jamais être placés entre guillemets

change(/host/key)

La différence entre la valeur précédente et la dernière valeur.
 Types de valeurs pris en charge : *Float*, *Integer*, *String*, *Text*, *Log*.
 Pour les chaînes, renvoie : 0 - les valeurs sont égales ; 1 - les valeurs diffèrent.

Paramètres : voir paramètres communs.

Commentaires :

- La différence numérique sera calculée, comme illustré par les exemples de valeurs entrantes suivants (« précédente » et « dernière » valeur = différence) :
'1' et '5' = +4
'3' et '1' = −2
'0' et '-2.5' = −2.5.
- Dans les déclencheurs, utile pour détecter des pics soudains (ou des baisses), des réinitialisations de compteurs, des transitions d'état numériques.
- Voir aussi : abs pour la comparaison.

Exemples :

```
change(/host/system.uptime)<0 #system uptime change has been negative since the last value (indicating a r
change(/host/system.cpu.load[all,avg1])>2 #CPU load (for one minute) has jumped by more than 2 since the l
change(/host/vfs.fs.size[/,free])<-1G #free disk space has dropped by more than 1 GB between checks
```

changecount(/host/key,(sec|#num)<:time shift>,<mode>)

Le nombre de changements entre des valeurs adjacentes au cours de la période d'évaluation définie.
 Types de valeurs pris en charge : *Float*, *Integer*, *String*, *Text*, *Log*.

Paramètres :

- Voir paramètres communs
- **mode** (doit être entre guillemets doubles) - valeurs possibles : *all* - compter tous les changements (par défaut) ; *dec* - compter les diminutions ; *inc* - compter les augmentations

Commentaires :

- Pour les types de valeurs non numériques, le paramètre *mode* est ignoré.
- Dans les déclencheurs, utile pour détecter des changements d'état fréquents (indiquant une instabilité).

Exemples :

```
changecount(/host/icmpping,10m)>5 #ping status has changed more than 5 times in 10 minutes
changecount(/host/vfs.file.contents["/sys/class/net/eth0/operstate"],1h)>5 #operational state of eth0 has
changecount(/host/proc.num[httpd],15m)>10 #the number of httpd processes has changed more than 10 times in
changecount(/host/key,#10,"inc") #the number of value increases (relative to the adjacent value) among the
changecount(/host/key,24h,"dec") #the number of value decreases (relative to the adjacent value) for the l
```

count(/host/key,(sec|#num)<:time shift>,<operator>,<pattern>)

Le nombre de valeurs dans la période d'évaluation définie.
 Types de valeurs pris en charge: *Float*, *Integer*, *String*, *Text*, *Log*.

Paramètres:

- Voir paramètres communs
- **operator** (doit être entre guillemets doubles) `operators` pris en charge:
eq - égal (par défaut pour integer, float)
ne - différent
gt - supérieur à
ge - supérieur ou égal à
lt - inférieur à
le - inférieur ou égal à
like (par défaut pour string, text, log) - correspond si le motif est contenu (sensible à la casse)
bitand - ET bit

à bit
regexp - correspondance sensible à la casse avec l'expression régulière donnée dans pattern
iregexp - correspondance insensible à la casse avec l'expression régulière donnée dans pattern

- **pattern** - le motif requis (les arguments de type chaîne doivent être entre guillemets doubles)

Commentaires:

- Les éléments Float correspondent avec une précision de 2.22e-16.
- *like* n'est pas pris en charge comme opérateur pour les valeurs integer.
- *like* et *bitand* ne sont pas pris en charge comme opérateurs pour les valeurs float.
- Pour les valeurs string, text et log, seuls les opérateurs *eq*, *ne*, *like*, *regexp* et *iregexp* sont pris en charge.
- Avec *bitand* comme opérateur, le quatrième paramètre pattern peut être spécifié sous forme de deux nombres, séparés par '/': **number_to_compare_with/mask**. count() calcule le "ET bit à bit" entre la valeur et le *mask* et compare le résultat à *number_to_compare_with*. Si le résultat du "ET bit à bit" est égal à *number_to_compare_with*, la valeur est comptée.
Si *number_to_compare_with* et *mask* sont égaux, seul le *mask* doit être spécifié (sans '/').
- Avec *regexp* ou *iregexp* comme opérateur, le quatrième paramètre pattern peut être une expression régulière ordinaire ou globale (commençant par '@'). Dans le cas des expressions régulières globales, la sensibilité à la casse est héritée des paramètres de l'expression régulière globale. Aux fins de la correspondance regexp, les valeurs float seront toujours représentées avec 4 chiffres décimaux après '.'. Notez également que pour les grands nombres, la différence entre la représentation décimale (stockée dans la base de données) et binaire (utilisée par le serveur Zabbix) peut affecter le 4e chiffre décimal.

Exemples:

```
count(/host/icmpping,30m,,"0")>5 #le ping a échoué plus de 5 fois en 30 minutes
count(/host/key,10m,"like","error") #le nombre de valeurs des 10 dernières minutes jusqu'à maintenant qui
count(/host/key,10m,,12) #le nombre de valeurs des 10 dernières minutes jusqu'à maintenant qui sont égales
count(/host/key,10m,"gt",12) #le nombre de valeurs des 10 dernières minutes jusqu'à maintenant qui sont su
count(/host/key,#10,"gt",12) #le nombre de valeurs parmi les 10 dernières valeurs jusqu'à maintenant qui s
count(/host/key,10m:now-1d,"gt",12) #le nombre de valeurs entre 24 heures et 10 minutes, et 24 heures avan
count(/host/key,10m,"bitand","6/7") #le nombre de valeurs des 10 dernières minutes jusqu'à maintenant ayan
count(/host/key,10m:now-1d) #le nombre de valeurs entre 24 heures et 10 minutes, et 24 heures avant mainte
```

countunique(/host/key,(sec|#num)<:time shift>,<operator>,<pattern>)

Le nombre de valeurs uniques dans la période d'évaluation définie.
 Types de valeurs pris en charge: *Float*, *Integer*, *String*, *Text*, *Log*.

Paramètres:

- Voir paramètres communs
- **operator** (doit être entre guillemets doubles). Opérateurs pris en charge:
eq - égal (par défaut pour integer, float)
ne - différent
gt - supérieur
ge - supérieur ou égal
lt - inférieur
le - inférieur ou égal
like (par défaut pour string, text, log) - correspond si le motif est contenu (sensible à la casse)
bitand - ET bit à bit
regexp - correspondance sensible à la casse avec l'expression régulière donnée dans pattern
iregexp - correspondance insensible à la casse avec l'expression régulière donnée dans pattern
- **pattern** - le motif requis (les arguments de type chaîne doivent être entre guillemets doubles)

Commentaires:

- Les éléments Float correspondent avec une précision de 2.22e-16.
- *like* n'est pas pris en charge comme opérateur pour les valeurs integer.
- *like* et *bitand* ne sont pas pris en charge comme opérateurs pour les valeurs float.
- Pour les valeurs string, text et log, seuls les opérateurs *eq*, *ne*, *like*, *regexp* et *iregexp* sont pris en charge.
- Avec *bitand* comme opérateur, le quatrième paramètre pattern peut être spécifié sous forme de deux nombres, séparés par '/': **number_to_compare_with/mask**. countunique() calcule le "ET bit à bit" de la valeur et du *mask* et compare le résultat à *number_to_compare_with*. Si le résultat du "ET bit à bit" est égal à *number_to_compare_with*, la valeur est comptée.
Si *number_to_compare_with* et *mask* sont égaux, seul le *mask* doit être spécifié (sans '/').
- Avec *regexp* ou *iregexp* comme opérateur, le quatrième paramètre pattern peut être une expression régulière ordinaire ou globale (commençant par '@'). Dans le cas des expressions régulières globales, la sensibilité à la casse est héritée des paramètres de l'expression régulière globale. Pour la correspondance regexp, les valeurs float seront toujours représentées avec 4 chiffres décimaux après '.'. Notez également que pour les grands nombres, la différence entre la représentation décimale (stockée dans la base de données) et binaire (utilisée par le serveur Zabbix) peut affecter le 4e chiffre décimal.

Exemples:

```
countunique(/host/key,10m) #le nombre de valeurs uniques pour les 10 dernières minutes jusqu'à maintenant
countunique(/host/key,10m,"like","error") #le nombre de valeurs uniques pour les 10 dernières minutes jusq
countunique(/host/key,10m,,12) #le nombre de valeurs uniques pour les 10 dernières minutes jusqu'à mainten
countunique(/host/key,10m,"gt",12) #le nombre de valeurs uniques pour les 10 dernières minutes jusqu'à mai
```

```
countunique(/host/key,#10,"gt",12) #le nombre de valeurs uniques parmi les 10 dernières valeurs jusqu'à ma
countunique(/host/key,10m:now-1d,"gt",12) #le nombre de valeurs uniques entre 24 heures et 10 minutes, il
countunique(/host/key,10m,"bitand","6/7") #le nombre de valeurs uniques pour les 10 dernières minutes jusq
countunique(/host/key,10m:now-1d) #le nombre de valeurs uniques entre 24 heures et 10 minutes, il y a 24 h
```

find(/host/key,(sec|#num)<:time shift>,<operator>,<pattern>)

Trouve une valeur correspondant à un critère dans la période d'évaluation définie.
 Types de valeurs pris en charge: *Float*, *Integer*, *String*, *Text*, *Log*.
 Retourne: 1 - trouvé; 0 - sinon.

Paramètres:

- Voir paramètres communs
- **sec** ou **#num** (facultatif) - prend par défaut la valeur la plus récente si non spécifié
- **operator** (doit être entre guillemets doubles) `operators` pris en charge:
eq - égal à (par défaut pour integer, float)
ne - différent de
gt - supérieur à
ge - supérieur ou égal à
lt - inférieur à
le - inférieur ou égal à
like (par défaut pour string, text, log) - correspond si la chaîne donnée dans `pattern` est contenue (sensible à la casse)
bitand - ET bit à bit
regexp - correspondance sensible à la casse avec l'expression régulière donnée dans `pattern`
iregexp - correspondance insensible à la casse avec l'expression régulière donnée dans `pattern`
- **pattern** - le motif requis (les arguments de type chaîne doivent être entre guillemets doubles); expression régulière Perl Compatible Regular Expression (PCRE) si `operator` est *regexp*, *iregexp*

Commentaires:

- Si plus d'une valeur est traitée, '1' est renvoyé s'il existe au moins une valeur correspondante.
- *like* n'est pas pris en charge comme opérateur pour les valeurs integer.
- *like* et *bitand* ne sont pas pris en charge comme opérateurs pour les valeurs float.
- Pour les valeurs string, text et log, seuls les opérateurs *eq*, *ne*, *like*, *regexp* et *iregexp* sont pris en charge.
- Avec *regexp* ou *iregexp* comme opérateur, le quatrième paramètre `pattern` peut être une expression régulière ordinaire ou globale (commençant par '@'). Dans le cas des expressions régulières globales, la sensibilité à la casse est héritée des paramètres de l'expression régulière globale.

Exemples:

```
find(/host/key,10m,"like","error") #trouve une valeur qui contient 'error' au cours des 10 dernières minut
find(/host/agent.version,,"like","beta")=1 #la version de l'agent Zabbix est bêta, elle doit être mise à n
find(/host/log[/var/log/nginx/access.log],,"regexp"," 500 ")=1 #une erreur interne du serveur web a été tr
```

first(/host/key,sec<:time shift>)

La première valeur (la plus ancienne) dans la période d'évaluation définie.
 Types de valeurs pris en charge : *Float*, *Integer*, *String*, *Text*, *Log*.

Paramètres :

- Voir paramètres communs

Voir aussi last().

Exemple :

```
first(/host/key,1h) #récupère la valeur la plus ancienne de la dernière heure jusqu'à maintenant
```

firstclock(/host/key,sec<:time shift>)

L'horodatage de la valeur la plus ancienne dans la période d'évaluation définie.
 Types de valeurs pris en charge : *Float*, *Integer*, *String*, *Text*, *Log*.

Paramètres :

- Voir paramètres communs.

La fonction échoue avec une erreur si aucune donnée n'a été collectée pendant la période donnée.

Voir aussi lastclock().

Exemples :

```
firstclock(/host/key,1h) #récupère l'horodatage de la valeur la plus ancienne au cours de la dernière heure
firstclock(/host/key,1h:now-24h) #récupère l'horodatage de la valeur la plus ancienne au cours de la dernie
```

fuzzytime(/host/key,sec)

Vérifie de combien l'heure de l'agent passif diffère de l'heure du serveur/proxy Zabbix.
 Types de valeurs pris en charge: *Float*, *Integer*.
 Retourne: 1 - la différence entre la valeur de l'élément passif (en tant qu'horodatage) et l'horodatage du serveur/proxy Zabbix (l'heure de collecte de la valeur) est inférieure ou égale à *sec* secondes; 0 - sinon.

Paramètres:

- Voir paramètres communs.

Commentaires:

- Utilisée généralement avec l'élément 'system.localtime' pour vérifier que l'heure locale est synchronisée avec l'heure locale du serveur Zabbix. *Note* que 'system.localtime' doit être configuré comme une vérification passive pour l'agent Zabbix; sur Zabbix agent 2, il peut être configuré comme une vérification active.
- Peut également être utilisée avec la clé `vfs.file.time[/path/file,modify]` pour vérifier que le fichier n'a pas été mis à jour depuis longtemps.
- Cette fonction n'est pas recommandée pour une utilisation dans des expressions de déclencheur complexes (impliquant plusieurs éléments), car elle peut provoquer des résultats inattendus (la différence de temps sera mesurée avec la métrique la plus récente), par exemple dans `fuzzytime(/Host/system.localtime,60s)=0 or last(/Host/trap)<>0`.

Exemples:

```
fuzzytime(/host/system.localtime,5m)=0 #l'heure locale du client diffère de l'heure du serveur/proxy Zabbi
fuzzytime(/host/system.localtime,5m)=0 and nodata(/host/system.localtime,10m)=0 #l'heure locale du client
```

jsonpath(/host/key,(sec|#num)<:time shift>,pattern)

Renvoie un tableau de résultats de requêtes JSONPath.
 Types de valeurs pris en charge : *String*, *Text*, *Log*.

Paramètres :

- Voir les paramètres communs
- **pattern** - le modèle requis (les arguments de type chaîne doivent être placés entre guillemets doubles)

Commentaires :

- Cette fonction est utilisée avec les fonctions d'agrégation.
- Une requête JSONPath est exécutée sur chaque valeur d'historique sélectionnée ; chaque résultat de requête réussi et non vide est ajouté au tableau. Si le résultat d'une requête pour une seule valeur d'historique contient plusieurs correspondances, chaque correspondance est ajoutée séparément au tableau.
- La fonction échoue avec une erreur si :
  - Le format de `pattern` n'est pas valide.
  - L'une des valeurs d'historique contient un JSON non valide ou si la requête JSONPath échoue pour cette valeur.
- La fonction renvoie un tableau vide si :
  - Aucune valeur d'historique n'a été sélectionnée.
  - Toutes les valeurs d'historique sélectionnées contiennent un JSON non valide.

Exemple :

```
max(jsonpath(/host/http.response,5m,"$.response_time")) > 500
```

last(/host/key,<#num<:time shift»)

La valeur la plus récente.
 Types de valeurs pris en charge : *Float*, *Integer*, *String*, *Text*, *Log*.

Paramètres :

- Voir paramètres communs
- **#num** (facultatif) - la Nième valeur la plus récente

Commentaires :

- Notez qu'une période de temps avec dièse (#N) fonctionne ici différemment de nombreuses autres fonctions. Par exemple : `last(/host/key)` est toujours égal à `last(/host/key,#1)` ; `last(/host/key,#3)` - la troisième valeur la plus récente (*et non* les trois dernières valeurs).
- Zabbix ne garantit pas l'ordre exact des valeurs si plus de deux valeurs existent dans l'historique au cours d'une même seconde.
- Voir aussi first().

Exemples :

```
last(/host/key) #retrieve the last value
last(/host/key,#2) #retrieve the previous value
last(/host/key,#1)<>last(/host/key,#2) #last two values differ
```

lastclock(/host/key,<#num<:time shift»)

L'horodatage de la Nième valeur la plus récente dans la période d'évaluation définie.
 Types de valeurs pris en charge : *Float*, *Integer*, *String*, *Text*, *Log*.

Paramètres :

- Voir paramètres communs;
- **#num** (facultatif) - la Nième valeur la plus récente.

La fonction échoue avec une erreur si aucune donnée n'a été collectée dans la période donnée ou si aucune Nième valeur n'a été collectée.

Voir aussi firstclock().

Exemples :

```
lastclock(/host/key) #récupérer l'horodatage de la dernière valeur
lastclock(/host/key,#2) #récupérer l'horodatage de la 2e dernière valeur
now()-lastclock(/host/heartbeat.item)>300 #plus de 5 minutes se sont écoulées depuis la mise à jour de l'é
lastclock(/host/system.cpu.load[all,avg1])-lastclock(/host/system.cpu.load[all,avg1],#2)>300 #l'intervalle
((now()-lastclock(/host/system.cpu.load[all,avg1])<120 and last(/host/system.cpu.load[all,avg1])>5)) #la v
```

logeventid(/host/key,<#num<:time shift»,<pattern>)

Vérifie si l'ID d'événement de la dernière entrée de journal correspond à une expression régulière.
 Types de valeurs pris en charge: *Log*.
 Retourne: 0 - ne correspond pas; 1 - correspond.

Paramètres:

- Voir paramètres communs
- **#num** (facultatif) - la Nième valeur la plus récente
- **pattern** (facultatif) - l'expression régulière décrivant le motif requis, au format Perl Compatible Regular Expression (PCRE) (les arguments de chaîne doivent être entre guillemets doubles)

Exemples:

```
logeventid(/host/eventlog[Security],,"^4625$")=1 #une entrée de journal avec l'ID correspondant à "4625" (
logeventid(/host/eventlog[System],,"^(6008|41)$")=1 #une entrée de journal avec l'ID correspondant à "6008
```

logseverity(/host/key,<#num<:time shift»)

Gravité du journal de la dernière entrée de journal.
 Types de valeurs pris en charge : *Log*.
 Renvoie : 0 - gravité par défaut ; N - gravité (integer, utile pour les journaux d'événements Windows : 1 - Information, 2 - Warning, 4 - Error, 7 - Failure Audit, 8 - Success Audit, 9 - Critical, 10 - Verbose).

Paramètres :

- Voir paramètres communs
- **#num** (facultatif) - la Nième valeur la plus récente

Zabbix récupère la gravité du journal à partir du champ **Information** du journal d'événements Windows.

Exemples :

```
logseverity(/host/log[/var/log/syslog],10m)>3 #a log entry with severity above "3" found
logseverity(/host/eventlog[System],10m)=4 #a log entry with severity equaling "Error" found
```

logsource(/host/key,<#num<:time shift»,<pattern>)

Vérifie si la source du journal de la dernière entrée de journal correspond à une expression régulière.
 Types de valeurs pris en charge: *Log*.
 Retourne: 0 - ne correspond pas; 1 - correspond.

Paramètres:

- Voir paramètres communs
- **#num** (facultatif) - la Nième valeur la plus récente
- **pattern** (facultatif) - l'expression régulière décrivant le motif requis, au format Perl Compatible Regular Expression (PCRE) (les arguments de chaîne doivent être entre guillemets doubles)

Normalement utilisé pour les journaux d'événements Windows.

Exemples:

```
logsource(/host/eventlog[Application],,"MSSQLSERVER")=1 #une entrée de journal avec une source correspond
logsource(/host/eventlog[System],,"Service Control Manager")=1 #une entrée de journal avec une source corr
logsource(/host/eventlog[System],,"Service Control Manager")=1 and logeventid(/host/eventlog[System],,"^70
```

logtimestamp(/host/key,<#num<:time shift»)

L'horodatage du message de journal de la Nième valeur la plus récente d'un élément de journal.
 Types de valeurs pris en charge : *Log*.

Paramètres :

- Voir les paramètres communs ;
- **#num** (facultatif) - la Nième valeur la plus récente.

Commentaires :

- Le calcul du décalage temporel est basé sur l'horloge de la valeur de l'élément, et non sur l'horodatage du message de journal ;
- La fonction échoue avec une erreur si :
  - un élément de type non-journal est reçu ;
  - aucune donnée n'a été collectée pendant la période donnée ;
  - aucune Nième valeur n'a été collectée ;
  - le message de journal ne contient pas de valeur d'horodatage.

Exemples :

```
logtimestamp(/host/key) #retrieve the timestamp of the latest log message
logtimestamp(/host/key,#2) #retrieve the timestamp of the 2nd latest log message
logtimestamp(/host/key,#3:now-1d) #retrieve the timestamp of the 3rd latest log message a day ago
```

monodec(/host/key,(sec|#num)<:time shift>,<mode>)

Vérifie s'il y a eu une diminution monotone des valeurs.
 Types de valeurs pris en charge : *Integer*.
 Renvoie : 1 - si tous les éléments de la période diminuent continuellement ; 0 - sinon.

Paramètres :

- Voir paramètres communs
- **mode** (doit être entre guillemets doubles) - *weak* (chaque valeur est inférieure ou égale à la précédente ; valeur par défaut) ou *strict* (chaque valeur a diminué)

Exemples :

```
monodec(/host/system.swap.size[all,free],60s) + monodec(/host2/system.swap.size[all,free],60s) + monodec(/
monodec(/host/proc.num[nginx],10m,"strict")=1 #le nombre de processus nginx a diminué de façon monotone pe
```

monoinc(/host/key,(sec|#num)<:time shift>,<mode>)

Vérifie s'il y a eu une augmentation monotone des valeurs.
 Types de valeurs pris en charge : *Integer*.
 Renvoie : 1 - si tous les éléments de la période augmentent continuellement ; 0 - sinon.

Paramètres :

- Voir paramètres communs
- **mode** (doit être entre guillemets doubles) - *weak* (chaque valeur est supérieure ou égale à la précédente ; valeur par défaut) ou *strict* (chaque valeur a augmenté)

Exemples :

```
monoinc(/host/system.localtime,#3,"strict")=0 #l'heure locale du système n'a pas augmenté de manière cohér
monoinc(/host/vfs.dir.count[/mnt/data/logs],24h,"weak")=0 #déclencher si le nombre de fichiers a stagné pe
```

nodata(/host/key,sec,<mode>)

Vérifie qu'aucune donnée n'a été reçue.
 Types de valeurs pris en charge : *Integer*, *Float*, *Character*, *Text*, *Log*.
 Renvoie : 1 - si aucune donnée n'a été reçue pendant la période définie ; 0 - dans le cas contraire.

Paramètres :

- Voir les paramètres communs
- **sec** - la période ne doit pas être inférieure à 30 secondes, car le processus history syncer ne calcule cette fonction que toutes les 30 secondes ; `nodata(/host/key,0)` est interdit

- **mode** - si défini sur *strict* (entre guillemets doubles), cette fonction ne tiendra pas compte de la disponibilité du proxy (voir les commentaires pour plus de détails)

Commentaires :

- Par défaut, les déclencheurs 'nodata' surveillés par un proxy tiennent compte de la disponibilité du proxy : si le proxy devient indisponible, les déclencheurs 'nodata' ne se déclencheront pas immédiatement après le rétablissement de la connexion, mais ignoreront les données correspondant à la période de retard. Notez que pour les proxies passifs, la suppression est activée si la connexion est rétablie plus de 15 secondes et au moins 2 secondes plus tard. Pour les proxies actifs, la suppression est activée si la connexion est rétablie plus de 15 secondes plus tard. Pour désactiver la prise en compte de la disponibilité du proxy, utilisez le troisième paramètre, par exemple : `nodata(/host/key,5m,"strict")` ; dans ce cas, la fonction se déclenchera dès que la période d'évaluation (cinq minutes) sans données sera écoulée.
- Cette fonction affichera une erreur si, au cours de la période définie par le premier paramètre :
- aucune donnée n'est disponible et le serveur Zabbix a été redémarré
- aucune donnée n'est disponible et la maintenance est terminée
- aucune donnée n'est disponible et l'élément a été ajouté ou réactivé.
- Les erreurs sont affichées dans la colonne *Info* de la liste des déclencheurs.
- Cette fonction peut ne pas fonctionner correctement s'il existe des différences horaires entre le serveur Zabbix, le proxy et l'agent. Voir également : Exigence de synchronisation de l'heure.

Exemple :

```
nodata(/host/agent.ping,5m)=1 #trigger if no data has been received from Zabbix agent for 5 minutes
```

percentile(/host/key,(sec|#num)<:time shift>,percentage)

Le percentile P d'une période, où P (pourcentage) est spécifié par le troisième paramètre.
 Types de valeurs pris en charge : *Float*, *Integer*.

Paramètres :

- Voir paramètres communs
- **percentage** - un nombre à virgule flottante compris entre 0 et 100 (inclus), avec jusqu'à 4 chiffres après la virgule

Exemples :

```
percentile(/host/net.if.in[eth0,bytes],1h,95)>1000000 #le 95e percentile du trafic réseau entrant (octets/
percentile(/host/system.cpu.util,5m,95)>80 #le 95e percentile du pourcentage d'utilisation CPU en temps ut
percentile(/host/icmppingsec[192.168.0.2],15m,95)>0.15 #la plupart des mesures de latence sont inférieures
percentile(/host/net.if.in[eth0,bytes],1h,50) #calcule le 50e percentile (la valeur médiane) du trafic rés
(percentile(/host/net.if.in[eth0,bytes],1h,50)+percentile(/host/net.if.in[eth0,bytes],1h,51))/2 #calcule l
```

rate(/host/key,sec<:time shift>)

L'augmentation moyenne par seconde d'un compteur croissant de manière monotone sur la période définie.
 Types de valeurs pris en charge : *Float*, *Integer*.

Paramètres :

- Voir paramètres communs

Correspond fonctionnellement à « rate » de PromQL.

Exemples :

```
rate(/host/key,30s) #si l'augmentation monotone sur 30 secondes est de 20, cette fonction renvoie 0.67.
rate(/host/net.if.in[eth0,bytes],5m)>500000 #le débit de trafic entrant de l'interface sur eth0 a dépassé
rate(/host/app.requests.count,1m)>100 #le compteur de requêtes a augmenté de plus de 100 requêtes par seco
```

xmlxpath(/host/key,(sec|#num)<:time shift>,pattern)

Renvoie un tableau de résultats de requêtes XPath XML.
 Types de valeurs pris en charge : *String*, *Text*, *Log*.

Paramètres :

- Voir les paramètres communs
- **pattern** - le modèle requis (les arguments de type chaîne doivent être placés entre guillemets doubles)

Commentaires :

- Cette fonction est utilisée avec les fonctions d'agrégation.
- Une requête XPath XML est exécutée sur chaque valeur d'historique sélectionnée ; chaque résultat non vide obtenu avec succès est ajouté au tableau. Si le résultat d'une requête pour une seule valeur d'historique contient plusieurs correspondances, chaque correspondance est ajoutée séparément au tableau.
- La fonction échoue avec une erreur si :

- Le format du `pattern` n'est pas valide.
- Zabbix a été compilé sans prise en charge de XML.
- L'une des valeurs d'historique contient un XML non valide ou si la requête XPath échoue pour cette valeur.

- La fonction renvoie un tableau vide si :
  - Aucune valeur d'historique n'a été sélectionnée.
  - Toutes les valeurs d'historique sélectionnées contiennent un XML non valide.

Exemple :

```
max(xmlxpath(/host/http.response,5m,"/response/response_time")) > 500
```

Voir toutes les fonctions prises en charge.

Fonctions de tendance

Les fonctions de tendance, contrairement aux fonctions d'historique, utilisent des données de tendance pour les calculs.

Les tendances stockent des valeurs agrégées horaires. Les fonctions de tendance utilisent ces moyennes horaires et sont donc utiles pour l'analyse à long terme.

Les résultats des fonctions de tendance sont mis en cache, de sorte que plusieurs appels à la même fonction avec les mêmes paramètres ne récupèrent les informations de la base de données qu'une seule fois. Le cache des fonctions de tendance est contrôlé par le paramètre serveur TrendFunctionCacheSize.

Utilisez `zabbix[tcache,cache,paccessed]` pour estimer quelle part du cache est réellement utilisée. Une valeur faible de `paccessed`, associée à de faibles valeurs de `pitems` et `pmisses`, signifie généralement que la taille du cache peut être réduite.

Les déclencheurs qui référencent uniquement des fonctions de tendance sont évalués une fois par la plus petite période de temps de l'expression. Par exemple, un déclencheur comme

```
trendavg(/host/key,1d:now/d) > 1 or trendavg(/host/key2,1w:now/w) > 2
```

sera évalué une fois par jour. Si le déclencheur contient à la fois des fonctions de tendance et d'historique (ou date et heure et/ou nodata()), il est calculé conformément aux principes habituels.

Toutes les fonctions répertoriées ici sont prises en charge dans :

- Expressions de déclencheur
- Éléments calculés
- Macros d'expression

Les fonctions sont listées sans informations supplémentaires. Cliquez sur la fonction pour voir les détails complets.

Function	Description
baselinedev	Retourne le nombre d'écarts (selon l'algorithme stddevpop) entre la dernière période de données et les mêmes périodes de données lors des saisons précédentes.
baselinewma	Calcule la ligne de base en moyennant les données de la même plage temporelle sur plusieurs périodes de temps égales (”saisons”) à l'aide de l'algorithme de moyenne mobile pondérée.
trendavg	La moyenne des valeurs de tendance sur la période de temps définie.
trendcount	Le nombre de valeurs d'historique récupérées avec succès et utilisées pour calculer la valeur de tendance sur la période de temps définie.
trendmax	Le maximum des valeurs de tendance sur la période de temps définie.
trendmin	Le minimum des valeurs de tendance sur la période de temps définie.
trendstl	Retourne le taux d'anomalies pendant la période de détection - une valeur décimale comprise entre 0 et 1, égale à `((le nombre de valeurs d'anomalie)/(nombre total de valeurs))`.
trendsum	La somme des valeurs de tendance sur la période de temps définie.

Paramètres communs

- `/host/key` est un premier paramètre commun obligatoire
- `time period:time shift` est un deuxième paramètre commun, où :
  - **time period** - la période de temps (minimum « 1h »), définie comme <N><time unit> où `N` - le nombre d'unités de temps, `time unit` - h (heure), d (jour), w (semaine), M (mois) ou y (année).
  - **time shift** - le décalage de période (voir les exemples de fonction)

Détails des fonctions

Quelques remarques générales concernant les paramètres des fonctions :

- Les paramètres des fonctions sont séparés par une virgule
- Les paramètres de fonction facultatifs (ou parties de paramètres) sont indiqués par < >
- Les paramètres spécifiques à chaque fonction sont décrits avec chaque fonction
- Les paramètres `/host/key` et `time period:time shift` ne doivent jamais être placés entre guillemets

baselinedev(/host/key,data period:time shift,season unit,num seasons)

Renvoie le nombre d'écarts (selon l'algorithme stddevpop) entre la dernière période de données et les mêmes périodes de données des saisons précédentes.

Paramètres :

- Voir les paramètres communs
- **data period** - la période de collecte des données au sein d'une saison, définie comme <N><time unit> où :
`N` - le nombre d'unités de temps
`time unit` - h (heure), d (jour), w (semaine), `M` (mois) ou y (année), doit être égal ou inférieur à la saison
- **season unit** - l'unité calendaire qui définit une saison (h, d, w, M, y) ; ne peut pas être plus petite que data period
- **num seasons** - le nombre de saisons à évaluer

Exemples :

```
baselinedev(/host/key,1d:now/d,"M",6) #calcule le nombre d'écarts-types (population) entre le jour précéde
baselinedev(/host/key,1h:now/h,"d",10)  #calcule le nombre d'écarts-types de population entre l'heure préc
```

baselinewma(/host/key,data period:time shift,season unit,num seasons)

Calcule la ligne de base en moyennant les données de la même période temporelle sur plusieurs périodes de temps égales (« saisons ») à l'aide de l'algorithme de moyenne mobile pondérée.

Paramètres :

- Voir paramètres communs
- **data period** - la période de collecte des données au sein d'une saison, définie comme <N><time unit> où :
`N` - le nombre d'unités de temps
`time unit` - h (heure), d (jour), w (semaine), `M` (mois) ou y (année), doit être égal ou inférieur à la saison
Time shift - le décalage temporel, définit la fin de la période de collecte des données en saisons (voir les exemples)
- **season unit** - l'unité calendaire qui définit une saison (h, d, w, M, y) ; ne peut pas être inférieure à data period
- **num seasons** - le nombre de saisons à évaluer

Exemples :

```
baselinewma(/host/key,1h:now/h,"d",3) #calcul de la ligne de base à partir de la même heure de la journée
baselinewma(/host/key,2h:now/h,"d",3) #calcul de la ligne de base à partir de la même plage de deux heures
baselinewma(/host/key,1d:now/d,"M",4) #calcul de la ligne de base à partir du même jour du mois que « hier
```

trendavg(/host/key,time period:time shift)

La moyenne des valeurs de tendance dans la période définie.

Paramètres :

- Voir paramètres communs

Exemples :

```
trendavg(/host/key,1h:now/h) #la moyenne pour l'heure précédente (par ex. 12:00-13:00)
trendavg(/host/key,1h:now/h-1h) #la moyenne pour il y a deux heures (11:00-12:00)
trendavg(/host/key,1h:now/h-2h) #la moyenne pour il y a trois heures (10:00-11:00)
trendavg(/host/key,1M:now/M-1y) #la moyenne pour le mois précédent il y a un an
```

trendcount(/host/key,time period:time shift)

Le nombre de valeurs d'historique récupérées avec succès utilisées pour calculer la valeur de tendance dans la période définie.

Paramètres :

- Voir paramètres communs

Exemples :

```
trendcount(/host/key,1h:now/h) #le nombre de valeurs pour l'heure précédente (p. ex. 12:00-13:00)
trendcount(/host/key,1h:now/h-1h) #le nombre de valeurs pour il y a deux heures (11:00-12:00)
trendcount(/host/key,1h:now/h-2h) #le nombre de valeurs pour il y a trois heures (10:00-11:00)
trendcount(/host/key,1M:now/M-1y) #le nombre de valeurs pour le mois précédent il y a un an
```

trendmax(/host/key,time period:time shift)

Le maximum des valeurs de tendance dans la période définie.

Paramètres :

- Voir paramètres communs

Exemples :

```
trendmax(/host/key,1h:now/h) #le maximum pour l'heure précédente (par ex. 12:00-13:00)
trendmax(/host/key,1h:now/h) - trendmin(/host/key,1h:now/h) → calculer la différence entre les valeurs max
trendmax(/host/key,1h:now/h-1h) #le maximum pour il y a deux heures (11:00-12:00)
trendmax(/host/key,1h:now/h-2h) #le maximum pour il y a trois heures (10:00-11:00)
trendmax(/host/key,1M:now/M-1y) #le maximum pour le mois précédent il y a un an
```

trendmin(/host/key,time period:time shift)

Le minimum des valeurs de tendance dans la période définie.

Paramètres :

- Voir paramètres communs

Exemples :

```
trendmin(/host/key,1h:now/h) #le minimum pour l'heure précédente (p. ex. 12:00-13:00)
trendmax(/host/key,1h:now/h) - trendmin(/host/key,1h:now/h) → calculer la différence entre les valeurs max
trendmin(/host/key,1h:now/h-1h) #le minimum pour il y a deux heures (11:00-12:00)
trendmin(/host/key,1h:now/h-2h) #le minimum pour il y a trois heures (10:00-11:00)
trendmin(/host/key,1M:now/M-1y) #le minimum pour le mois précédent il y a un an
```

trendstl(/host/key,eval period:time shift,detection period,season,<deviations>,<devalg>,<s window>)

Renvoie le taux d'anomalies pendant la période de détection - une valeur décimale comprise entre 0 et 1, soit `((le nombre de valeurs anormales)/(le nombre total de valeurs))`.

Paramètres :

- Voir paramètres communs
- **eval period** - la période de temps qui doit être décomposée (minimum '1h'), définie comme <N><time unit> où
`N` - le nombre d'unités de temps
`time unit` - h (heure), d (jour), w (semaine), M (mois) ou y (année)
- **detection period** - la période de temps avant la fin de eval period pour laquelle les anomalies sont calculées (minimum '1h', ne peut pas être plus longue que eval period), définie comme <N><time unit> où
`N` - le nombre d'unités de temps
`time unit` - h (heure), d (jour), w (semaine)
- **season** - la plus courte période de temps pour laquelle un motif répétitif ("saison") est attendu (minimum '2h', ne peut pas être plus longue que eval period, le nombre d'entrées dans eval period doit être supérieur à deux fois la fréquence résultante (season/h)), définie comme <N><time unit> où
`N` - le nombre d'unités de temps
`time unit` - h (heure), d (jour), w (semaine)
- **deviations** - le nombre d'écarts (calculés par devalg) à considérer comme une anomalie (peut être décimal), (doit être supérieur ou égal à 1, la valeur par défaut est 3)
- **devalg** (doit être entre guillemets doubles) - l'algorithme d'écart, peut être *stddevpop*, *stddevsamp* ou *mad* (par défaut)
- **s window** - l'étendue (en retards) de la fenêtre loess pour l'extraction saisonnière (la valeur par défaut est 10 * nombre d'entrées dans eval period + 1)

Exemples :

```
trendstl(/host/key,100h:now/h,10h,2h) #analyse les 100 dernières heures de données de tendances, trouve le
trendstl(/host/key,100h:now/h-10h,100h,2h,2.1,"mad") #analyse la période de 100 heures de données de tenda
trendstl(/host/key,100d:now/d-1d,10d,1d,4,,10) #analyse 100 jours de données de tendances jusqu'à il y a u
trendstl(/host/key,1M:now/M-1y,1d,2h,,"stddevsamp") #analyse le mois précédent il y a un an, trouve le tau
```

trendsum(/host/key,time period:time shift)

La somme des valeurs de tendance dans la période définie.

Paramètres :

- Voir paramètres communs

Exemples :

```
trendsum(/host/key,1h:now/h) #la somme pour l'heure précédente (par ex. 12:00-13:00)
trendsum(/host/key,1h:now/h-1h) #la somme pour il y a deux heures (11:00-12:00)
trendsum(/host/key,1h:now/h-2h) #la somme pour il y a trois heures (10:00-11:00)
trendsum(/host/key,1M:now/M-1y) #la somme pour le mois précédent il y a un an
```

Voir toutes les fonctions prises en charge.

Fonctions mathématiques

Toutes les fonctions répertoriées ici sont prises en charge dans :

- Expressions de déclencheur
- Formules d'élément calculé
- Macros d'expression

Les fonctions mathématiques sont prises en charge avec les types de valeurs flottant et entier, sauf indication contraire.

Les fonctions sont répertoriées sans informations supplémentaires. Cliquez sur la fonction pour afficher les détails complets.

Function	Description
abs	La valeur absolue d'une valeur.
acos	L'arccosinus d'une valeur, en tant qu'angle exprimé en radians.
asin	L'arcsinus d'une valeur, en tant qu'angle exprimé en radians.
atan	L'arctangente d'une valeur, en tant qu'angle exprimé en radians.
atan2	L'arctangente des coordonnées de l'ordonnée (valeur) et de l'abscisse spécifiées en tant qu'angle, exprimé en radians.
avg	La valeur moyenne des valeurs de l'élément référencé.
cbrt	La racine cubique d'une valeur.
ceil	Arrondit la valeur à l'entier supérieur le plus proche ou égal.
cos	Le cosinus d'une valeur, où la valeur est un angle exprimé en radians.
cosh	Le cosinus hyperbolique d'une valeur.
cot	La cotangente d'une valeur, où la valeur est un angle exprimé en radians.
degrees	Convertit une valeur de radians en degrés.
e	Le nombre d'Euler (2.718281828459045).
exp	Le nombre d'Euler élevé à la puissance d'une valeur.
expm1	Le nombre d'Euler élevé à la puissance d'une valeur moins 1.
floor	Arrondit la valeur à l'entier inférieur le plus proche ou égal.
log	Le logarithme naturel.
log10	Le logarithme décimal.
max	La valeur la plus élevée des valeurs de l'élément référencé.
min	La valeur la plus faible des valeurs de l'élément référencé.
mod	Le reste de la division.
pi	La constante Pi (3.14159265358979).
power	La puissance d'une valeur.
radians	Convertit une valeur de degrés en radians.
rand	Retourne une valeur entière aléatoire.
round	Arrondit la valeur à un nombre de décimales.
signum	Retourne '-1' si une valeur est négative, '0' si une valeur est nulle, '1' si une valeur est positive.
sin	Le sinus d'une valeur, où la valeur est un angle exprimé en radians.
sinh	Le sinus hyperbolique d'une valeur, où la valeur est un angle exprimé en radians.
sqrt	La racine carrée d'une valeur.
sum	La somme des valeurs de l'élément référencé.
tan	La tangente d'une valeur.
truncate	Tronque la valeur à un nombre de décimales.

Détails des fonctions

Quelques remarques générales sur les paramètres de fonction :

- Les paramètres de fonction sont séparés par une virgule
- Les expressions sont acceptées comme paramètres
- Les paramètres de fonction optionnels (ou parties de paramètres) sont indiqués par < >

abs(value)

La valeur absolue (à partir de 0) d'une valeur.

Paramètre :

- **value** - la valeur à vérifier

Par exemple, la valeur absolue de « 3 » ou de « -3 » sera « 3 ».

Exemple :

```
abs(last(/host/key))>10
```

acos(value)

L'arccosinus d'une valeur sous forme d'angle, exprimé en radians.

Paramètre :

- **value** - la valeur à vérifier

La valeur doit être comprise entre -1 et 1. Par exemple, l'arccosinus de la valeur « 0.5 » sera « 2.0943951 ».

Exemple :

```
acos(last(/host/key))
```

asin(value)

L'arcsinus d'une valeur sous forme d'angle, exprimé en radians.

Paramètre :

- **value** - la valeur à vérifier

La valeur doit être comprise entre -1 et 1. Par exemple, l'arcsinus d'une valeur « 0.5 » sera « -0.523598776 ».

Exemple :

```
asin(last(/host/key))
```

atan(value)

L'arc tangente d'une valeur sous forme d'angle, exprimée en radians.

Paramètre :

- **value** - la valeur à vérifier

Par exemple, l'arc tangente d'une valeur « 1 » sera « 0.785398163 ».

Exemple :

```
atan(last(/hôte/key))
```

atan2(value,abscissa)

L'arctangente de l'ordonnée (value) et des coordonnées de l'abscisse, spécifiée sous forme d'angle, exprimée en radians.

Paramètres :

- **value** - la valeur à vérifier ;
- **abscissa** - la valeur de l'abscisse.

Par exemple, l'arctangente des coordonnées de l'ordonnée et de l'abscisse d'une valeur « 1 » sera « 2.21429744 ».

Exemple :

```
atan2(last(/host/key),2)
```

avg(<value1>,<value2>,...)

La valeur moyenne des valeurs d'élément référencées.

Paramètre :

- **valueX** - la valeur renvoyée par une autre fonction qui fonctionne avec l'historique des éléments.

Exemple :

```
avg(avg(/host/key,1h),avg(/host2/key2,1h))
```

cbrt(value)

La racine cubique d'une valeur.

Paramètre :

- **value** - la valeur à vérifier

Par exemple, la racine cubique de « 64 » sera « 4 », celle de « 63 » sera « 3.97905721 ».

Exemple :

```
cbrt(last(/host/key))
```

ceil(value)

Arrondit la valeur à l'entier supérieur ou égal le plus proche.

Paramètre :

- **value** - la valeur à vérifier

Par exemple, « 2.4 » sera arrondi à « 3 ». Voir aussi floor().

Exemple :

```
ceil(last(/host/key))
```

cos(value)

Le cosinus d'une valeur, où la valeur est un angle exprimé en radians.

Paramètre :

- **value** - la valeur à vérifier

Par exemple, le cosinus d'une valeur « 1 » sera « 0.54030230586 ».

Exemple :

```
cos(last(/host/key))
```

cosh(value)

Le cosinus hyperbolique d'une valeur. Renvoie la valeur sous forme de nombre réel, et non en notation scientifique.

Paramètre :

- **value** - la valeur à vérifier

Par exemple, le cosinus hyperbolique de la valeur « 1 » sera « 1.54308063482 ».

Exemple :

```
cosh(last(/host/key))
```

cot(value)

La cotangente d'une valeur, où la valeur est un angle exprimé en radians.

Paramètre :

- **value** - la valeur à vérifier

Par exemple, la cotangente d'une valeur « 1 » sera « 0.54030230586 ».

Exemple :

```
cot(last(/host/key))
```

degrees(value)

Convertit une valeur de radians en degrés.

Paramètre :

- **value** - la valeur à vérifier

Par exemple, une valeur « 1 » convertie en degrés donnera « 57.2957795 ».

Exemple :

```
degrees(last(/host/key))
```

e

Le nombre d'Euler (2.718281828459045).

Exemple :

```
e()
```

exp(value)

Le nombre d'Euler élevé à la puissance d'une valeur.

Paramètre :

- **value** - la valeur à vérifier

Par exemple, le nombre d'Euler élevé à la puissance de la valeur « 2 » sera « 7.38905609893065 ».

Exemple :

```
exp(last(/host/key))
```

expm1(value)

Le nombre d'Euler élevé à la puissance d'une valeur, moins 1.

Paramètre :

- **value** - la valeur à vérifier

Par exemple, le nombre d'Euler élevé à la puissance de la valeur « 2 », moins 1, sera « 6.38905609893065 ».

Exemple :

```
expm1(last(/host/key))
```

floor(value)

Arrondit la valeur à l'entier inférieur ou égal le plus proche.

Paramètre :

- **value** - la valeur à vérifier

Par exemple, « 2.6 » sera arrondi à « 2 ». Voir aussi ceil().

Exemple :

```
floor(last(/host/key))
```

log(value)

Le logarithme naturel.

Paramètre :

- **value** - la valeur à vérifier

Par exemple, le logarithme naturel d'une valeur « 2 » sera « 0.69314718055994529 ».

Exemple :

```
log(last(/host/key))
```

log10(value)

Le logarithme décimal.

Paramètre :

- **value** - la valeur à vérifier

Par exemple, le logarithme décimal d'une valeur « 5 » sera « 0.69897000433 ».

Exemple :

```
log10(last(/host/key))
```

max(<value1>,<value2>,...)

La valeur la plus élevée des valeurs d'élément référencées.

Paramètre :

- **valueX** - la valeur renvoyée par une autre fonction qui fonctionne avec l'historique des éléments.

Exemple :

```
max(avg(/host/key,1h),avg(/host2/key2,1h))
```

min(<value1>,<value2>,...)

La valeur la plus faible des valeurs d'élément référencées.

Paramètre :

- **valueX** - la valeur renvoyée par une autre fonction qui fonctionne avec l'historique des éléments.

Exemple :

```
min(avg(/host/key,1h),avg(/host2/key2,1h))
```

mod(value,denominator)

Le reste de la division.

Paramètres :

- **value** - la valeur à vérifier ;
- **denominator** - le dénominateur de la division.

Par exemple, le reste de la division d'une valeur « 5 » par le dénominateur « 2 » sera « 1 ».

Exemple :

```
mod(last(/host/key),2)
```

pi

La constante Pi (3.14159265358979).

Exemple :

```
pi()
```

power(valeur,valeur de puissance)

La puissance d'une valeur.

Paramètres :

- **valeur** - la valeur à vérifier ;
- **valeur de puissance** - la puissance N à utiliser.

Par exemple, la 3e puissance d'une valeur « 2 » sera « 8 ».

Exemple :

```
power(last(/host/key),3)
```

radians(value)

Convertit une valeur de degrés en radians.

Paramètre :

- **value** - la valeur à vérifier

Par exemple, une valeur « 1 » convertie en radians sera « 0.0174532925 ».

Exemple :

```
radians(last(/host/key))
```

rand

Renvoie une valeur entière aléatoire. Il s'agit d'un nombre pseudo-aléatoire généré en utilisant le temps comme graine (suffisant à des fins mathématiques, mais pas pour la cryptographie).

Exemple :

```
rand()
```

round(value,decimal places)

Arrondit la valeur au nombre de décimales spécifié.

Paramètres :

- **value** - la valeur à vérifier ;
- **decimal places** - spécifiez le nombre de décimales pour l'arrondi (0 est également possible).

Par exemple, une valeur « 2.5482 » arrondie à 2 décimales deviendra « 2.55 ».

Exemple :

```
round(last(/host/key),2)
```

signum(value)

Renvoie '-1' si une valeur est négative, '0' si une valeur est nulle, '1' si une valeur est positive.

Paramètre :

- **value** - la valeur à vérifier.

Exemple :

```
signum(last(/host/key))
```

sin(value)

Le sinus d'une valeur, où la valeur est un angle exprimé en radians.

Paramètre :

- **value** - la valeur à vérifier

Par exemple, le sinus d'une valeur « 1 » sera « 0.8414709848 ».

Exemple :

```
sin(last(/host/key))
```

sinh(value)

Le sinus hyperbolique d'une valeur.

Paramètre :

- **value** - la valeur à vérifier

Par exemple, le sinus hyperbolique d'une valeur « 1 » sera « 1.17520119364 ».

Exemple :

```
sinh(last(/host/key))
```

sqrt(value)

La racine carrée d'une valeur.
 Cette fonction échouera avec une valeur négative.

Paramètre :

- **value** - la valeur à vérifier

Par exemple, la racine carrée d'une valeur « 3.5 » sera « 1.87082869339 ».

Exemple :

```
sqrt(last(/host/key))
```

sum(<value1>,<value2>,...)

La somme des valeurs des éléments référencés.

Paramètre :

- **valueX** - la valeur renvoyée par une autre fonction qui fonctionne avec l'historique des éléments.

Exemple :

```
sum(avg(/host/key,1h),avg(/host2/key2,1h))
```

tan(value)

La tangente d'une valeur.

Paramètre :

- **value** - la valeur à vérifier

Par exemple, la tangente d'une valeur « 1 » sera « 1.55740772465 ».

Exemple :

```
tan(last(/host/key))
```

truncate(value,decimal places)

Tronquer la valeur au nombre de décimales spécifié.

Paramètres :

- **value** - la valeur à vérifier ;
- **decimal places** - spécifiez le nombre de décimales pour la troncature (0 est également possible).

Par exemple, une valeur « 2.5482 » tronquée à 2 décimales deviendra « 2.54 ».

Exemple :

```
truncate(last(/host/key),2)
```

Voir toutes les fonctions prises en charge.

Fonctions d'opérateur

Toutes les fonctions répertoriées ici sont prises en charge dans :

- Expressions de déclencheur
- Formules d'élément calculé
- Macros d'expression

Les fonctions sont répertoriées sans informations supplémentaires. Cliquez sur une fonction pour afficher tous les détails.

Fonction	Description
between	Vérifier si la valeur appartient à la plage donnée.
in	Vérifier si la valeur est égale à au moins l'une des valeurs répertoriées.

Détails de la fonction

Quelques remarques générales sur les paramètres de fonction :

- Les paramètres de fonction sont séparés par une virgule
- Les expressions sont acceptées comme paramètres

between(value,min,max)

Vérifie si la valeur appartient à la plage donnée.
 Types de valeurs pris en charge : *Integer*, *Float*.
 Renvoie : 1 - dans la plage ; 0 - sinon.

Paramètres :

- **value** - la valeur à vérifier ;
- **min** - la valeur minimale ;
- **max** - la valeur maximale.

Exemple :

```
between(last(/host/key),1,10)=1 #déclencheur si la valeur est comprise entre 1 et 10
```

in(value,value1,value2,...valueN)

Vérifie si la valeur est égale à au moins une des valeurs listées.
 Types de valeurs pris en charge : *Integer*, *Float*, *Character*, *Text*, *Log*.
 Renvoie : 1 - si égal ; 0 - sinon.

Paramètres :

- **value** - la valeur à vérifier ;

- **valueX** - les valeurs listées (les valeurs de chaîne doivent être entre guillemets doubles).

La valeur est comparée aux valeurs listées comme des nombres, si toutes ces valeurs peuvent être converties en valeurs numériques ; sinon, elle est comparée comme une chaîne.

Exemple :

```
in(last(/host/key),5,10)=1 #déclencheur si la dernière valeur est égale à 5 ou 10
in("text",last(/host/key),last(/host/key,#2))=1 #déclencheur si "text" est égal à l'une des 2 dernières va
```

Voir toutes les fonctions prises en charge.

Fonctions prédictives

Toutes les fonctions répertoriées ici sont prises en charge dans :

- Expressions de déclencheur
- Formules d'élément calculé
- Macros d'expression

Les fonctions sont répertoriées sans information supplémentaire. Cliquez sur la fonction pour afficher les détails complets.

Function	Description
forecast	La valeur future, le maximum, le minimum, le delta ou la moyenne de l'élément.
timeleft	Le temps, en secondes, nécessaire pour qu'un élément atteigne le seuil spécifié.

Paramètres communs

- `/host/key` est un premier paramètre obligatoire commun aux fonctions faisant référence à l'historique de l'élément du hôte.
- `(sec|#num)<:time shift>` est un deuxième paramètre commun aux fonctions faisant référence à l'historique de l'élément du hôte, où :
  - **sec** - période d'évaluation maximale en secondes (des suffixes de temps peuvent être utilisés), ou
  - **#num** - plage d'évaluation maximale dans les dernières valeurs collectées (si précédée d'un symbole dièse).
  - **time shift** (facultatif) - permet de déplacer le point d'évaluation dans le passé. Consultez plus de détails sur la spécification du décalage temporel.

Détails des fonctions

Quelques remarques générales sur les paramètres des fonctions :

- Les paramètres des fonctions sont séparés par une virgule
- Les paramètres de fonction facultatifs (ou parties de paramètres) sont indiqués par < >
- Les paramètres spécifiques à chaque fonction sont décrits avec chaque fonction
- Les paramètres `/host/key` et `(sec|#num)<:time shift>` ne doivent jamais être entre guillemets

forecast(/host/key,(sec|#num)<:time shift>,time,<fit>,<mode>)

La valeur future, maximale, minimale, la variation ou la moyenne de l'élément.
 Types de valeurs pris en charge : *Float*, *Integer*.

Paramètres :

- Voir les paramètres communs ;
- **time** - l'horizon de prévision en secondes (par exemple, 1, 60) (les suffixes de temps peuvent être utilisés, par exemple 30s, 1m) ; les valeurs négatives sont prises en charge ;
- **fit** (facultatif ; doit être placé entre guillemets doubles) - la fonction utilisée pour ajuster les données historiques. Ajustements pris en charge :
linear - fonction linéaire (par défaut)
polynomialN - polynôme de degré N (1 <= N <= 6)
exponential - fonction exponentielle
logarithmic - fonction logarithmique
power - fonction puissance
Notez que *polynomial1* est équivalent à *linear* ;
- **mode** (facultatif ; doit être placé entre guillemets doubles) - le résultat demandé. Modes pris en charge :
value - valeur (par défaut)
max - maximum
min - minimum
delta - *max-min*
avg - moyenne
Notez que *value* estime la valeur de l'élément au moment `now + time` ; *max*, *min*, *delta* et *avg* évaluent l'estimation de la valeur de l'élément sur l'intervalle compris entre `now` et `now + time`.

Remarques :

- Si la valeur à retourner est supérieure à 1.7976931348623158E+308 ou inférieure à -1.7976931348623158E+308, la valeur retournée est limitée respectivement à 1.7976931348623158E+308 ou -1.7976931348623158E+308 ;

- Devient non pris en charge uniquement en cas de mauvaise utilisation dans l'expression (type d'élément incorrect, paramètres non valides) ; sinon, retourne -1 en cas d'erreur ;
- Consultez également les informations supplémentaires sur les fonctions de déclencheur prédictives.

Exemples :

```
forecast(/host/key,#10,1h) #prévoir la valeur de l'élément dans une heure sur la base des 10 dernières val
forecast(/host/key,1h,30m) #prévoir la valeur de l'élément dans 30 minutes sur la base des données de la d
forecast(/host/key,1h:now-1d,12h) #prévoir la valeur de l'élément dans 12 heures sur la base des données d
forecast(/host/key,1h,10m,"exponential") #prévoir la valeur de l'élément dans 10 minutes sur la base des d
forecast(/host/key,1h,2h,"polynomial3","max") #prévoir la valeur maximale que l'élément peut atteindre au
forecast(/host/key,#2,-20m) #estimer la valeur de l'élément il y a 20 minutes sur la base des deux dernièr
```

timeleft(/host/key,(sec|#num)<:time shift>,threshold,<fit>)

Le temps, en secondes, nécessaire à un élément pour atteindre le seuil spécifié.
 Types de valeurs pris en charge : *Float*, *Integer*.

Paramètres :

- Voir les paramètres communs ;
- **threshold** - la valeur à atteindre (des suffixes d'unité peuvent être utilisés) ;
- **fit** (facultatif ; doit être placé entre guillemets doubles) - voir forecast().

Commentaires :

- Si la valeur à retourner est supérieure à 1.7976931348623158E+308, la valeur retournée est limitée à 1.7976931348623158E+308 ;
- Retourne 1.7976931348623158E+308 si le seuil ne peut pas être atteint ;
- Devient non pris en charge uniquement en cas de mauvaise utilisation dans l'expression (type d'élément incorrect, paramètres non valides) ; sinon, retourne -1 en cas d'erreur ;
- Voir également les informations supplémentaires sur les fonctions de déclencheur prédictives.

Exemples :

```
timeleft(/host/key,#10,0) #le temps nécessaire pour que la valeur de l'élément atteigne zéro, sur la base
timeleft(/host/key,1h,100) #le temps nécessaire pour que la valeur de l'élément atteigne 100, sur la base
timeleft(/host/key,1h:now-1d,100) #le temps nécessaire pour que la valeur de l'élément atteigne 100, sur l
timeleft(/host/key,1h,200,"polynomial2") #le temps nécessaire pour que la valeur de l'élément atteigne 200
```

Voir toutes les fonctions prises en charge.

Fonctions de déclencheur prédictives

Vue d'ensemble

Il arrive parfois que des signes annoncent un problème à venir. Ces signes peuvent être détectés afin que des mesures puissent être prises à l'avance pour prévenir le problème ou au moins en minimiser l'impact.

Zabbix dispose d'outils permettant de prédire le comportement futur du système surveillé sur la base des données historiques. Ces outils sont mis en œuvre au moyen de fonctions de déclencheur prédictives.

Fonctions

Avant de configurer un déclencheur, il est nécessaire de définir ce qu'est un état de problème et de déterminer combien de temps est nécessaire pour intervenir. Il existe ensuite deux façons de configurer un déclencheur signalant une situation potentiellement indésirable. Premièrement : le déclencheur doit se déclencher lorsque le système est censé être dans un état de problème après le « temps d'intervention ». Deuxièmement : le déclencheur doit se déclencher lorsque le système va atteindre l'état de problème dans un délai inférieur au « temps d'intervention ». Les fonctions de déclencheur correspondantes à utiliser sont **forecast** et **timeleft**. Notez que l'analyse statistique sous-jacente est fondamentalement identique pour les deux fonctions. Vous pouvez configurer un déclencheur de la manière que vous préférez, avec des résultats similaires.

Paramètres

Les deux fonctions utilisent presque le même ensemble de paramètres. Utilisez la liste des fonctions prises en charge comme référence.

Intervalle de temps

Tout d'abord, vous devez spécifier la période historique que Zabbix doit analyser afin d'établir la prédiction. Vous le faites de manière familière au moyen du paramètre `time period` et d'un décalage temporel facultatif, comme vous le faites avec les fonctions **avg**, **count**, **delta**, **max**, **min** et **sum**.

Horizon de prévision

(**forecast** uniquement)
 Le paramètre `time` spécifie jusqu'où dans le futur Zabbix doit extrapoler les dépendances qu'il trouve dans les données historiques. Que vous utilisiez `time_shift` ou non, `time` est toujours compté à partir du moment actuel.

Seuil à atteindre

(**timeleft** uniquement)
 Le paramètre `threshold` spécifie une valeur que l'élément analysé doit atteindre, sans différence qu'elle soit atteinte par au-dessus ou par en dessous. Une fois f(t) déterminée (voir ci-dessous), il faut résoudre l'équation f(t) = `threshold` et renvoyer la racine la plus proche du moment actuel et située à droite de celui-ci, ou 1.7976931348623158E+308 s'il n'existe pas de telle racine.

Note:
Lorsque les valeurs de l'élément s'approchent du seuil puis le franchissent, **timeleft** suppose que cette intersection se situe déjà dans le passé et passe donc à l'intersection suivante avec le niveau `threshold`, s'il y en a une. Il est recommandé d'utiliser les prévisions comme complément au diagnostic habituel des problèmes, et non comme substitut.[1]

Fonctions d'ajustement

La fonction `fit` par défaut est la fonction *linéaire*. Mais si votre système surveillé est plus complexe, vous avez davantage d'options parmi lesquelles choisir.

`fit`	x = f(t)
linear	$x = a + b*t$
polynomialN[2]	$x = a_0 + a_1*t + a_2*t^2 + ... + a_n*t^n$
exponential	$x = a*exp(b*t)$
logarithmic	$x = a + b*log(t)$
power	$x = a*t^b$

Modes

(**forecast** uniquement)
 Chaque fois qu'une fonction de déclencheur est évaluée, elle obtient les données de la période d'historique spécifiée et ajuste une fonction spécifiée à ces données. Ainsi, si les données sont légèrement différentes, la fonction ajustée sera légèrement différente. Si nous calculons simplement la valeur de la fonction ajustée à un moment spécifié dans le futur, vous ne saurez rien de la manière dont l'élément analysé est censé se comporter entre maintenant et ce moment futur. Pour certaines options de `fit` (comme *polynomial*), une simple valeur future peut être trompeuse.

`mode`	Résultat de **forecast**
value	f(now + `time`)
max	$max_{now <= t <= now + time}$ f(t)
min	$min_{now <= t <= now + time}$ f(t)
delta	*max - min*
avg	moyenne de f(t) (now <= t <= now + `time`) selon la définition

Détails

Pour éviter les calculs avec de très grands nombres, nous considérons l'horodatage de la première valeur dans la période spécifiée, plus 1 ns, comme un nouveau temps zéro (le temps epoch actuel est de l'ordre de 10^9, l'epoch au carré est de 10^{18}, la précision double est d'environ 10^{-16}). 1 ns est ajouté afin de fournir uniquement des valeurs de temps positives pour les ajustements *logarithmiques* et *de puissance*, qui impliquent le calcul de log(t). Le décalage temporel n'affecte pas les fonctions *linéaires*, *polynomiales*, *exponentielles* (hormis des calculs plus simples et plus précis), mais modifie la forme des fonctions *logarithmiques* et *de puissance*.

Erreurs potentielles

Les fonctions renvoient -1 dans les situations suivantes :

- la période d'évaluation spécifiée ne contient aucune donnée ;
- le résultat de l'opération mathématique n'est pas défini[3] ;
- des complications numériques surviennent (malheureusement, pour certains ensembles de données d'entrée, l'étendue et la précision du format à virgule flottante en double précision deviennent insuffisantes)[4].

Note:
Aucun avertissement ni aucune erreur ne sont signalés si l'ajustement choisi décrit mal les données fournies ou s'il y a tout simplement trop peu de données pour une prédiction précise.

Exemples et gestion des erreurs

Pour obtenir un avertissement lorsque vous êtes sur le point de manquer d'espace disque libre sur votre hôte, vous pouvez utiliser une expression de déclencheur comme celle-ci :

```
timeleft(/host/vfs.fs.size[/,free],1h,0)<1h
```

Cependant, le code d'erreur -1 peut entrer en jeu et placer votre déclencheur dans un état de problème. En général, c'est une bonne chose, car vous recevez un avertissement indiquant que vos prévisions ne fonctionnent pas correctement et que vous devriez les examiner plus en détail pour en comprendre la raison. Mais parfois, c'est problématique, car -1 peut simplement signifier qu'aucune donnée sur l'espace disque libre de l'hôte n'a été obtenue au cours de la dernière heure. Si vous recevez trop de fausses alertes positives, envisagez d'utiliser une expression de déclencheur plus complexe [5] :

```
timeleft(/host/vfs.fs.size[/,free],1h,0)<1h and timeleft(/host/vfs.fs.size[/,free],1h,0)<>-1
```

La situation est un peu plus difficile avec **forecast**. Tout d'abord, -1 peut ou non placer le déclencheur dans un état de problème selon que vous avez une expression comme `forecast(/host/item,(...))<...` ou comme `forecast(/host/item,(...))>...`

En outre, -1 peut être une prévision valide s'il est normal que la valeur de l'élément soit négative. Mais la probabilité de cette situation dans un contexte réel est négligeable (voir comment fonctionne l'opérateur =). Ajoutez donc `... or forecast(/host/item,(...))=-1` ou `... and forecast(/host/item,(...))<>-1` si vous souhaitez respectivement traiter ou ne pas traiter -1 comme un problème.

Notes de bas de page

[1] Par exemple, un déclencheur simple comme `timeleft(/host/item,1h,X) < 1h` peut passer à l'état de problème lorsque la valeur de l'élément approche de X, puis se rétablir soudainement une fois la valeur X atteinte. Si le problème est que la valeur de l'élément est inférieure à X, utilisez : `last(/host/item) < X or timeleft(/host/item,1h,X) < 1h` Si le problème est que la valeur de l'élément est supérieure à X, utilisez : `last(/host/item) > X or timeleft(/host/item,1h,X) < 1h`

[2] Le degré du polynôme peut aller de 1 à 6, *polynomial1* est équivalent à *linear*. Cependant, utilisez les polynômes de degré supérieur avec prudence. Si la période d'évaluation contient moins de points que nécessaire pour déterminer les coefficients du polynôme, le degré du polynôme sera abaissé (par exemple, *polynomial5* est demandé, mais il n'y a que 4 points, par conséquent *polynomial3* sera ajusté).

[3] Par exemple, l'ajustement de fonctions *exponential* ou *power* implique le calcul de log() des valeurs de l'élément. Si les données contiennent des zéros ou des nombres négatifs, vous obtiendrez une erreur, car log() n'est défini que pour les valeurs positives.

[4] Pour les ajustements *linear*, *exponential*, *logarithmic* et *power*, tous les calculs nécessaires peuvent être écrits explicitement. Pour *polynomial*, seule *value* peut être calculée sans étape supplémentaire. Le calcul de *avg* implique le calcul de la primitive du polynôme (analytiquement). Le calcul de *max*, *min* et *delta* implique le calcul de la dérivée du polynôme (analytiquement) et la recherche de ses racines (numériquement). La résolution de f(t) = 0 implique la recherche des racines du polynôme (numériquement).

[5] Mais dans ce cas, -1 peut amener votre déclencheur à sortir de l'état de problème. Pour être totalement protégé, utilisez : `timeleft(/host/vfs.fs.size[/,free],1h,0)<1h and ({TRIGGER.VALUE}=0 and timeleft(/host/vfs.fs.size[/,free], or {TRIGGER.VALUE}=1)`

Fonctions de chaîne

Toutes les fonctions répertoriées ici sont prises en charge dans :

- Expressions de déclencheur
- Formules d'éléments calculés
- Macros d'expression

Les fonctions sont répertoriées sans informations supplémentaires. Cliquez sur une fonction pour afficher tous les détails.

Fonction	Description
ascii	Le code ASCII du caractère le plus à gauche de la valeur.
bitlength	La longueur de la valeur en bits.
bytelength	La longueur de la valeur en octets.
char	Renvoie le caractère en interprétant la valeur comme un code ASCII.

Fonction	Description
concat	La chaîne résultant de la concaténation des valeurs des éléments référencés ou des valeurs constantes.
contains	Renvoie 1 si la valeur contient la sous-chaîne spécifiée par `search value`, sinon 0.
insert	Insère les caractères ou espaces spécifiés dans la chaîne de caractères, à partir de la position spécifiée dans la chaîne.
jsonpath	Renvoie le résultat JSONPath.
left	Renvoie les caractères les plus à gauche de la valeur.
length	La longueur de la valeur en caractères.
ltrim	Supprime les caractères spécifiés au début de la chaîne.
mid	Renvoie une sous-chaîne de N caractères commençant à la position de caractère spécifiée par `start`. Cette fonction est un alias de la fonction `substring()`.
repeat	Répète une chaîne.
replace	Recherche le motif dans la valeur et le remplace par la chaîne de remplacement.
right	Renvoie les caractères les plus à droite de la valeur.
rtrim	Supprime les caractères spécifiés à la fin de la chaîne.
substring	Renvoie une sous-chaîne de N caractères commençant à la position de caractère spécifiée par `start`.
trim	Supprime les caractères spécifiés au début et à la fin de la chaîne.
xmlxpath	Renvoie le résultat XML XPath.

Détails des fonctions

Quelques remarques générales sur les paramètres de fonction :

- Les paramètres de fonction sont séparés par une virgule
- Les expressions sont acceptées comme paramètres
- Les paramètres de chaîne doivent être placés entre guillemets doubles ; sinon, ils risquent d'être mal interprétés
- Les paramètres de fonction facultatifs (ou parties de paramètres) sont indiqués par < >

ascii(value)

Le code ASCII du caractère le plus à gauche de la valeur.
 Types de valeurs pris en charge : *String*, *Text*, *Log*.

Paramètre :

- **value** - la valeur à vérifier

Par exemple, une valeur comme 'Abc' renverra '65' (code ASCII de 'A').

Exemple :

```
ascii(last(/host/key))
```

bitlength(value)

La longueur de value en bits.
 Types de valeurs pris en charge : *String*, *Text*, *Log*, *Integer*.

Paramètre :

- **value** - la valeur à vérifier

Exemple :

```
bitlength(last(/host/key))
```

bytelength(value)

La longueur de value en octets.
 Types de valeurs pris en charge : *String*, *Text*, *Log*, *Integer*.

Paramètre :

- **value** - la valeur à vérifier

Exemple :

```
bytelength(last(/host/key))
```

char(value)

Renvoie le caractère en interprétant la valeur comme un code ASCII.
 Types de valeurs pris en charge : *Integer*.

Paramètre :

- **value** - la valeur à vérifier

La valeur doit être comprise dans la plage 0-255. Par exemple, une valeur telle que « 65 » (interprétée comme un code ASCII) renverra « A ».

Exemple :

```
char(last(/host/key))
```

concat(<value1>,<value2>,...)

La chaîne résultant de la concaténation des valeurs d'élément référencées ou de valeurs constantes.
 Types de valeurs pris en charge : *String*, *Text*, *Log*, *Float*, *Integer*.

Paramètre :

- **valueX** - la valeur renvoyée par l'une des fonctions d'historique ou une valeur constante (chaîne, entier ou nombre à virgule flottante). Doit contenir au moins deux paramètres.

Par exemple, une valeur telle que 'Zab' concaténée à 'bix' (la chaîne constante) renverra 'Zabbix'.

Exemples :

```
concat(last(/host/key),"bix")
concat("1 min: ",last(/host/system.cpu.load[all,avg1]),", 15 min: ",last(/host/system.cpu.load[all,avg15])
```

contains(value, search value)

Renvoie 1 si la valeur contient la sous-chaîne spécifiée par `search value`, sinon 0.| Types de valeurs pris en charge : *String*, *Text*, *Log*.

Paramètres :

- **value** - la valeur à vérifier ;
- **search value** - la valeur à rechercher.

Remarques :

- La recherche est sensible à la casse.
- Lorsque la valeur est valide et que `search value` est vide, la fonction renvoie 1.

Par exemple, la fonction renvoie 1 si la valeur contient « Zabbix ».

Exemple :

```
contains(last(/host/key),"Zabbix")
```

insert(value,start,length,replacement)

Insère les caractères spécifiés ou des espaces dans la chaîne de caractères à partir de la position spécifiée dans la chaîne.
 Types de valeurs pris en charge : *String*, *Text*, *Log*.

Paramètres :

- **value** - la valeur à vérifier ;
- **start** - position de début ;
- **length** - positions à remplacer ;
- **replacement** - chaîne de remplacement.

Par exemple, une valeur telle que 'Zabbbix' sera remplacée par 'Zabbix' si 'bb' (position de début 3, 2 positions à remplacer) est remplacé par 'b'.

Exemple :

```
insert(last(/host/key),3,2,"b")
```

jsonpath(value,path,<default>)

Renvoie le résultat JSONPath.
 Types de valeurs pris en charge : *String*, *Text*, *Log*.

Paramètres :

- **value** - la valeur à vérifier ;
- **path** - le chemin (doit être entre guillemets) ;
- **default** - la valeur de repli facultative si la requête JSONPath ne renvoie aucune donnée. Notez qu'en cas d'autres erreurs, un échec est renvoyé (par ex. « unsupported construct »).

Exemple :

```
jsonpath(last(/host/proc.get[zabbix_agentd,,,summary]),"$..size")
```

left(value,count)

Renvoie les caractères les plus à gauche de la valeur.
 Types de valeurs pris en charge : *String*, *Text*, *Log*.

Paramètres :

- **value** - la valeur à vérifier ;
- **count** - le nombre de caractères à renvoyer.

Par exemple, vous pouvez renvoyer 'Zab' à partir de 'Zabbix' en spécifiant 3 caractères les plus à gauche à renvoyer. Voir aussi right().

Exemple :

```
left(last(/host/key),3) #renvoie les trois caractères les plus à gauche
```

length(value)

La longueur de value en caractères.
 Types de valeurs pris en charge : *String*, *Text*, *Log*.

Paramètre :

- **value** - la valeur à vérifier.

Exemples :

```
length(last(/host/key)) #la longueur de la dernière valeur
length(last(/host/key,#3)) #la longueur de la troisième valeur la plus récente
length(last(/host/key,#1:now-1d)) #la longueur de la valeur la plus récente il y a un jour
```

ltrim(value,<chars>)

Supprime les caractères spécifiés du début de la chaîne.
 Types de valeurs pris en charge : *String*, *Text*, *Log*.

Paramètres :

- **value** - la valeur à vérifier ;
- **chars** (facultatif) - spécifie les caractères à supprimer.

Les espaces sont supprimés par défaut au début de la chaîne (si aucun caractère facultatif n'est spécifié). Voir aussi : rtrim(), trim().

Exemples :

```
ltrim(last(/host/key)) #supprime les espaces du début de la chaîne
ltrim(last(/host/key),"Z") #supprime tous les 'Z' du début de la chaîne
ltrim(last(/host/key)," Z") #supprime tous les espaces et les 'Z' du début de la chaîne
```

mid(value,start,length)

Renvoie une sous-chaîne de N caractères commençant à la position de caractère spécifiée par « start ».
Cette fonction est un alias de la fonction `substring()`.
 Types de valeurs pris en charge : *String*, *Text*, *Log*.

Paramètres :

- **value** - la valeur à vérifier ;
- **start** - la position de début de la sous-chaîne ;
- **length** - le nombre de positions à renvoyer dans la sous-chaîne.

Par exemple, il est possible de renvoyer « abbi » à partir d'une valeur telle que « Zabbix » si la position de début est 2 et que le nombre de positions à renvoyer est 4.

Exemple :

```
mid(last(/host/key),2,4)="abbi"
```

repeat(value,count)

Répète une chaîne.
 Types de valeurs pris en charge : *String*, *Text*, *Log*.

Paramètres :

- **value** - la valeur à vérifier ;
- **count** - le nombre de répétitions.

Exemple :

```
repeat(last(/host/key),2) #répète la valeur deux fois
```

replace(value,pattern,replacement)

Recherchez le motif dans la valeur et remplacez-le par replacement. Toutes les occurrences du motif seront remplacées.
 Types de valeurs pris en charge : *String*, *Text*, *Log*.

Paramètres :

- **value** - la valeur à vérifier ;
- **pattern** - le motif à rechercher ;
- **replacement** - la chaîne par laquelle remplacer le motif.

Exemple :

```
replace(last(/host/key),"ibb","abb") #remplace toutes les occurrences de 'ibb' par 'abb'
```

right(value,count)

Renvoie les caractères les plus à droite de la valeur.
 Types de valeurs pris en charge : *String*, *Text*, *Log*.

Paramètres :

- **value** - la valeur à vérifier ;
- **count** - le nombre de caractères à renvoyer.

Par exemple, vous pouvez renvoyer « bix » à partir de « Zabbix » en indiquant de renvoyer les 3 caractères les plus à droite. Voir aussi left().

Exemple :

```
right(last(/host/key),3) #renvoie les trois caractères les plus à droite
```

rtrim(value,<chars>)

Supprime les caractères spécifiés de la fin de la chaîne.
 Types de valeurs pris en charge : *String*, *Text*, *Log*.

Paramètres :

- **value** - la valeur à vérifier ;
- **chars** (optionnel) - spécifie les caractères à supprimer.

Les espaces sont supprimés par défaut à droite (si aucun caractère optionnel n'est spécifié). Voir aussi : ltrim(), trim().

Exemples :

```
rtrim(last(/host/key)) #supprime les espaces de la fin de la chaîne
rtrim(last(/host/key),"x") #supprime tous les 'x' de la fin de la chaîne
rtrim(last(/host/key),"x ") #supprime tous les 'x' et les espaces de la fin de la chaîne
```

substring(value,start,length)

Renvoie une sous-chaîne de N caractères commençant à la position de caractère spécifiée par « start ».
 Types de valeurs pris en charge : *String*, *Text*, *Log*.

Paramètres :

- **value** - la valeur à vérifier ;
- **start** - la position de début de la sous-chaîne ;
- **length** - le nombre de positions à renvoyer dans la sous-chaîne.

Par exemple, il est possible de renvoyer « abbi » à partir d'une valeur telle que « Zabbix » si la position de début est 2 et que le nombre de positions à renvoyer est 4.

Exemple :

```
substring(last(/host/key),2,4)="abbi"
```

trim(value,<chars>)

Supprime les caractères spécifiés au début et à la fin de la chaîne.
 Types de valeurs pris en charge : *String*, *Text*, *Log*.

Paramètres :

- **value** - la valeur à vérifier ;
- **chars** (facultatif) - spécifie les caractères à supprimer.

Par défaut, les espaces sont supprimés des deux côtés (si aucun caractère facultatif n'est spécifié). Voir aussi : ltrim(), rtrim().

Exemples :

```
trim(last(/host/key)) #supprime les espaces au début et à la fin de la chaîne
trim(last(/host/key),"_") #supprime '_' au début et à la fin de la chaîne
```

xmlxpath(value,path,<default>)

Renvoie le résultat XML XPath.
 Types de valeurs pris en charge : *String*, *Text*, *Log*.

Paramètres :

- **value** - la valeur à vérifier ;
- **path** - le chemin (doit être entre guillemets) ;
- **default** - la valeur de repli facultative si la requête XML XPath renvoie un ensemble de nœuds vide. Elle ne sera pas renvoyée si le résultat vide n'est pas un ensemble de nœuds (c'est-à-dire une chaîne vide). En cas d'autres erreurs, un échec est renvoyé (par exemple, "invalid expression").

Exemple :

```
xmlxpath(last(/host/xml_result),"/response/error/status")
```

Voir toutes les fonctions prises en charge.

Bonnes pratiques pour les déclencheurs

Les déclencheurs sont un outil puissant, mais ils peuvent aussi générer un bruit d'alertes indésirable. Pour obtenir davantage de signaux réels et moins de bruit, suivez ces conseils :

1. Désensibilisez les déclencheurs. Au lieu de déclencher une alerte sur la dernière valeur (trop élevée/trop basse), analysez la moyenne sur une période prolongée à l'aide de fonctions comme **avg**, **min** et **max**.
2. Envisagez d'utiliser la fonction **percentile** (réglez-la sur 95 % ou 5 %) dans les déclencheurs, si vous souhaitez éviter les alertes dues à des pics et des baisses aléatoires.
3. Utilisez l'hystérésis pour éviter le flottement des déclencheurs, c'est-à-dire les changements fréquents d'état du déclencheur (OK ↔ Problème). Une continuité pour l'état de problème peut être définie en ajoutant une expression de récupération (une condition distincte pour la résolution du problème).
4. Utilisez les dépendances des déclencheurs pour éviter les alertes qui ne sont pas liées à la cause racine.
5. Utilisez la gravité des déclencheurs pour n'alerter que sur les problèmes les plus sérieux.
6. Définissez des fenêtres de maintenance.

Hystérésis

Parfois, un intervalle est nécessaire entre les états de problème et de rétablissement, plutôt qu'un simple seuil. Par exemple, si nous voulons définir un déclencheur qui signale un problème lorsque la température de la salle des serveurs dépasse 20 °C et que nous voulons qu'il reste à l'état de problème jusqu'à ce que la température descende en dessous de 15 °C, un simple seuil de déclencheur à 20 °C ne sera pas suffisant.

À la place, nous devons d'abord définir une expression de déclencheur pour l'événement de problème (température supérieure à 20 °C). Ensuite, nous devons définir une condition de rétablissement supplémentaire (température inférieure à 15 °C). Cela se fait en définissant une *Expression de rétablissement* lors de la configuration d'un déclencheur.

Dans ce cas, le rétablissement du problème se déroule en deux étapes :

- D'abord, l'expression de problème (température supérieure à 20 °C) doit être évaluée à FALSE
- Ensuite, l'expression de rétablissement (température inférieure à 15 °C) doit être évaluée à TRUE

L'expression de rétablissement n'est évaluée qu'après la résolution de l'événement de problème. Le seul fait que l'expression de rétablissement soit TRUE ne résout pas un problème si l'expression de problème est toujours TRUE !

Exemple

La température dans la salle des serveurs est trop élevée.

Expression de problème :

```
last(/server/temp)>20
```

Expression de rétablissement :

```
last(/server/temp)<=15
```

Note:
Il n'est pas utile d'utiliser la macro {TRIGGER.VALUE} dans une expression de rétablissement, car cette expression n'est évaluée que lorsque le déclencheur est dans l'état « Problème ». Par conséquent, {TRIGGER.VALUE} sera toujours résolue en « 1 » (ce qui indique un état « Problème ») lors de l'évaluation de l'expression.

Dépendances de déclencheurs

Les dépendances de déclencheurs peuvent être utilisées pour éviter les alertes qui ne sont pas liées à la cause racine.

Voir toutes les bonnes pratiques.

Vue d'ensemble

Parfois, la disponibilité d'un hôte dépend d'un autre. Un serveur situé derrière un routeur devient inaccessible si le routeur tombe en panne. Si des déclencheurs sont configurés pour les deux, vous pouvez recevoir des notifications concernant deux hôtes indisponibles, alors que seul le routeur était en cause.

C'est dans ce cas qu'une dépendance entre hôtes peut être utile. Lorsqu'une dépendance est définie, les notifications concernant les éléments dépendants peuvent être supprimées et seule la notification relative au problème racine peut être envoyée.

Bien que Zabbix ne prenne pas directement en charge les dépendances entre hôtes, celles-ci peuvent être définies à l'aide d'une méthode plus flexible : les dépendances entre déclencheurs. Un déclencheur peut dépendre d'un ou de plusieurs autres déclencheurs.

Dans notre exemple simple, nous ouvrons le formulaire de configuration du déclencheur du serveur et indiquons qu'il dépend du déclencheur correspondant du routeur. Avec cette dépendance, l'état du déclencheur du serveur ne changera pas tant que le déclencheur dont il dépend sera à l'état « PROBLEM » ; aucune action dépendante ne sera donc exécutée et aucune notification ne sera envoyée.

Si le serveur et le routeur sont tous deux indisponibles et qu'une dépendance est définie entre le déclencheur « serveur indisponible » et le déclencheur « routeur indisponible », Zabbix n'exécutera pas les actions associées au déclencheur dépendant.

Tant que le déclencheur parent est à l'état PROBLEM, les éléments dépendants peuvent signaler des valeurs qui ne sont pas fiables. Par conséquent, les déclencheurs dépendants ne seront pas réévalués tant que le déclencheur parent (le routeur dans l'exemple ci-dessus) :

- ne repasse pas de l'état « PROBLEM » à l'état « OK » ;
- ne passe pas de l'état « PROBLEM » à l'état « UNKNOWN » ;
- n'est pas fermé manuellement, par corrélation ou à l'aide des fonctions date et heure et/ou nodata() ;
- n'est pas résolu par la valeur d'un élément qui n'est pas utilisé dans le déclencheur dépendant ;
- n'est pas désactivé, ne possède pas d'élément désactivé ou n'appartient pas à un hôte dont l'élément est désactivé.

Dans tous les cas mentionnés ci-dessus, le déclencheur dépendant (du serveur) ne sera réévalué que lorsqu'une nouvelle métrique le concernant sera reçue. Cela signifie que le déclencheur dépendant peut ne pas être mis à jour immédiatement.

De plus :

- Une dépendance entre déclencheurs peut être ajoutée entre le déclencheur de n'importe quel hôte et celui de n'importe quel autre hôte, à condition que cela n'entraîne pas de dépendance circulaire.
- Une dépendance entre déclencheurs peut être ajoutée d'un modèle à un autre. Si un déclencheur du modèle A dépend d'un déclencheur du modèle B, le modèle A ne peut être lié à un hôte (ou à un autre modèle) qu'avec le modèle B, tandis que le modèle B peut être lié seul à un hôte (ou à un autre modèle).
- Une dépendance entre déclencheurs peut être ajoutée d'un déclencheur de modèle à un déclencheur d'hôte. Dans ce cas, le fait de lier ce modèle à un hôte créera un déclencheur d'hôte qui dépend du même modèle de déclencheur dont dépendait le déclencheur initial. Cela vous permet, par exemple, de disposer d'un modèle dans lequel certains déclencheurs dépendent des déclencheurs du routeur (hôte). Tous les hôtes liés à ce modèle dépendront de ce routeur spécifique.
- Une dépendance entre déclencheurs ne peut pas être ajoutée d'un déclencheur d'hôte à un déclencheur de modèle.
- Une dépendance entre déclencheurs peut être ajoutée d'un prototype de déclencheur à un autre prototype de déclencheur (au sein de la même règle de découverte de bas niveau) ou à un déclencheur réel. Un prototype de déclencheur ne peut pas dépendre d'un prototype de déclencheur provenant d'une autre règle LLD ni d'un déclencheur créé à partir d'un prototype de déclencheur. Un prototype de déclencheur d'hôte ne peut pas dépendre d'un déclencheur provenant d'un modèle.

Configuration

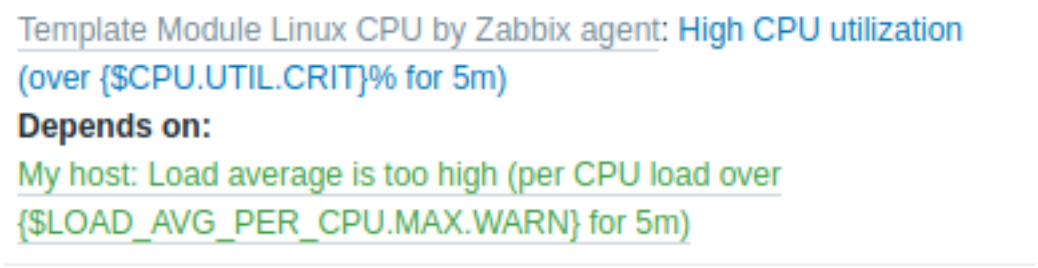
Pour définir une dépendance :

1. Ouvrez l'onglet *Dépendances* dans le formulaire de configuration du déclencheur.
2. Cliquez sur *Ajouter* dans la section *Dépendances* et sélectionnez un ou plusieurs déclencheurs dont dépendra le déclencheur.



3. Cliquez sur *Mettre à jour*.

Le déclencheur indique désormais sa dépendance dans la liste.



Exemple de plusieurs dépendances

Par exemple, l'hôte se trouve derrière Router2 et Router2 se trouve derrière Router1.

```
Zabbix - Router1 - Router2 - hôte
```

Si Router1 est en panne, alors, évidemment, l'hôte et Router2 sont également injoignables, mais recevoir trois notifications indiquant que l'hôte, Router1 et Router2 sont tous en panne est excessif.

Dans ce cas, nous définissons donc deux dépendances :

```
le déclencheur 'hôte en panne' dépend du déclencheur 'Router2 en panne'
le déclencheur 'Router2 en panne' dépend du déclencheur 'Router1 en panne'
```

Avant de modifier l'état du déclencheur 'hôte en panne', Zabbix vérifiera les dépendances de déclencheur correspondantes. Si de telles dépendances sont trouvées et que l'un de ces déclencheurs est dans l'état `Problem`, alors l'état du déclencheur ne sera pas modifié, les actions ne seront pas exécutées et aucune notification ne sera envoyée.

Zabbix effectue cette vérification de manière récursive. Si Router1 ou Router2 est injoignable, le déclencheur de l'hôte ne sera pas mis à jour.

Gravité du déclencheur

La gravité du déclencheur peut être utilisée pour éviter le bruit d'alerte sur les problèmes moins sérieux.

Voir toutes les bonnes pratiques.

Aperçu

La gravité représente l'importance relative d'un déclencheur.



L'attribution d'une gravité n'est pas seulement utile pour la représentation visuelle, mais aussi pour éviter les notifications sur des problèmes mineurs.

Les gravités des déclencheurs sont utilisées pour :

- La représentation visuelle (couleur distincte pour chaque problème créé, basée sur la gravité d'origine du déclencheur)
- La limitation des notifications selon la gravité (voir conditions d'action)
- Les médias utilisateur - différents médias (canaux de notification) pour différentes gravités (par exemple, SMS pour les déclencheurs de gravité *High* et *Disaster*, et e-mail pour les déclencheurs d'autres gravités)
- L'audio dans les alarmes globales - un son différent pour différentes gravités

Les gravités de déclencheur suivantes sont prises en charge par défaut :

Gravité	Couleur	Description
Non classé	Gris	Peut être utilisée lorsque le niveau de gravité d'un événement est inconnu, n'a pas été déterminé, ne fait pas partie du périmètre de supervision habituel, etc., par exemple lors de la configuration initiale, comme espace réservé pour une évaluation future, ou dans le cadre d'un processus d'intégration.
Information	Bleu clair	Peut être utilisée pour des événements informatifs qui ne nécessitent pas d'attention immédiate, mais qui peuvent tout de même fournir des informations utiles.
Avertissement	Jaune	Peut être utilisée pour indiquer un problème potentiel pouvant nécessiter une investigation ou une action, mais qui n'est pas critique.
Moyenne	Orange	Peut être utilisée pour indiquer un problème important qui devrait être traité assez rapidement afin d'éviter d'autres problèmes.
Élevée	Rouge clair	Peut être utilisée pour indiquer des problèmes critiques nécessitant une attention immédiate afin d'éviter des perturbations importantes.
Désastre	Rouge	Peut être utilisée pour indiquer un incident grave nécessitant une action immédiate afin d'éviter, par exemple, des pannes système ou une perte de données.

Les noms et les couleurs des gravités de déclencheur peuvent être **personnalisés** dans *Administration* > *General* > *Trigger displaying options*.
Les couleurs sont partagées entre tous les thèmes de l'interface web.

Traduction des noms de sévérité personnalisés

> **Attention:**
> Si les traductions de l'interface web Zabbix sont utilisées, les noms de sévérité personnalisés remplaceront par défaut les noms traduits.

Les noms de sévérité de déclencheur par défaut peuvent être traduits dans toutes les langues.
Si un nom de sévérité est modifié, un nom personnalisé est utilisé dans toutes les langues et une traduction manuelle supplémentaire est nécessaire.

Procédure de traduction des noms de sévérité personnalisés :

- définir le nom de sévérité personnalisé requis, par exemple « Important »
- modifier <frontend_dir>/locale/<required_locale>/LC_MESSAGES/frontend.po
- ajouter 2 lignes :

```
msgid "Important"
msgstr "<chaîne de traduction>"
```

et enregistrer le fichier.

- créer les fichiers .mo comme décrit dans <frontend_dir>/locale/README

Ici, **msgid** doit correspondre au nouveau nom de sévérité personnalisé et **msgstr** doit être sa traduction dans la langue spécifique.

Cette procédure doit être effectuée après chaque modification du nom de sévérité.

Visualisation

Vue d'ensemble

La visualisation offre différentes façons d'afficher les données de supervision collectées par Zabbix. Les graphiques montrent les tendances et les métriques, les cartes réseau affichent la topologie de l'infrastructure, et les tableaux de bord combinent des widgets personnalisables pour fournir des vues de supervision complètes.

Outils de visualisation

- Graphiques - affichez les données des éléments sous forme de graphiques simples générés automatiquement, créez des graphiques personnalisés ou comparez plusieurs éléments dans des graphiques ad hoc.
- Cartes réseau - créez des représentations visuelles de l'infrastructure réseau avec des éléments et des indicateurs personnalisables.
- Tableaux de bord - personnalisez des mises en page combinant divers widgets de tableau de bord qui visualisent les données dans différents formats.

Graphiques

Vue d'ensemble

Avec la grande quantité de données qui arrivent dans Zabbix, il est beaucoup plus facile pour les utilisateurs de consulter une représentation visuelle de ce qui se passe plutôt que de ne voir que des chiffres.

C'est là que les graphiques interviennent. Les graphiques permettent de comprendre le flux de données en un coup d'œil, de corréler les problèmes, de déterminer quand quelque chose a commencé ou d'illustrer quand une situation risque de devenir problématique.

Zabbix offre aux utilisateurs :

- Des graphiques simples intégrés représentant les données d'un élément.
- La possibilité de créer des graphiques personnalisés plus complexes.
- Un accès rapide à une comparaison de plusieurs éléments dans des graphiques ad hoc.
- Des graphiques vectoriels modernes et personnalisables, ainsi que des graphiques circulaires.

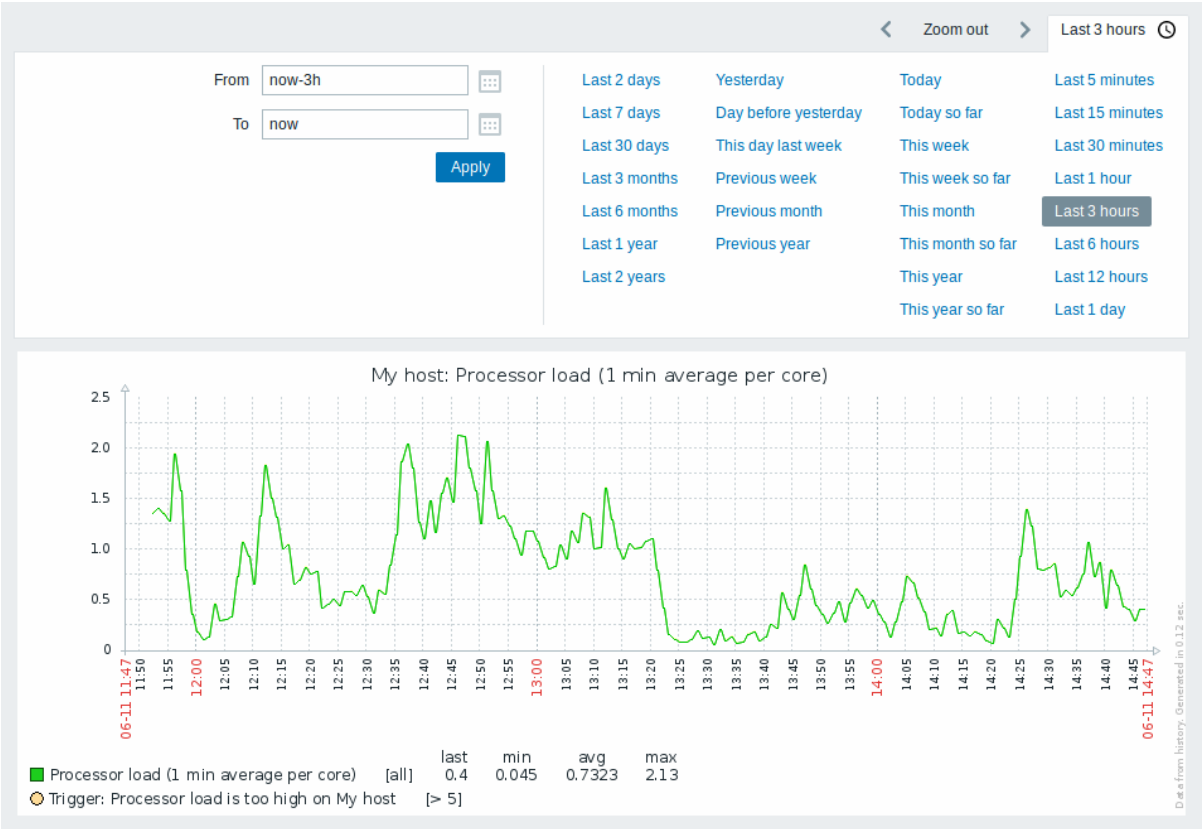
Graphiques simples

Présentation

Des graphiques simples sont disponibles pour visualiser les données collectées par les éléments.

Aucune configuration n'est nécessaire de la part de l'utilisateur pour afficher les graphiques simples. Ils sont mis gratuitement à disposition par Zabbix.

Accédez simplement à *Monitoring* > *Latest data* et cliquez sur le lien *Graph* correspondant à l'élément concerné pour afficher un graphique.



Note:
Des graphiques simples sont disponibles pour tous les éléments numériques. Pour les éléments textuels, un lien vers *History* est disponible dans *Monitoring* > *Latest data*.

Sélecteur de période

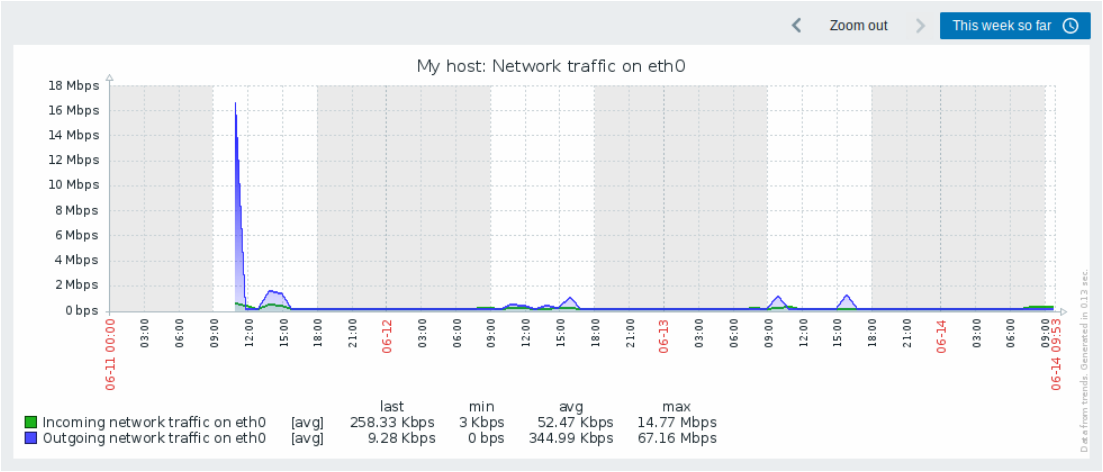
Le sélecteur de période situé au-dessus du graphique vous permet de sélectionner les périodes fréquemment utilisées en un clic de souris. Pour plus d'informations, consultez Sélecteurs de période et d'hôte.

Données récentes vs périodes plus longues

Pour les données très récentes, une **seule** ligne est tracée en reliant chaque valeur reçue. La ligne unique est tracée tant qu'au moins un pixel horizontal est disponible pour une valeur.

Pour les données couvrant une période plus longue, **trois lignes** sont tracées : une ligne vert foncé indique la moyenne, tandis qu'une ligne rose clair et une ligne vert clair indiquent respectivement les valeurs maximale et minimale à ce moment-là. L'espace entre les valeurs hautes et basses est rempli d'un fond jaune.

Le temps de travail (jours ouvrés) est affiché dans les graphiques avec un fond blanc, tandis que le temps non ouvré est affiché en gris (avec le thème par défaut de l'interface *Original blue*).



Le temps de travail est toujours affiché dans les graphiques simples, tandis que son affichage dans les graphiques personnalisés dépend des préférences de l'utilisateur.

Le temps de travail n'est pas affiché si le graphique couvre plus de 3 mois.

Lignes de déclencheur

Les déclencheurs simples sont affichés sous forme de lignes en pointillés noirs sur la couleur de gravité du déclencheur - notez la ligne bleue sur le graphique ainsi que les informations du déclencheur affichées dans la légende. Jusqu'à 3 lignes de déclencheur peuvent être affichées sur le graphique ; s'il y a davantage de déclencheurs, ceux ayant une gravité plus faible sont prioritaires. Les déclencheurs sont toujours affichés dans les graphiques simples, tandis que leur affichage dans les graphiques personnalisés dépend des préférences de l'utilisateur.



Génération à partir de l'historique/des tendances

Les graphiques peuvent être tracés à partir de l'historique ou des tendances d'un élément.

Pour les utilisateurs dont le mode débogage de l'interface est activé, une légende grise verticale s'affiche dans l'angle inférieur droit d'un graphique et indique l'origine des données.

Plusieurs facteurs déterminent si l'historique ou les tendances sont utilisés :

- Durée de conservation de l'historique de l'élément. Par exemple, l'historique d'un élément peut être conservé pendant 14 jours. Dans ce cas, toutes les données datant de plus de quatorze jours proviennent des tendances.
- Densité des données dans le graphique. Si le nombre de secondes à afficher dans un pixel horizontal du graphique dépasse 3600/16, les données de tendance sont affichées (même si l'historique de l'élément est toujours disponible pour la même période).
- Si les tendances sont désactivées, l'historique de l'élément est utilisé pour générer le graphique, s'il est disponible pour cette période.

Absence de données

Pour les éléments avec un intervalle de mise à jour régulier, rien n'est affiché dans le graphique si les données de l'élément ne sont pas collectées.

Cependant, pour les éléments de type trapper et les éléments avec un intervalle de mise à jour planifié (et un intervalle de mise à jour régulier défini sur 0), une ligne droite est tracée jusqu'à la première valeur collectée et depuis la dernière valeur collectée jusqu'à la fin du graphique ; la ligne se situe respectivement au niveau de la première/dernière valeur.

Passer aux valeurs brutes

Une liste déroulante située en haut à droite vous permet de passer du graphique simple aux listes *Valeurs/500 dernières valeurs*. Cela peut être utile pour afficher les valeurs numériques qui composent le graphique.

Les valeurs présentées ici sont brutes : aucune unité ni aucun post-traitement des valeurs n'est utilisé. Le mappage des valeurs est toutefois appliqué.

Problèmes connus

Voir les problèmes connus pour les graphiques.

Graphiques personnalisés

Aperçu

Les graphiques personnalisés, comme leur nom l'indique, offrent des possibilités de personnalisation.

Bien que les graphiques simples soient adaptés à l'affichage des données d'un seul élément, ils n'offrent pas de possibilités de configuration.

Ainsi, si vous souhaitez modifier le style du graphique ou la manière dont les lignes sont affichées, ou comparer plusieurs éléments, par exemple le trafic entrant et sortant dans un seul graphique, vous avez besoin d'un graphique personnalisé.

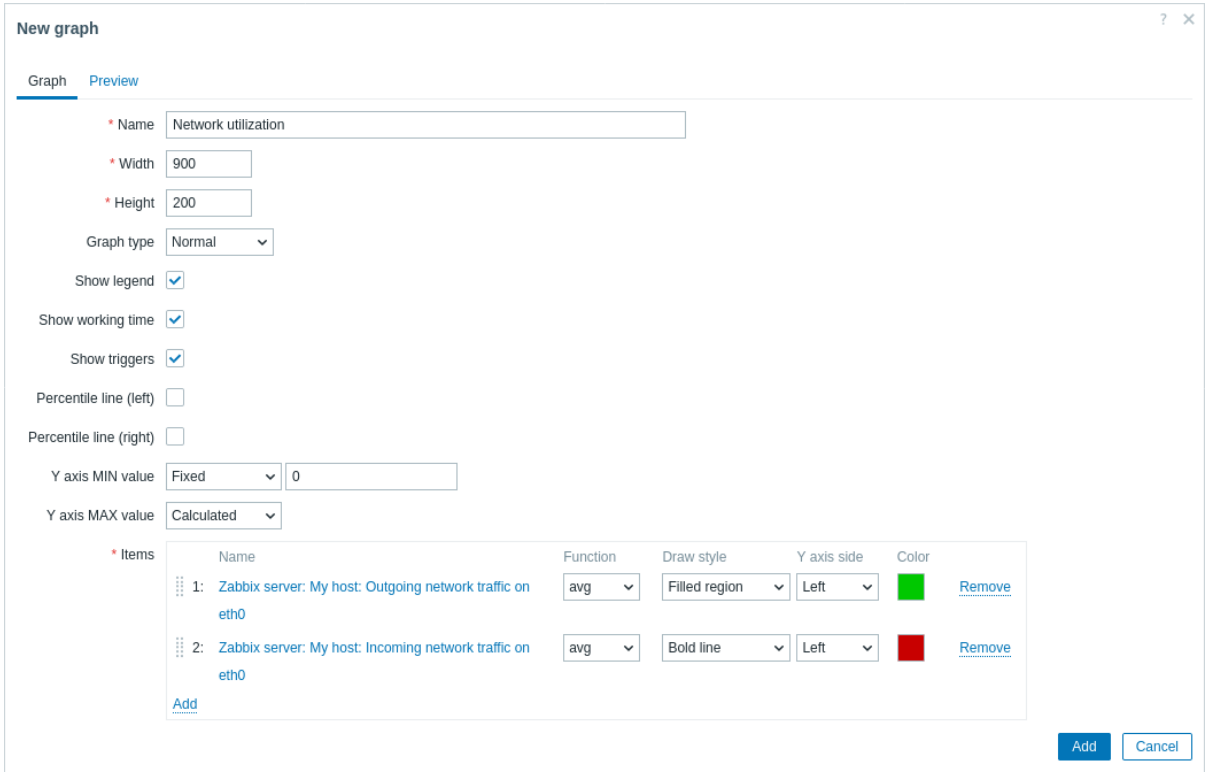
Les graphiques personnalisés sont configurés manuellement.

Ils peuvent être créés pour un hôte, plusieurs hôtes ou un seul modèle.

Configuration de graphes personnalisés

Pour créer un graphe personnalisé, procédez comme suit :

1. Accédez à *Collecte de données* > *Hôtes* (ou *Modèles*).
2. Cliquez sur *Graphes* dans la ligne à côté de l'hôte ou du modèle souhaité.
3. Dans l'écran Graphes, cliquez sur *Créer un graphe*.
4. Modifiez les attributs du graphe.



Tous les champs obligatoires sont marqués d'un astérisque rouge.

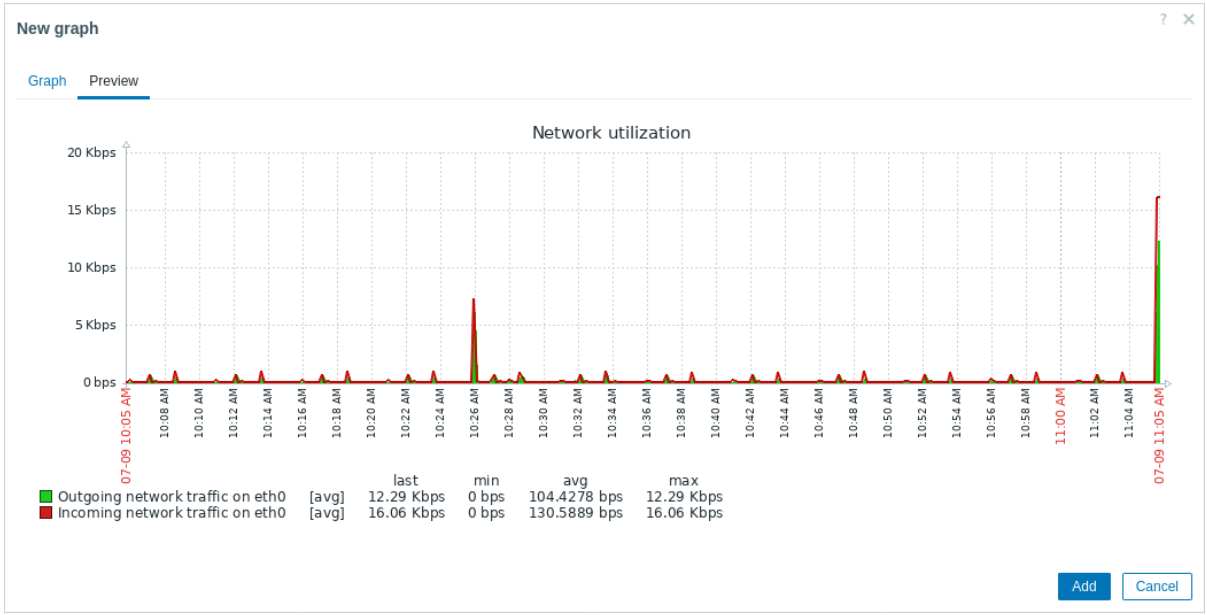
Attributs du graphe :

Paramètre	Description
Nom	Nom unique du graphe. Les expressions macros sont prises en charge dans ce champ, mais uniquement avec les fonctions `avg`, `last`, `min` et `max`, avec le temps comme paramètre (par exemple, `{?avg(/host/key,1h)}`). Les macros {HOST.HOST<1-9>} sont prises en charge pour une utilisation dans cette macro, en faisant référence au premier, deuxième, troisième, etc. hôte du graphe, par exemple, `{?avg(/{HOST.HOST2}/key,1h)}`. Notez que la référence au premier hôte avec cette macro est redondante, car le premier hôte peut être référencé implicitement, par exemple, `{?avg(//key,1h)}`.
Largeur	Largeur du graphe en pixels (pour l'aperçu et les graphes en camembert/explosés uniquement).
Hauteur	Hauteur du graphe en pixels.
Type de graphe	Type de graphe : **Normal** - graphe normal, les valeurs sont affichées sous forme de lignes **Empilé** - graphe empilé, les zones remplies sont affichées **Camembert** - graphe en camembert **Explosé** - graphe en camembert "explosé", les portions sont affichées comme "découpées" du camembert
Afficher la légende	Cochez cette case pour afficher la légende du graphe.
Afficher le temps de travail	Si cette option est sélectionnée, les heures non travaillées seront affichées avec un fond gris. Ce paramètre n'est pas disponible pour les graphes en camembert et en camembert explosé.
Afficher les déclencheurs	Si cette option est sélectionnée, les déclencheurs simples seront affichés sous forme de lignes avec des tirets noirs superposés à la couleur de gravité du déclencheur. Ce paramètre n'est pas disponible pour les graphes en camembert et en camembert explosé.

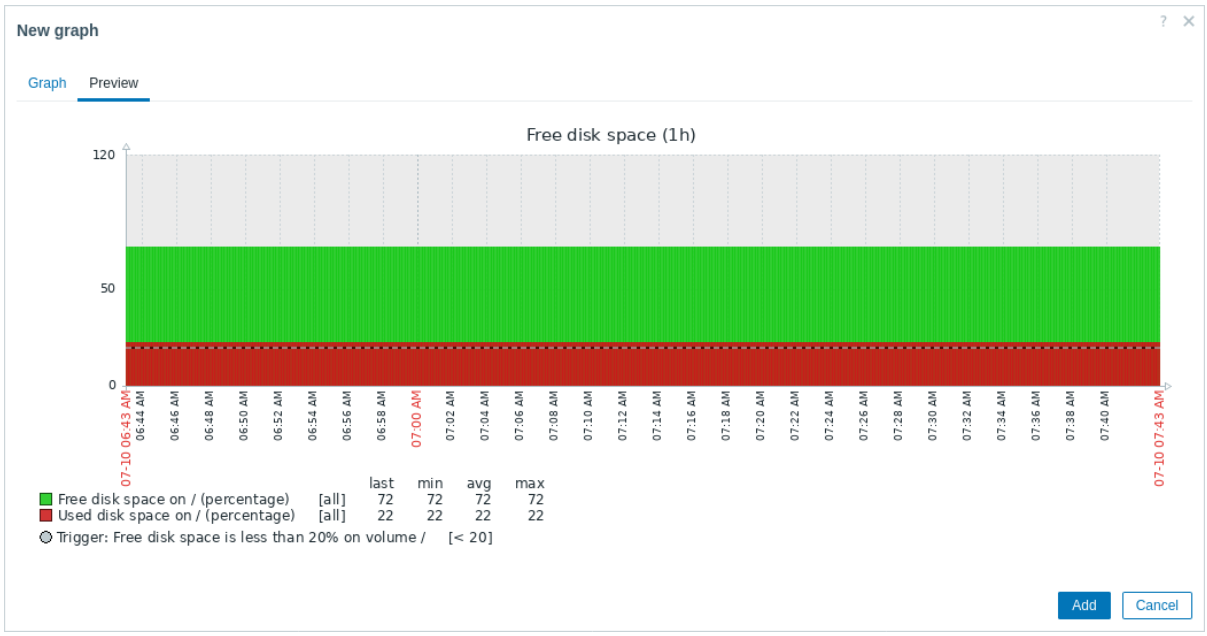
Paramètre	Description
Ligne de percentile (gauche)	Afficher le percentile pour l'axe Y gauche. Par exemple, si le percentile 95 % est défini, la ligne de percentile sera au niveau en dessous duquel se trouvent 95 % des valeurs. Affichée sous forme de ligne vert vif. Disponible uniquement pour les graphes normaux.
Ligne de percentile (droite)	Afficher le percentile pour l'axe Y droit. Par exemple, si le percentile 95 % est défini, la ligne de percentile sera au niveau en dessous duquel se trouvent 95 % des valeurs. Affichée sous forme de ligne rouge vif. Disponible uniquement pour les graphes normaux.
Valeur MIN de l'axe Y	Valeur minimale de l'axe Y : **Calculée** - la valeur minimale de l'axe Y sera calculée automatiquement. **Fixe** - valeur minimale fixe pour l'axe Y. **Élément** - la dernière valeur de l'élément sélectionné sera la valeur minimale. Ce paramètre n'est pas disponible pour les graphes en camembert et en camembert explosé.
Valeur MAX de l'axe Y	Valeur maximale de l'axe Y : **Calculée** - la valeur maximale de l'axe Y sera calculée automatiquement. **Fixe** - valeur maximale fixe pour l'axe Y. **Élément** - la dernière valeur de l'élément sélectionné sera la valeur maximale. Ce paramètre n'est pas disponible pour les graphes en camembert et en camembert explosé.
Vue 3D	Activer le style 3D. Uniquement pour les graphes en camembert et en camembert explosé.
Éléments	Éléments dont les données doivent être affichées dans ce graphe. Cliquez sur *Ajouter* pour sélectionner des éléments. Vous pouvez également sélectionner diverses options d'affichage (fonction, style de tracé, affichage sur l'axe gauche/droit, couleur).
Ordre de tri (0→100)	Ordre de tracé. 0 sera traité en premier. Peut être utilisé pour tracer des lignes ou des zones derrière (ou devant) une autre. Vous pouvez faire glisser et déposer les éléments à l'aide de l'icône au début d'une ligne pour définir l'ordre de tri ou l'élément affiché devant l'autre.
Nom	Le nom de l'élément sélectionné est affiché sous forme de lien. En cliquant sur le lien, vous ouvrez la liste des autres éléments disponibles.
Type	Type (disponible uniquement pour les graphes en camembert et en camembert explosé) : **Simple** - la valeur de l'élément est représentée proportionnellement sur le camembert **Somme du graphe** - la valeur de l'élément représente l'ensemble du camembert Notez que la coloration de l'élément "somme du graphe" ne sera visible que dans la mesure où elle n'est pas occupée par des éléments "proportionnels".
Fonction	Sélectionnez les valeurs à afficher lorsqu'il existe plus d'une valeur par pixel vertical du graphe pour un élément : **all** - affiche toutes les valeurs possibles (minimum, maximum, moyenne) dans le graphe. Notez que pour des périodes plus courtes, ce paramètre n'a aucun effet ; ce n'est que pour des périodes plus longues, lorsque la congestion des données dans un pixel vertical du graphe augmente, que 'all' commence à afficher les valeurs minimum, maximum et moyenne. Cette fonction est disponible uniquement pour le type de graphe *Normal*. Voir aussi : Génération de graphes à partir de l'historique/des tendances. **avg** - affiche les valeurs moyennes **last** - affiche les dernières valeurs. Cette fonction est disponible uniquement si *Camembert/Explosé* est sélectionné comme type de graphe. **max** - affiche les valeurs maximales **min** - affiche les valeurs minimales
Style de tracé	Sélectionnez le style de tracé (disponible uniquement pour les graphes normaux ; pour les graphes empilés, la région remplie est toujours utilisée) à appliquer aux données de l'élément - *Ligne*, *Ligne épaisse*, *Région remplie*, *Point*, *Ligne en pointillés*, *Ligne dégradée*.
Côté de l'axe Y	Sélectionnez le côté de l'axe Y sur lequel afficher les données de l'élément - *Gauche*, *Droite*.
Couleur	Sélectionnez la couleur à appliquer aux données de l'élément.

Aperçu du graphique

Dans l'onglet *Aperçu*, un aperçu du graphique est affiché afin que vous puissiez voir immédiatement ce que vous êtes en train de créer.



Notez que l'aperçu n'affichera aucune donnée pour les éléments de modèle.



Dans cet exemple, faites attention à la ligne en pointillés épaisse affichant le niveau du déclencheur ainsi qu'aux informations du déclencheur affichées dans la légende.

Note:
Pas plus de 3 lignes de déclencheur peuvent être affichées. S'il y a davantage de déclencheurs, ceux ayant une sévérité plus faible sont prioritaires pour l'affichage.

 Si la hauteur du graphique est définie à moins de 120 pixels, aucun déclencheur ne sera affiché dans la légende.

Graphiques ad hoc

Vue d'ensemble

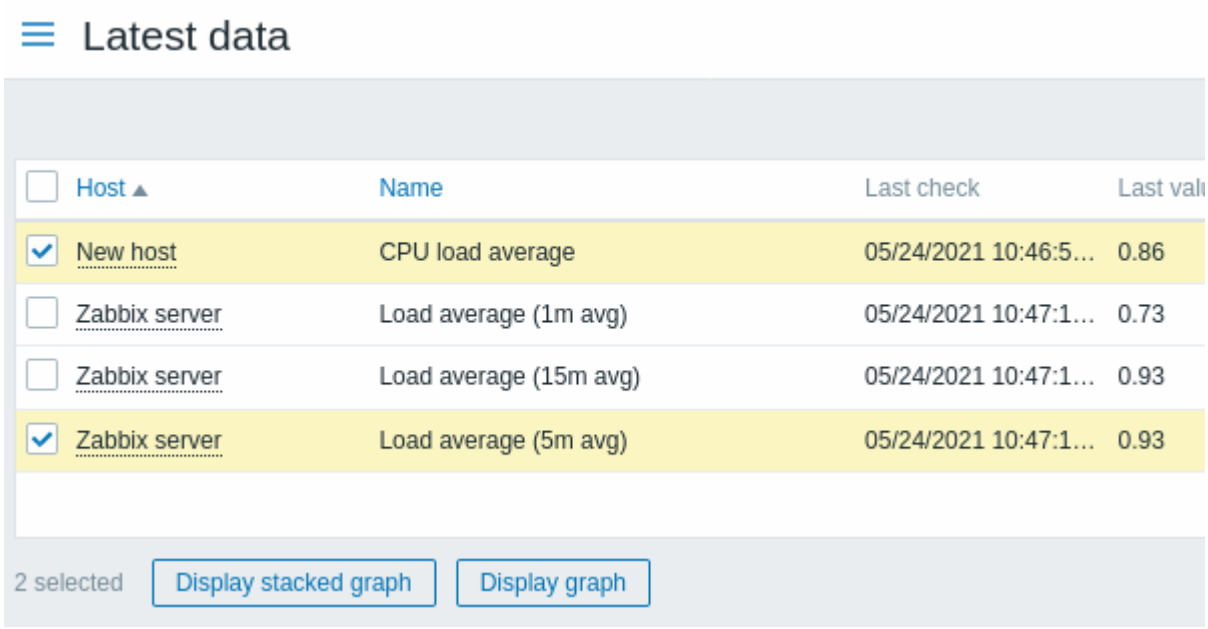
Alors qu'un graphique simple est utile pour afficher les données d'un seul élément et que les graphiques personnalisés offrent des options de personnalisation, aucun des deux ne permet de créer rapidement un graphique comparatif pour plusieurs éléments avec un minimum d'efforts et de maintenance.

Pour résoudre ce problème, il est possible de créer très rapidement des graphiques ad hoc pour plusieurs éléments.

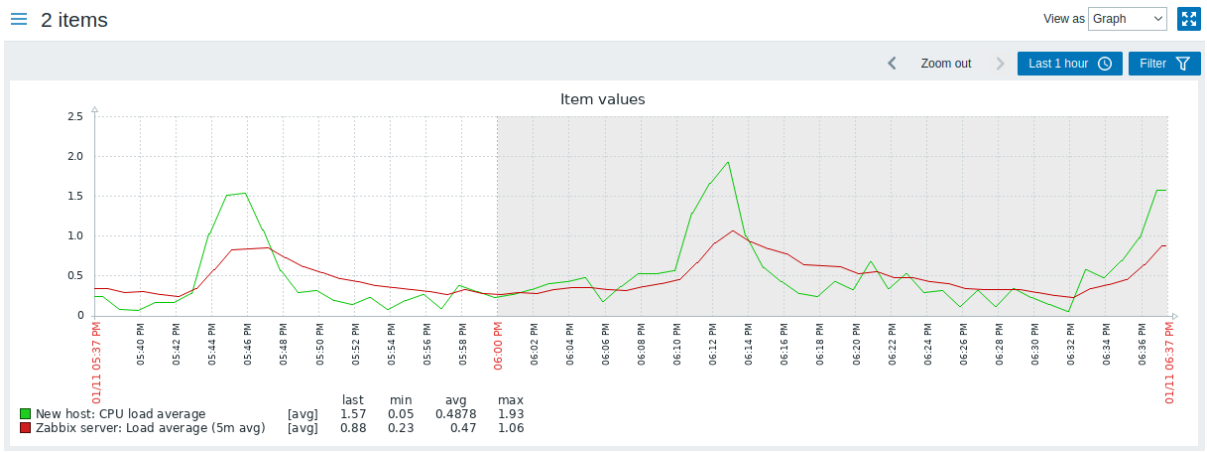
Configuration

Pour créer un graphique ad hoc, procédez comme suit :

1. Accédez à *Monitoring* > *Dernières données*.
2. Utilisez le filtre pour afficher les éléments souhaités.
3. Cochez les cases des éléments que vous souhaitez représenter sur le graphique.
4. Cliquez sur les boutons *Afficher le graphique empilé* ou *Afficher le graphique*.

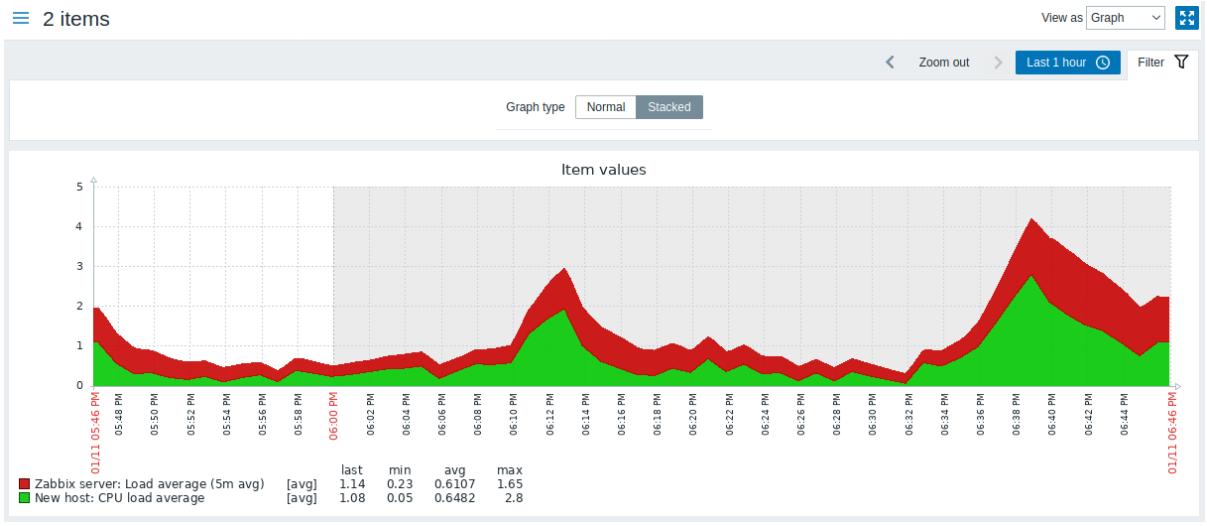


Votre graphique est créé instantanément :



Notez que, afin d'éviter l'affichage d'un trop grand nombre de lignes dans le graphique, seule la valeur moyenne de chaque élément est affichée (les lignes de valeur min/max ne sont pas affichées). Les déclencheurs et les informations de déclencheur ne sont pas affichés dans le graphique.

Dans la fenêtre du graphique créé, vous disposez du sélecteur de période et de la possibilité de passer du graphique en lignes "normal" à un graphique empilé (et inversement).



Cartes réseau

Aperçu

Si vous avez un réseau à gérer, vous voudrez peut-être avoir une vue d'ensemble de votre infrastructure quelque part. À cette fin, vous pouvez créer des cartes dans Zabbix - des réseaux et de tout ce que vous voulez.

Tous les utilisateurs peuvent créer des cartes réseau. Les cartes peuvent être publiques (disponibles pour tous les utilisateurs) ou privées (disponibles pour certains utilisateurs).

Passez à configurer une carte réseau.

Configuration d'une carte réseau

Vue d'ensemble

La configuration d'une carte dans Zabbix nécessite d'abord de créer une carte en définissant ses paramètres généraux, puis de commencer à remplir la carte réelle avec des éléments et leurs liens.

Vous pouvez remplir la carte avec des éléments qui sont un hôte, un groupe d'hôtes, un déclencheur, une image ou une autre carte.

Les icônes sont utilisées pour représenter les éléments de la carte. Vous pouvez définir les informations qui seront affichées avec les icônes et configurer l'affichage des problèmes récents d'une manière spéciale. Vous pouvez relier les icônes et définir les informations à afficher sur les liens.

Vous pouvez ajouter des URL personnalisées accessibles en cliquant sur les icônes. Ainsi, vous pouvez lier une icône d'hôte aux propriétés de l'hôte ou une icône de carte à une autre carte.

Le nombre de problèmes sur les cartes est affiché uniquement pour les problèmes causes.

Les cartes sont gérées dans *Monitoring* > *Maps*, où elles peuvent être configurées, gérées et consultées. Dans la vue de supervision, vous pouvez cliquer sur les icônes et profiter des liens vers certains scripts et URL.

Les cartes réseau sont basées sur des graphiques vectoriels (SVG).

Cartes publiques et privées

Tous les utilisateurs de Zabbix (y compris les utilisateurs non administrateurs) peuvent créer des cartes réseau. Les cartes ont un propriétaire : l'utilisateur qui les a créées. Les cartes peuvent être publiques ou privées.

- Les cartes *publiques* sont visibles par tous les utilisateurs, bien que, pour les voir, l'utilisateur doive avoir un accès en lecture à au moins un élément de la carte. Les cartes publiques peuvent être modifiées si un utilisateur/groupe d'utilisateurs dispose des autorisations de lecture-écriture pour cette carte et d'au moins des autorisations de lecture sur tous les éléments de la carte correspondante, y compris les déclencheurs dans les liens.
- Les cartes *privées* ne sont visibles que par leur propriétaire et les utilisateurs/groupes d'utilisateurs avec lesquels la carte est partagée par le propriétaire. Les utilisateurs ordinaires (non Super admin) ne peuvent partager qu'avec les groupes auxquels ils appartiennent et les utilisateurs qui appartiennent à ces groupes. Les utilisateurs de niveau Admin peuvent voir les cartes privées, qu'ils en soient propriétaires ou qu'ils figurent ou non dans la liste des utilisateurs avec lesquels elles sont partagées. Les cartes privées peuvent être modifiées par le propriétaire de la carte et si un utilisateur/groupe

d'utilisateurs dispose des autorisations de lecture-écriture pour cette carte et d'au moins des autorisations de lecture sur tous les éléments de la carte correspondante, y compris les déclencheurs dans les liens.

Les éléments de carte pour lesquels l'utilisateur ne dispose pas d'une autorisation de lecture sont affichés avec une icône grisée et toutes les informations textuelles sur l'élément sont masquées. Cependant, l'étiquette du déclencheur reste visible même si l'utilisateur n'a aucune autorisation sur le déclencheur.

Pour ajouter un élément à la carte, l'utilisateur doit également disposer d'au moins une autorisation de lecture sur cet élément.

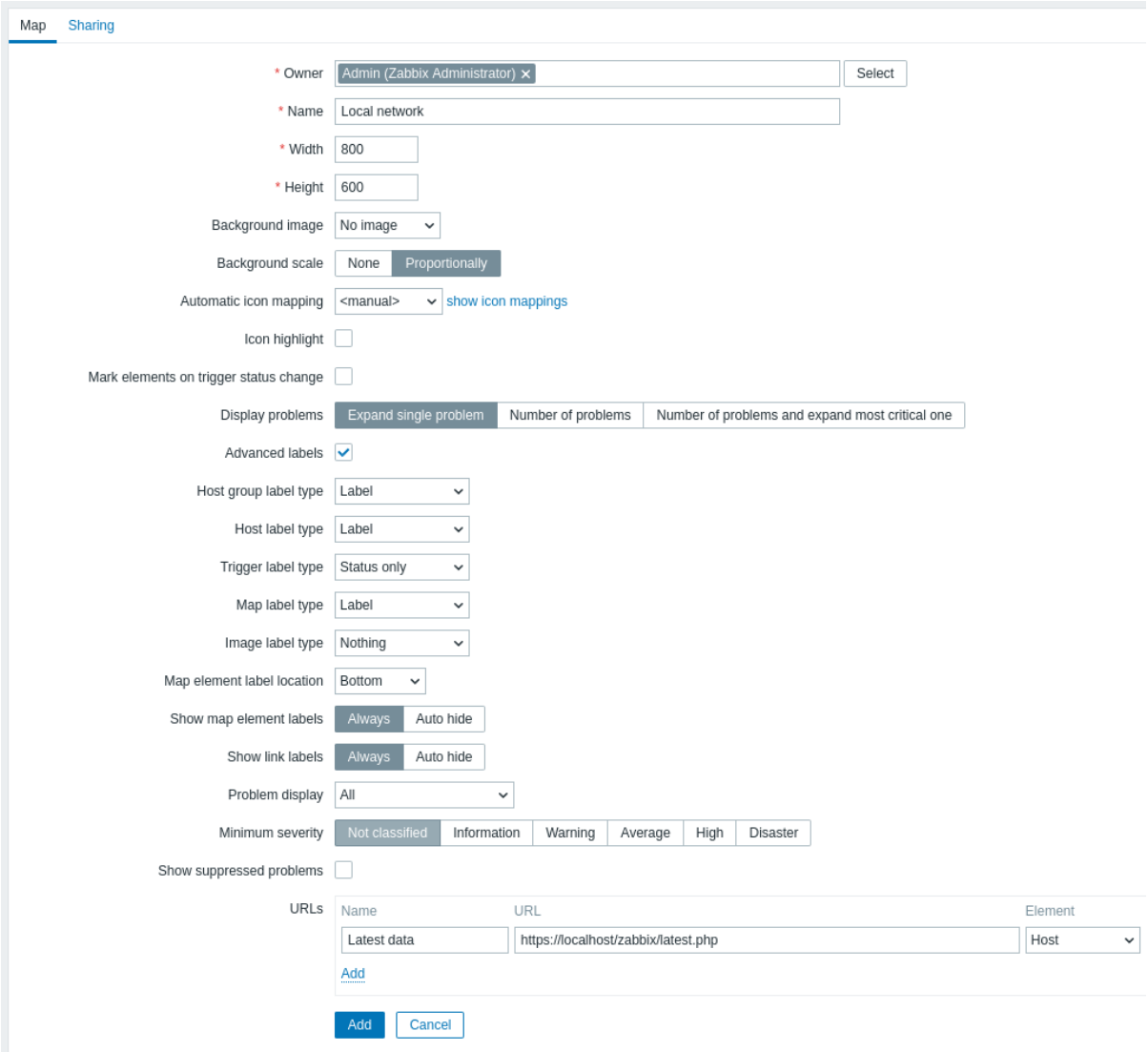
Création d'une carte

Pour créer une carte, procédez comme suit :

1. Accédez à *Monitoring* > *Maps*.
2. Accédez à la vue contenant toutes les cartes.
3. Cliquez sur *Create map*.
4. Configurez les détails de la carte.

Vous pouvez également utiliser le bouton *Clone* dans le formulaire de configuration d'une carte existante pour créer une nouvelle carte. Cette carte possédera toutes les propriétés de la carte existante, notamment les attributs généraux de mise en page, ainsi que les éléments de la carte existante.

L'onglet **Map** contient les attributs généraux de la carte :



Tous les champs de saisie obligatoires sont marqués d'un astérisque rouge.

Attributs généraux de la carte :

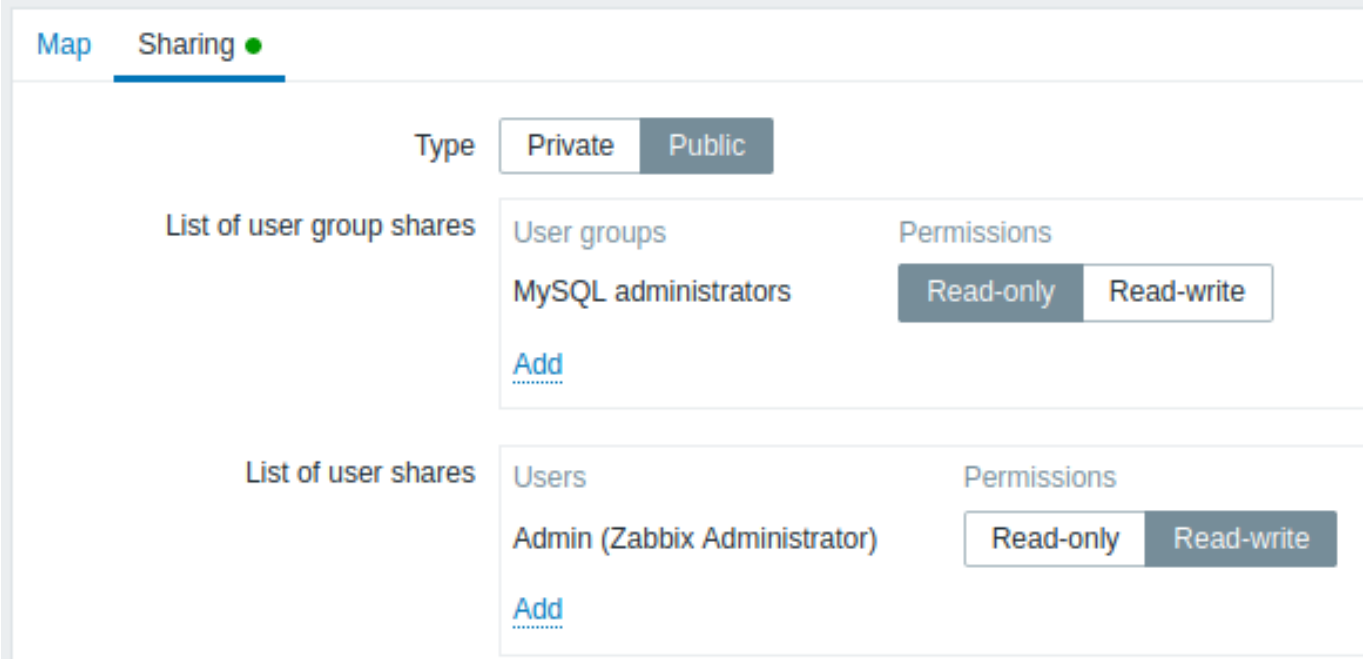
Paramètre	Description
Owner	Nom du propriétaire de la carte.
Name	Nom unique de la carte.
Width	Largeur de la carte en pixels.

Paramètre	Description
Height	Hauteur de la carte en pixels.
Background image	Utiliser une image d'arrière-plan : **No image** - aucune image d'arrière-plan (arrière-plan blanc) **Image** - image sélectionnée à utiliser comme image d'arrière-plan. Vous pouvez utiliser une carte géographique ou toute autre image pour améliorer votre carte.
Background scale	Utiliser la mise à l'échelle de l'arrière-plan : **None** - aucune mise à l'échelle ; **Proportionally** - couvrir tout l'arrière-plan de la carte, tout en conservant les proportions de l'image (par défaut).
Automatic icon mapping	Vous pouvez choisir d'utiliser une correspondance automatique des icônes, configurée dans *Administration > General > Icon mapping*. La correspondance des icônes permet d'associer certaines icônes à certains champs d'inventaire des hôtes.
Icon highlight	Si vous cochez cette case, les éléments de la carte seront mis en évidence. Les éléments associés à un déclencheur actif recevront un arrière-plan circulaire, de la même couleur que le déclencheur présentant la sévérité la plus élevée. De plus, une ligne verte épaisse sera affichée autour du cercle si tous les problèmes sont reconnus. Les éléments dont l'état est « désactivé » ou « en maintenance » recevront respectivement un arrière-plan carré gris ou orange. Voir également : Affichage des cartes
Mark elements on trigger status change	Une modification récente de l'état d'un déclencheur (problème récent ou résolution) sera mise en évidence par des marqueurs (triangles rouges pointant vers l'intérieur) sur les trois côtés de l'icône de l'élément qui ne sont pas occupés par le libellé. Les marqueurs sont affichés pendant 30 minutes.
Display problems	Sélectionnez la manière dont les problèmes sont affichés avec un élément de carte : **Expand single problem** - s'il n'y a qu'un seul problème, son nom est affiché. Sinon, le nombre total de problèmes est affiché. **Number of problems** - le nombre total de problèmes est affiché **Number of problems and expand most critical one** - le nom du problème le plus critique et le nombre total de problèmes sont affichés. Le problème « le plus critique » est déterminé en fonction de la sévérité du problème et, en cas d'égalité, de l'ID de l'événement du problème (l'ID le plus élevé, ou le problème le plus récent, est affiché en premier). Pour un *élément de carte de déclencheur*, il est déterminé en fonction de la sévérité du problème et, en cas d'égalité, de la position du déclencheur dans la liste des déclencheurs. En cas de plusieurs problèmes associés au même déclencheur, le plus récent est affiché.
Advanced labels	Si vous cochez cette case, vous pourrez définir des types de libellés distincts pour les différents types d'éléments.
Host group label type	Type de libellé utilisé pour l'élément de carte : **Label** - libellé de l'élément de carte **IP address** - adresse IP **Element name** - nom de l'élément (par exemple, nom de l'hôte) **Status only** - état uniquement (OK ou PROBLEM) **Nothing** - aucun libellé n'est affiché
Host label type *Trigger label type* *Map label type* *Image label type*	
Map element label location	Emplacement du libellé par rapport à l'élément de carte : **Bottom** - sous l'élément de carte **Left** - à gauche **Right** - à droite **Top** - au-dessus de l'élément de carte
Show map element labels	Sélectionnez la manière d'afficher les libellés des éléments de carte : **Always** - toujours affichés (par défaut) ; **Auto hide** - masquer le libellé lorsqu'il n'est pas survolé ou sélectionné.
Show link labels	Sélectionnez la manière d'afficher les libellés des liens : **Always** - toujours affichés (par défaut) ; **Auto hide** - masquer le libellé lorsqu'il n'est pas survolé ou sélectionné.

Paramètre	Description
Problem display	Afficher le nombre de problèmes comme suit : **All** - le nombre total de problèmes est affiché **Separated** - le nombre de problèmes non reconnus est affiché séparément, sous la forme d'un nombre parmi le nombre total de problèmes **Unacknowledged only** - seul le nombre de problèmes non reconnus est affiché
Minimum severity	Les problèmes dont la sévérité est inférieure au niveau minimal sélectionné ne seront pas affichés sur la carte. Par exemple, si *Warning* est sélectionné, les modifications associées à des déclencheurs de niveau *Information* et *Not classified* ne seront pas reflétées sur la carte.
Show suppressed problems	Cochez cette case pour afficher les problèmes qui seraient autrement supprimés (non affichés) en raison de la maintenance de l'hôte.
URLs	Les URL de chaque type d'élément peuvent être définies ici, où : **Name** - un libellé pour l'URL ; **URL** - une URL (jusqu'à 2048 caractères) ; **Element** - le type d'élément (par exemple, *Host*, *Host group*, etc.). Elles seront affichées sous forme de liens lorsqu'un utilisateur cliquera sur l'élément en mode d'affichage de la carte. Des macros peuvent être utilisées dans les noms et les valeurs des URL de carte. Pour obtenir la liste complète, consultez les macros prises en charge et recherchez « map URL names and values ».

Partage

L'onglet **Partage** contient le type de carte ainsi que les options de partage (groupes d'utilisateurs, utilisateurs) pour les cartes privées :



Parameter	Description
Type	Sélectionnez le type de carte : **Private** - la carte est visible uniquement pour les groupes d'utilisateurs et les utilisateurs sélectionnés **Public** - la carte est visible pour tous
List of user group shares	Sélectionnez les groupes d'utilisateurs qui peuvent accéder à la carte. Vous pouvez autoriser un accès en lecture seule ou en lecture-écriture.
List of user shares	Sélectionnez les utilisateurs qui peuvent accéder à la carte. Vous pouvez autoriser un accès en lecture seule ou en lecture-écriture.

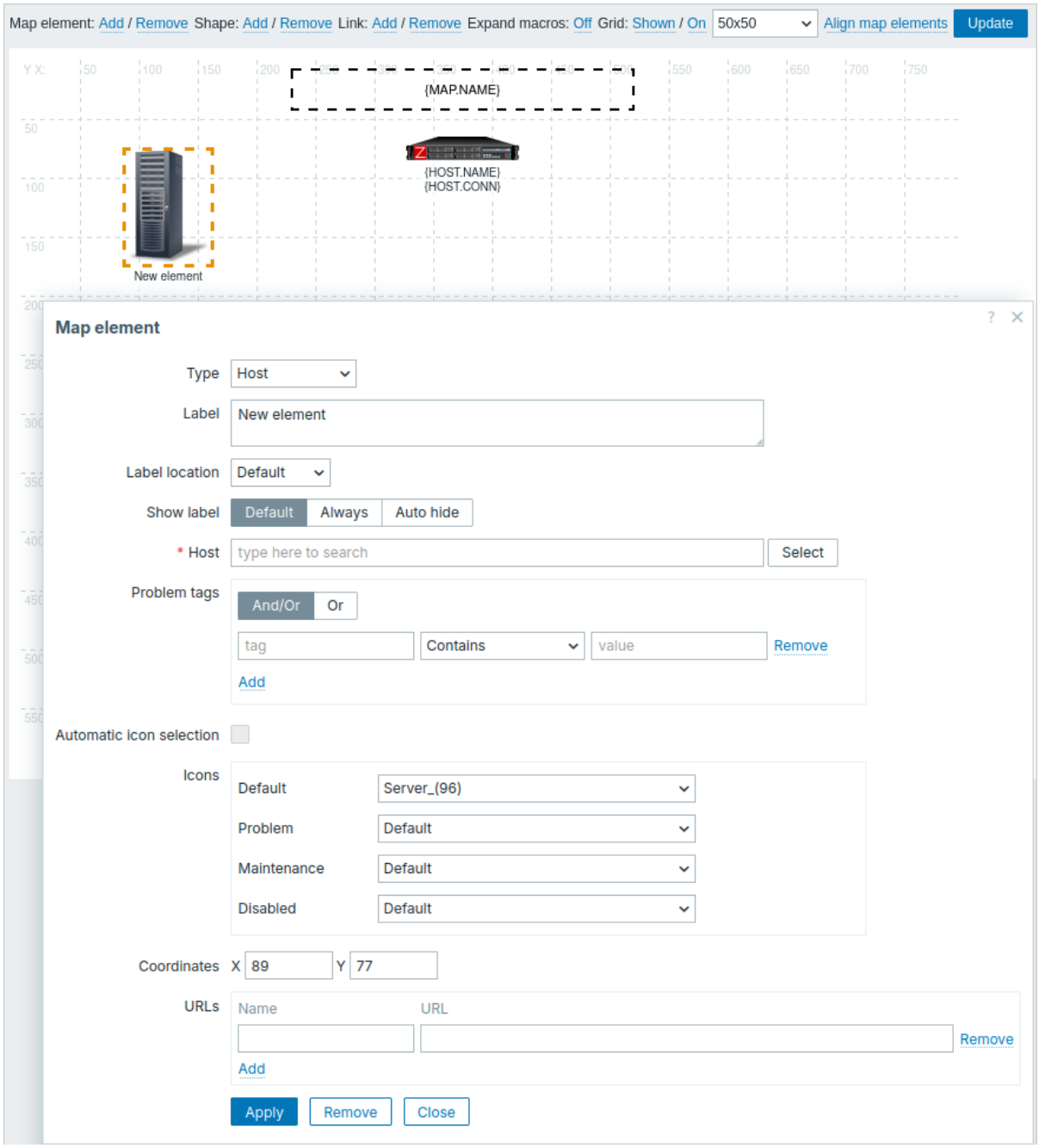
Lorsque vous cliquez sur *Add* pour enregistrer cette carte, vous avez créé une carte vide avec un nom, des dimensions et certaines préférences. Vous devez maintenant ajouter des éléments. Pour cela, cliquez sur *Edit* dans la liste des cartes pour ouvrir la zone modifiable.

Ajout d'éléments

Pour ajouter un élément, cliquez sur *Add* à côté de `Map element`. Le nouvel élément apparaîtra dans le coin supérieur gauche de la carte. Faites-le glisser et déposez-le où vous le souhaitez.

Notez qu'avec l'option Grid sur "On", les éléments s'aligneront toujours sur la grille (vous pouvez choisir différentes tailles de grille dans la liste déroulante, ainsi qu'afficher ou masquer la grille). Si vous souhaitez placer des éléments n'importe où sans alignement, réglez l'option sur "Off". (Vous pouvez aligner plus tard des éléments placés aléatoirement sur la grille en cliquant sur *Align map elements*.)

Maintenant que vous avez placé quelques éléments, vous souhaiterez peut-être commencer à les différencier en leur donnant des noms, etc. En cliquant sur l'élément, un formulaire s'affiche et vous pouvez définir le type d'élément, lui donner un nom, choisir une icône différente, etc.



Attributs de l'élément de carte :

Parameter	Description
Type	Type de l'élément : **Host** - icône représentant l'état de tous les déclencheurs de l'hôte sélectionné **Map** - icône représentant l'état de tous les éléments d'une carte **Trigger** - icône représentant l'état d'un ou plusieurs déclencheurs **Host group** - icône représentant l'état de tous les déclencheurs de tous les hôtes appartenant au groupe sélectionné **Image** - une icône, non liée à une ressource
Label	Libellé de l'icône, toute chaîne de caractères. Les macros et les chaînes multilignes peuvent être utilisées. Les macros d'expression sont prises en charge dans ce champ, mais uniquement avec les fonctions `avg`, `last`, `min` et `max`, avec le temps comme paramètre (par exemple, `{?avg(/host/key,1h)}`). Pour une liste complète des macros prises en charge, consultez supported macros et recherchez 'map element labels'.
Label location	Emplacement du libellé par rapport à l'icône : **Default** - emplacement par défaut du libellé de la carte ; **Bottom** - sous l'icône ; **Left** - à gauche ; **Right** - à droite ; **Top** - au-dessus de l'icône.
Show label	Sélectionnez la manière d'afficher le libellé de l'élément : **Default** - utilise les paramètres *Show map element labels* de la configuration de la carte (par défaut) ; **Always** - toujours affiché (par défaut) ; **Auto hide** - masque le libellé s'il n'est ni survolé ni sélectionné.
Host	Sélectionnez un hôte ; vous pouvez aussi saisir le nom de l'hôte. Ce champ prend en charge l'autocomplétion, donc commencer à taper le nom d'un hôte affichera une liste déroulante des hôtes correspondants. Faites défiler vers le bas pour en sélectionner un. Cliquez sur l'icône de suppression (x) pour retirer la sélection. Ce champ est disponible uniquement pour les éléments de type *Host*.
Map	Sélectionnez une carte ; vous pouvez aussi saisir le nom de la carte. Ce champ prend en charge l'autocomplétion, donc commencer à taper le nom d'une carte affichera une liste déroulante des cartes correspondantes. Faites défiler vers le bas pour en sélectionner une. Cliquez sur l'icône de suppression (x) pour retirer la sélection. Ce champ est disponible uniquement pour les éléments de type *Map*.
Triggers	Sélectionnez un ou plusieurs déclencheurs dans le champ *New triggers* ci-dessous, puis cliquez sur *Add*. Vous pouvez aussi commencer à saisir le nom d'un déclencheur et en sélectionner un dans la liste déroulante des déclencheurs correspondants. Cliquez sur l'icône de suppression (x) pour retirer la sélection. Ce champ est disponible uniquement pour les éléments de type *Trigger*. L'ordre des déclencheurs sélectionnés peut être modifié, mais uniquement au sein d'une même gravité de déclencheurs. La sélection de plusieurs déclencheurs affecte également la résolution des macros {HOST.*} en mode édition et en mode affichage. En mode édition, les premières macros {HOST.*} affichées seront résolues en fonction du premier déclencheur de la liste (selon la gravité du déclencheur). Le mode affichage dépend du paramètre Display problems dans les attributs généraux de la carte : - Si le mode *Expand single problem* est choisi, les premières macros {HOST.*} affichées seront résolues en fonction du dernier déclencheur de problème détecté (quelle que soit la gravité) ou du premier déclencheur de la liste (si aucun problème n'est détecté). - Si le mode *Number of problems and expand most critical one* est choisi, les premières macros {HOST.*} affichées seront résolues en fonction de la gravité du déclencheur.
Host group	Sélectionnez un groupe d'hôtes ; vous pouvez aussi saisir le groupe d'hôtes. Ce champ prend en charge l'autocomplétion, donc commencer à taper le nom d'un groupe d'hôtes affichera une liste déroulante des groupes d'hôtes correspondants. Faites défiler vers le bas pour en sélectionner un. Cliquez sur l'icône de suppression (x) pour retirer la sélection. Ce champ est disponible uniquement pour les éléments de type *Host group*.

Parameter	Description
Problem tags	Spécifiez des tags pour limiter le nombre de problèmes affichés dans le widget. Il est possible d'inclure ou d'exclure des tags et des valeurs de tags spécifiques. Plusieurs conditions peuvent être définies. La correspondance du nom du tag est toujours sensible à la casse. Plusieurs opérateurs sont disponibles pour chaque condition : **Exists** - inclure les noms de tags spécifiés **Equals** - inclure les noms et valeurs de tags spécifiés (sensible à la casse) **Contains** - inclure les noms de tags spécifiés dont les valeurs contiennent la chaîne saisie (correspondance de sous-chaîne, insensible à la casse) **Does not exist** - exclure les noms de tags spécifiés **Does not equal** - exclure les noms et valeurs de tags spécifiés (sensible à la casse) **Does not contain** - exclure les noms de tags spécifiés dont les valeurs contiennent la chaîne saisie (correspondance de sous-chaîne, insensible à la casse) Il existe deux types de calcul pour les conditions : **And/Or** - toutes les conditions doivent être remplies, les conditions ayant le même nom de tag seront regroupées par la condition Or **Or** - il suffit qu'une condition soit remplie Ce champ est disponible uniquement pour les éléments de type *Host* et *Host group*.
Automatic icon selection	Lorsque la case est cochée, le mappage des icônes sera utilisé pour déterminer quelle icône afficher.
Icons	Vous pouvez choisir d'afficher différentes icônes pour l'élément dans les cas suivants : *Default*, *Problem*, *Maintenance* et *Disabled*. Ce champ est disponible uniquement pour les éléments de type *Host*.
Coordinates	Coordonnées **X** et **Y** de l'élément de carte.
URLs	Des URL spécifiques à l'élément peuvent être définies ici pour l'élément, où : **Name** - un libellé pour l'URL ; **URL** - une URL (jusqu'à 2048 caractères). Elles seront affichées sous forme de liens lorsqu'un utilisateur cliquera sur l'élément en mode affichage de la carte. Si l'élément possède ses propres URL et que des URL au niveau de la carte sont définies pour son type, elles seront combinées dans le même menu. Les macros peuvent être utilisées dans les noms et les valeurs des éléments de carte. Pour une liste complète, consultez supported macros et recherchez 'map URL names and values'.

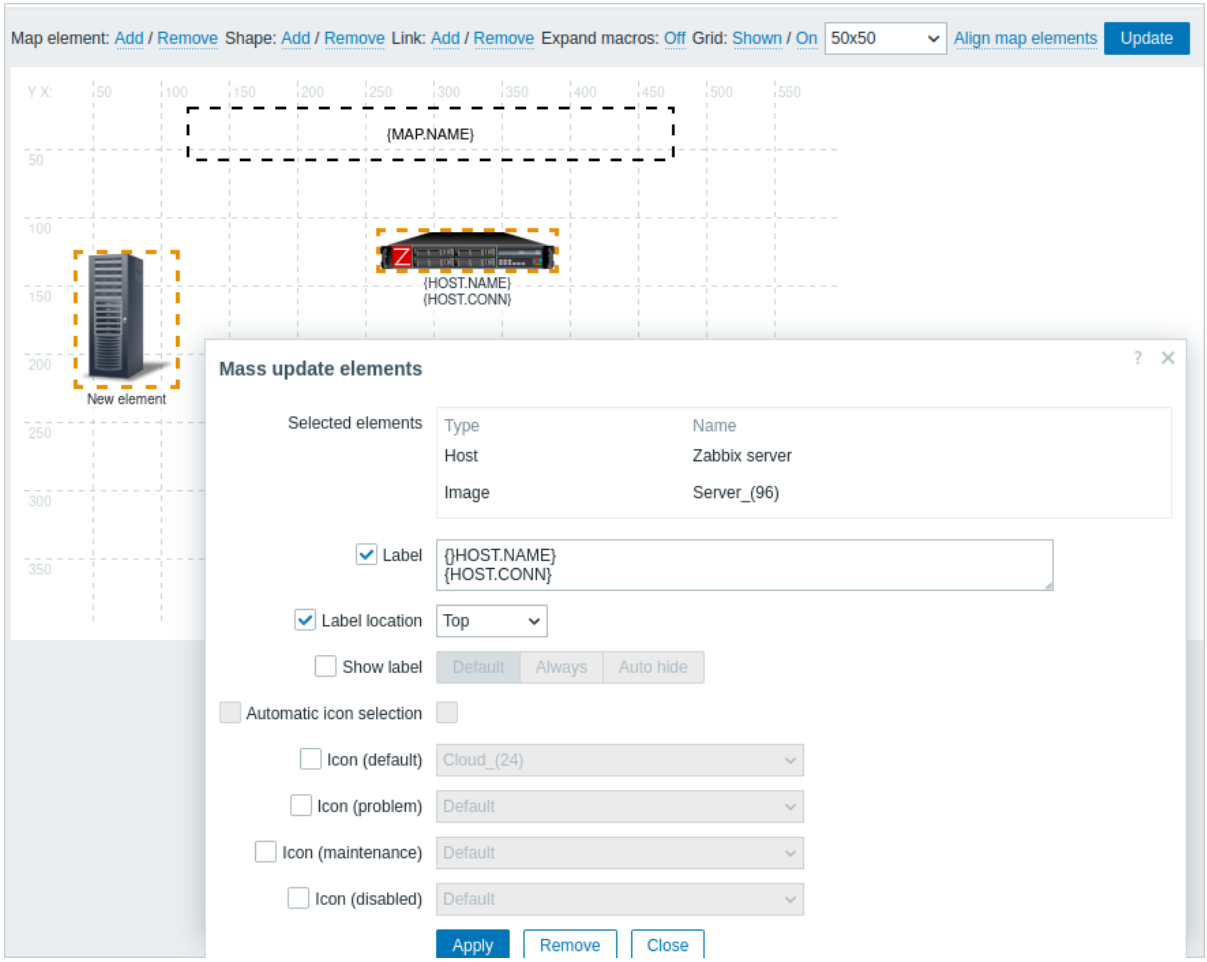
Attention:
Les éléments ajoutés ne sont pas enregistrés automatiquement. Si vous quittez la page, toutes les modifications peuvent être perdues.
Il est donc conseillé de cliquer sur le bouton **Update** dans le coin supérieur droit. Une fois cliqué, les modifications sont enregistrées, quel que soit ce que vous choisissez dans la fenêtre contextuelle suivante.
Les options de grille sélectionnées sont également enregistrées avec chaque carte.

Sélection d'éléments

Pour sélectionner des éléments, sélectionnez-en un, puis maintenez la touche *Ctrl* enfoncée pour sélectionner les autres.

Vous pouvez également sélectionner plusieurs éléments en faisant glisser un rectangle dans la zone modifiable et en sélectionnant tous les éléments qu'il contient.

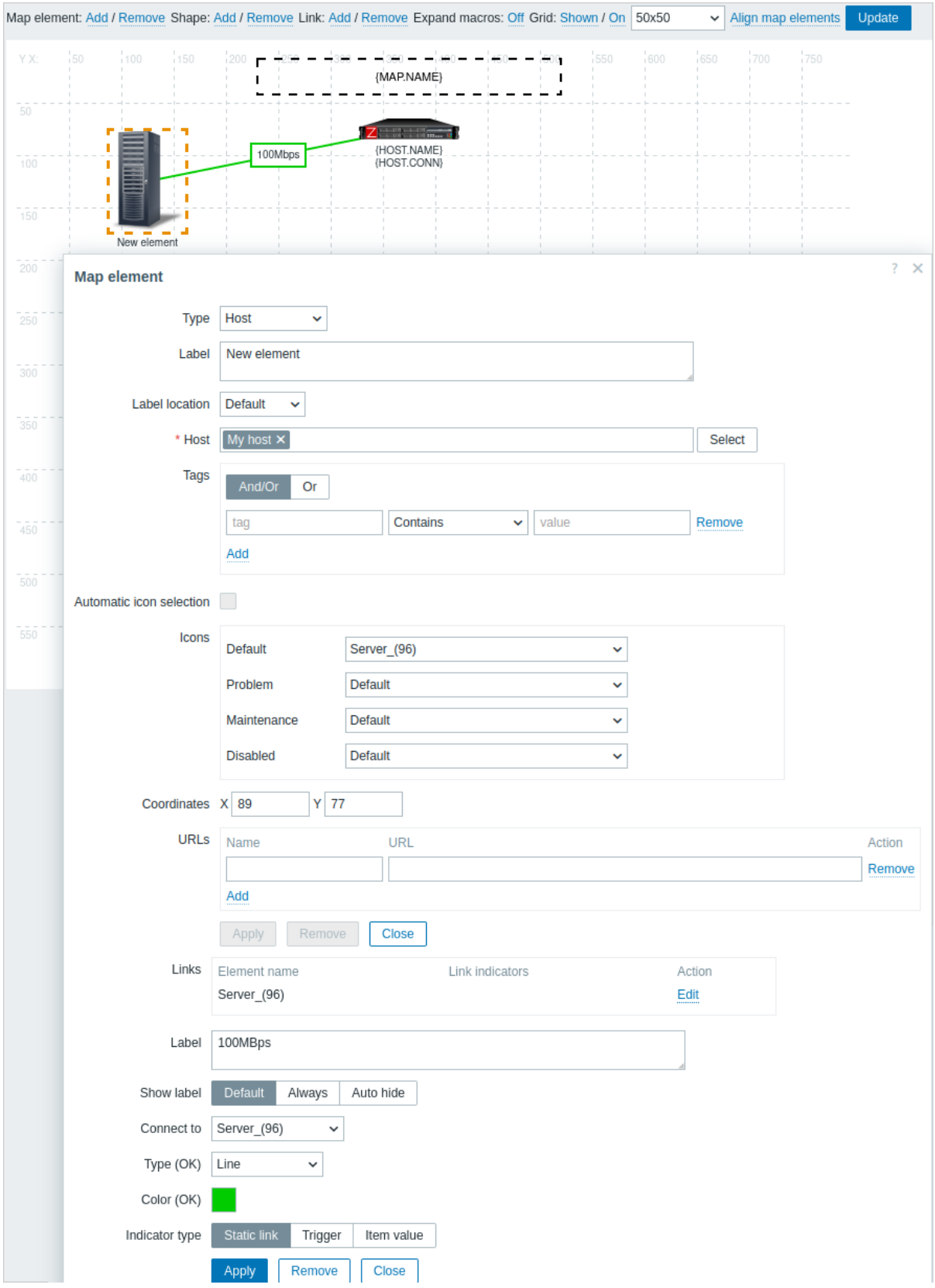
Une fois que vous avez sélectionné plusieurs éléments, le formulaire de propriété de l'élément passe en mode de mise à jour en masse afin que vous puissiez modifier les attributs des éléments sélectionnés en une seule fois. Pour ce faire, marquez l'attribut à l'aide de la case à cocher et entrez une nouvelle valeur pour celui-ci. Vous pouvez utiliser des macros (par exemple, {HOST.NAME} pour l'étiquette de l'élément).



Liaison des éléments

Une fois que vous avez placé des éléments sur la carte, il est temps de commencer à les lier. Pour lier deux éléments, vous devez d'abord les sélectionner. Une fois les éléments sélectionnés, cliquez sur *Ajouter* à côté de Lien.

Une fois un lien créé, le formulaire d'un élément unique contient désormais une section supplémentaire *Liens*. Cliquez sur *Modifier* pour modifier les attributs du lien.



Attributs du lien :

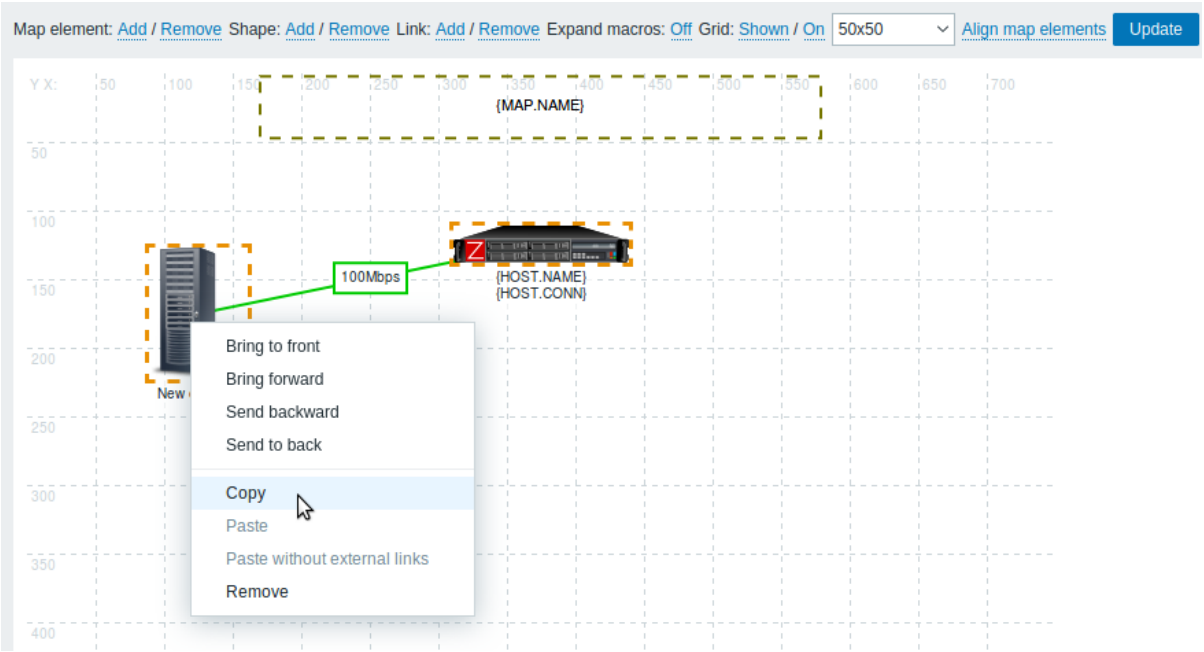
Parameter	Description
Label	Libellé qui sera affiché au-dessus du lien. Les expressions macros sont prises en charge dans ce champ, mais uniquement avec les fonctions `avg`, `last`, `min` et `max`, avec le temps comme paramètre (par exemple, `{?avg(/host/key,1h)}`).

Parameter	Description
Show label	Sélectionnez la manière d'afficher le libellé du lien : **Default** - utilise les paramètres *Afficher les libellés des éléments de la carte* de la configuration de la carte (par défaut) ; **Always** - toujours affiché (par défaut) ; **Auto hide** - masque le libellé s'il n'est ni survolé ni sélectionné.
Connect to	L'élément auquel le lien se connecte.
Type (OK)	Style de lien par défaut. Sélectionnez le style : *Line*, *Bold line*, *Dot* ou *Dashed line*.
Color (OK)	Couleur de lien par défaut. Cliquez sur l'échantillon de couleur pour sélectionner une autre couleur.
Indicator type	Sélectionnez le type d'indicateur du lien : **Static link** - aucun indicateur ; **Trigger** - permet des indicateurs de lien basés sur des déclencheurs ; **Item value** - permet des indicateurs de lien basés sur la valeur d'un élément.
Item	Sélectionnez l'élément. La valeur de cet élément influencera le style du lien. Ce champ est disponible lorsque *Item value* est sélectionné pour *Indicator type*.
Indicators	La liste des indicateurs de lien. Ce champ est disponible lorsque *Indicator type* est défini sur *Trigger* ou *Item value*. Si *Trigger* est sélectionné, des déclencheurs peuvent être attribués comme indicateurs de lien. Lorsqu'un déclencheur est dans l'état *Problem*, son style (couleur et type de ligne sélectionnés) est appliqué au lien. Si *Item value* est sélectionné et qu'un élément est choisi dans le champ *Item*, les valeurs d'élément peuvent être attribuées comme indicateurs de lien. Lorsque la valeur de l'élément atteint le seuil spécifié (pour un élément numérique) ou correspond au modèle (pour un élément texte), son style (couleur et type de ligne sélectionnés) est appliqué au lien.

Déplacement et copie-collage d'éléments

Plusieurs éléments sélectionnés peuvent être **déplacés** vers un autre emplacement sur la carte en cliquant sur l'un des éléments sélectionnés, en maintenant le bouton de la souris enfoncé, puis en déplaçant le curseur vers l'emplacement souhaité.

Un ou plusieurs éléments peuvent être **copiés** en sélectionnant les éléments, puis en cliquant avec le bouton droit de la souris sur un élément sélectionné et en choisissant *Copy* dans le menu.

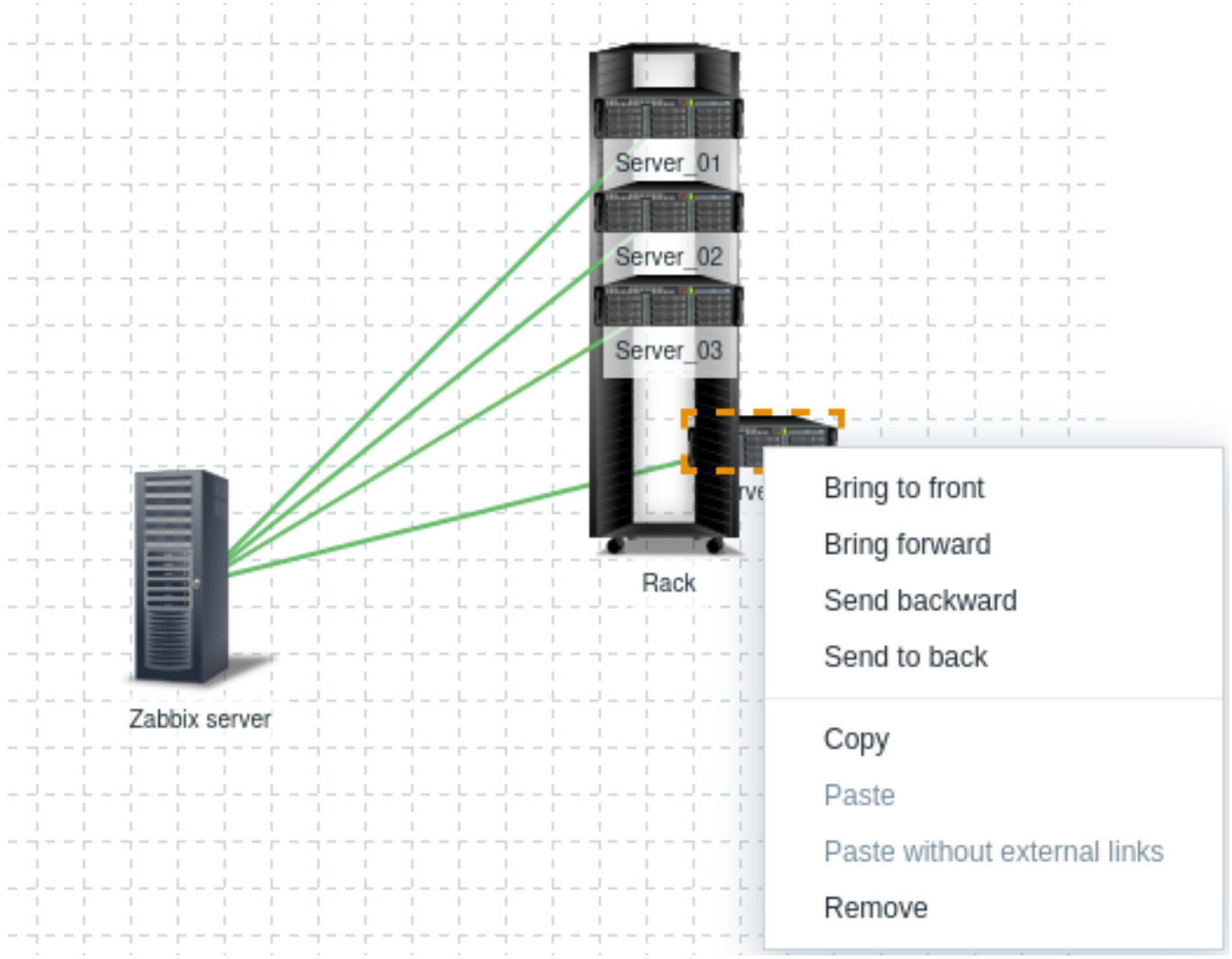


Pour coller les éléments, cliquez avec le bouton droit de la souris sur une zone de la carte et sélectionnez *Paste* dans le menu. L'option *Paste without external links* collera les éléments en conservant uniquement les liens entre les éléments sélectionnés.

Le copier-coller fonctionne dans la même fenêtre du navigateur. Les raccourcis clavier ne sont pas pris en charge.

Ordre des éléments

Pour placer un élément devant un autre (ou inversement), cliquez avec le bouton droit de la souris sur l'élément et sélectionnez *Avancer/Mettre au premier plan* ou *Reculer/Envoyer à l'arrière-plan*.

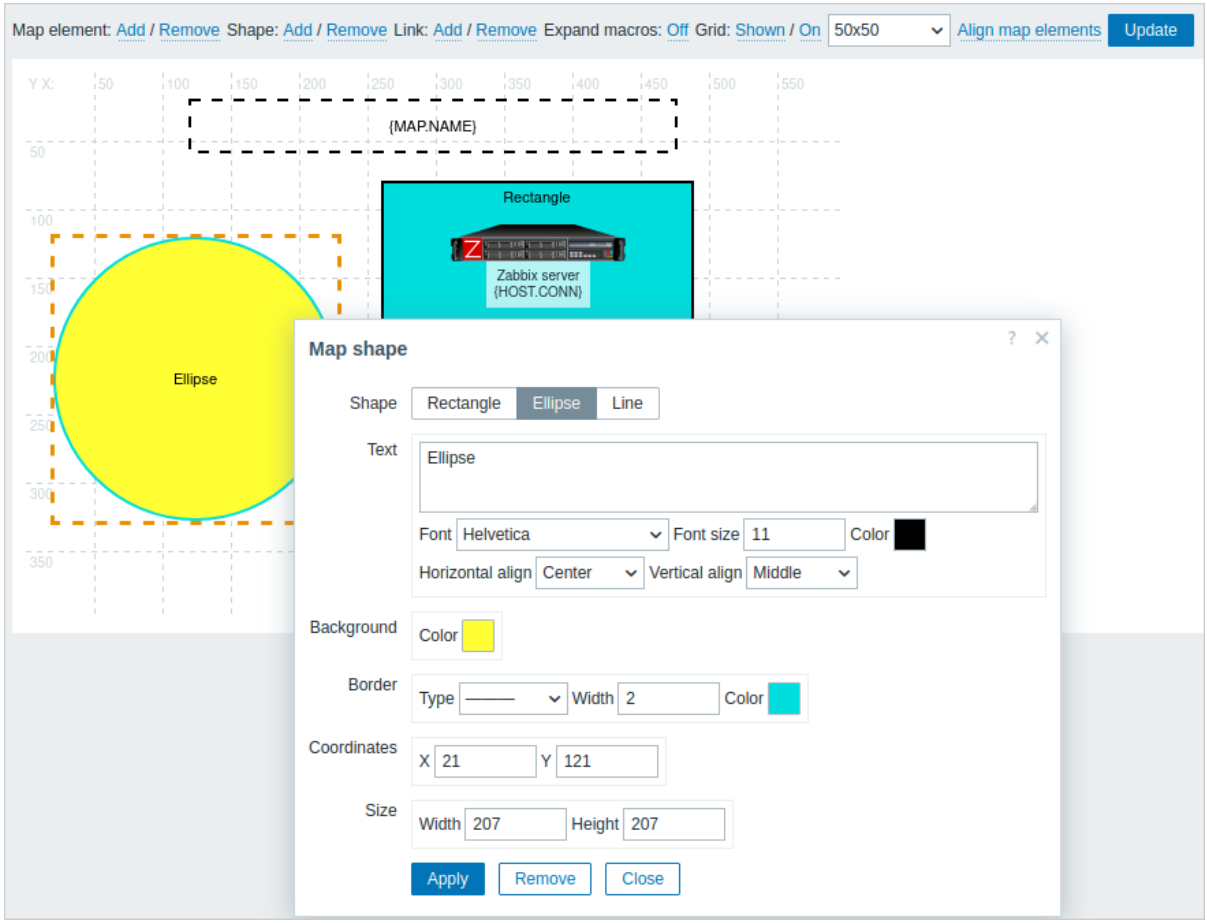


Ajout de formes

En plus des éléments de carte, il est également possible d'ajouter certaines formes. Les formes ne sont pas des éléments de carte ; elles ne sont qu'une représentation visuelle. Par exemple, une forme rectangulaire peut être utilisée comme arrière-plan pour regrouper certains hôtes. Des formes rectangulaires et elliptiques peuvent être ajoutées.

Pour ajouter une forme, cliquez sur *Ajouter* à côté de Forme. La nouvelle forme apparaîtra dans le coin supérieur gauche de la carte. Faites-la glisser et déposez-la où vous le souhaitez.

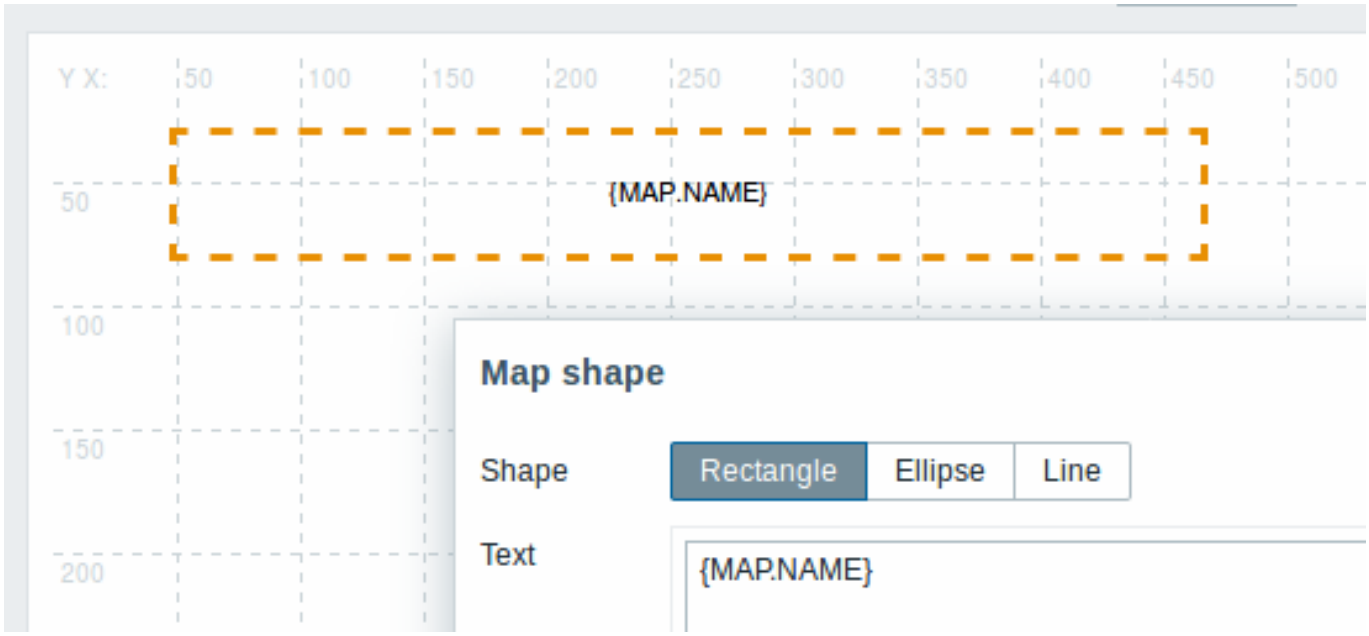
Une nouvelle forme est ajoutée avec des couleurs par défaut. En cliquant sur la forme, un formulaire s'affiche et vous pouvez personnaliser l'apparence de la forme, ajouter du texte, etc.



Pour sélectionner des formes, sélectionnez-en une, puis maintenez *Ctrl* enfoncé pour sélectionner les autres. Lorsque plusieurs formes sont sélectionnées, les propriétés communes peuvent être mises à jour en masse, de la même manière que pour les éléments.

Du texte peut être ajouté dans les formes. Les macros d'expression sont prises en charge dans le texte, mais uniquement avec les fonctions `avg`, `last`, `min` et `max`, avec le temps comme paramètre (par exemple, `{?avg(/host/key,1h)}`).

Pour afficher uniquement du texte, la forme peut être rendue invisible en supprimant la bordure de la forme (sélectionnez 'None' dans le champ *Border*). Par exemple, notez que la macro {MAP.NAME}, visible dans la capture d'écran ci-dessus, est en réalité une forme rectangulaire avec du texte, ce que l'on peut voir en cliquant sur la macro :



{MAP.NAME} est résolue en le nom de carte configuré lors de l'affichage de la carte.

Si des hyperliens sont utilisés dans le texte, ils deviennent cliquables lors de l'affichage de la carte.

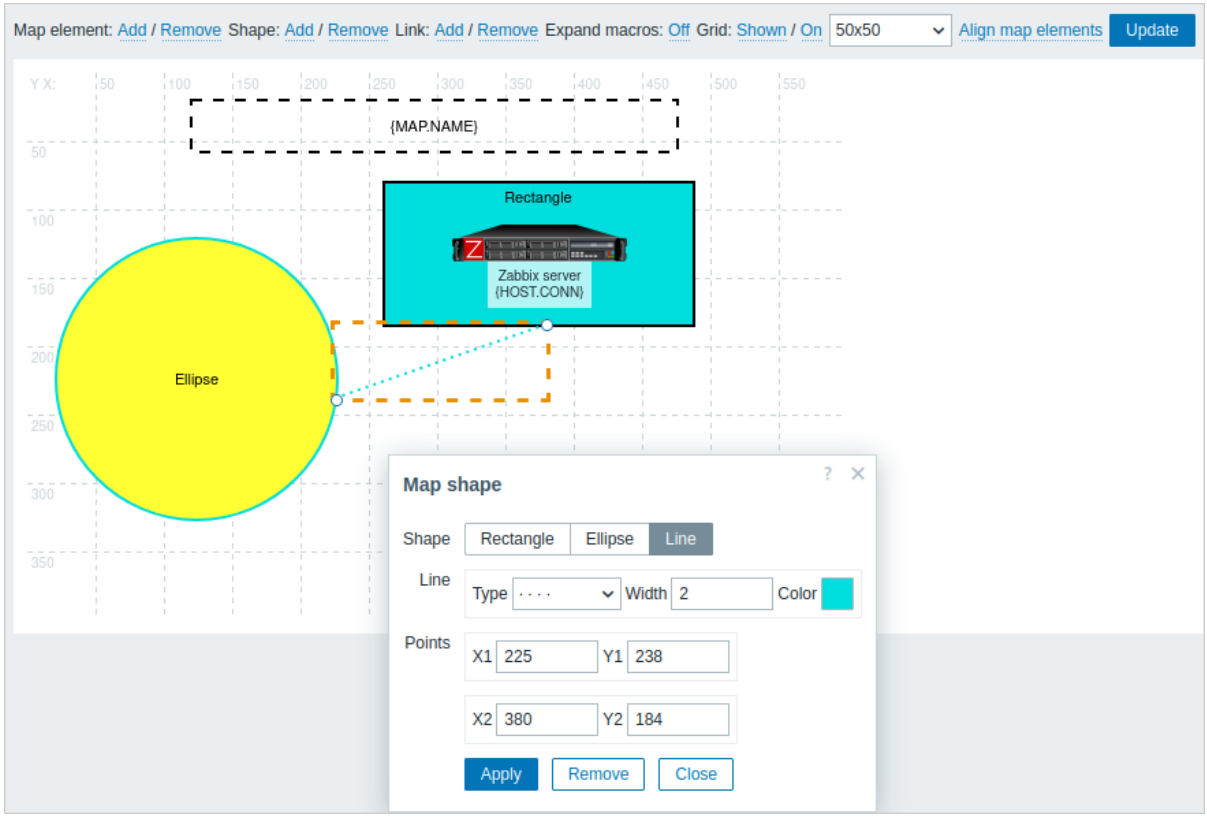
Le retour à la ligne du texte est toujours activé dans les formes. Cependant, dans une ellipse, les lignes sont renvoyées à la ligne

comme si l'ellipse était un rectangle. La coupure des mots n'est pas implémentée, donc les mots longs (les mots qui ne tiennent pas dans la forme) ne sont pas renvoyés à la ligne, mais sont masqués (sur la page d'édition de la carte) ou tronqués (sur les autres pages avec des cartes).

Ajout de lignes

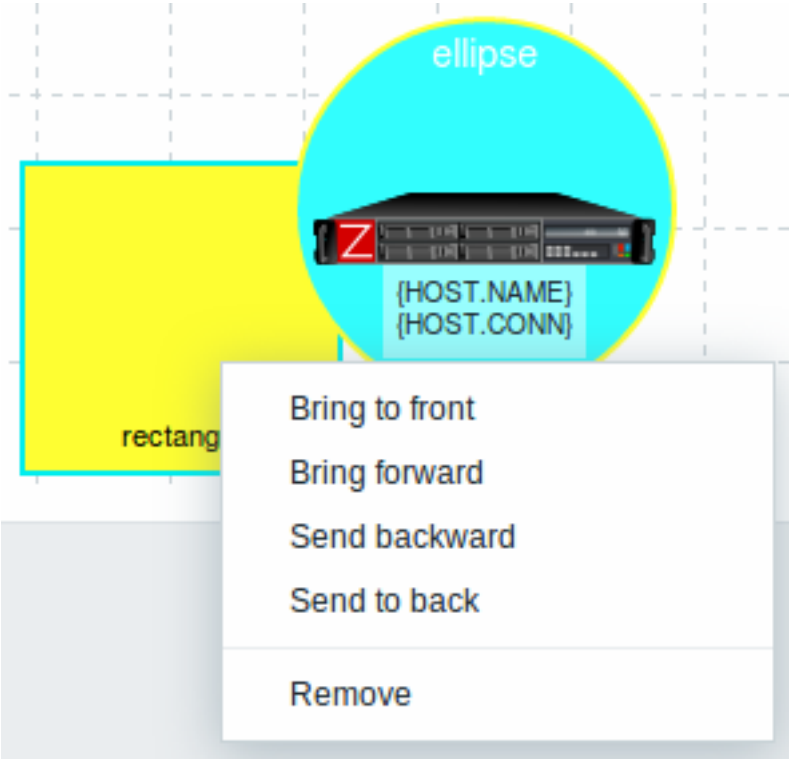
En plus des formes, il est également possible d'ajouter des lignes. Les lignes peuvent être utilisées pour relier des éléments ou des formes dans une carte.

Pour ajouter une ligne, cliquez sur *Add* à côté de Shape. Une nouvelle forme apparaîtra dans le coin supérieur gauche de la carte. Sélectionnez-la et cliquez sur *Line* dans le formulaire d'édition pour transformer la forme en ligne. Ajustez ensuite les propriétés de la ligne, telles que le type de ligne, la largeur, la couleur, etc.



Ordre des formes et des lignes

Pour placer une forme devant une autre (ou inversement), cliquez sur la forme avec le bouton droit de la souris afin d'afficher le menu des formes de la carte.

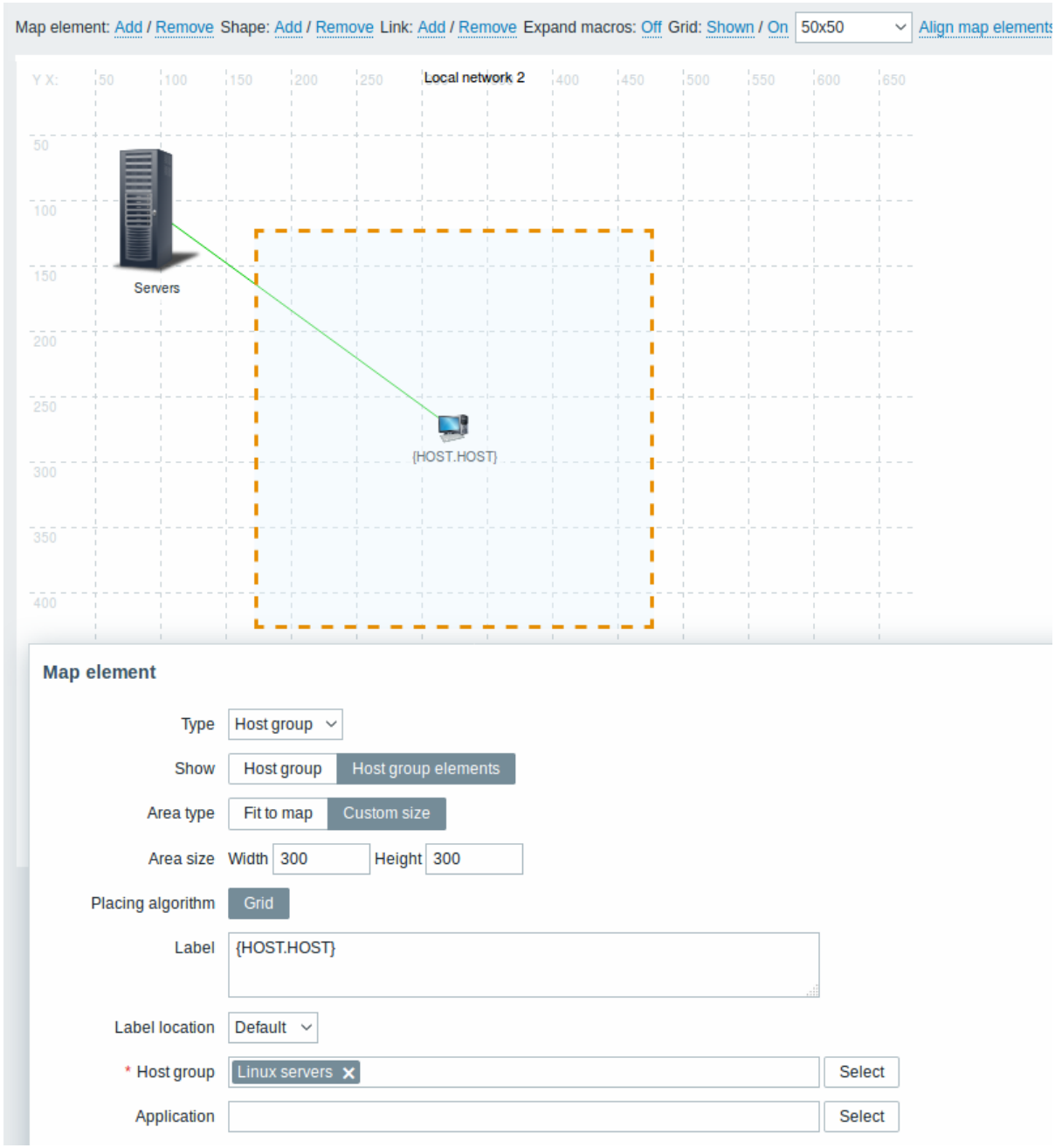


Éléments du groupe d’hôtes

Aperçu

Cette section explique comment ajouter un élément de type *Groupe d’hôtes* lors de la configuration d’une carte réseau.

Configuration



Tous les champs de saisie obligatoires sont marqués d'un astérisque rouge.

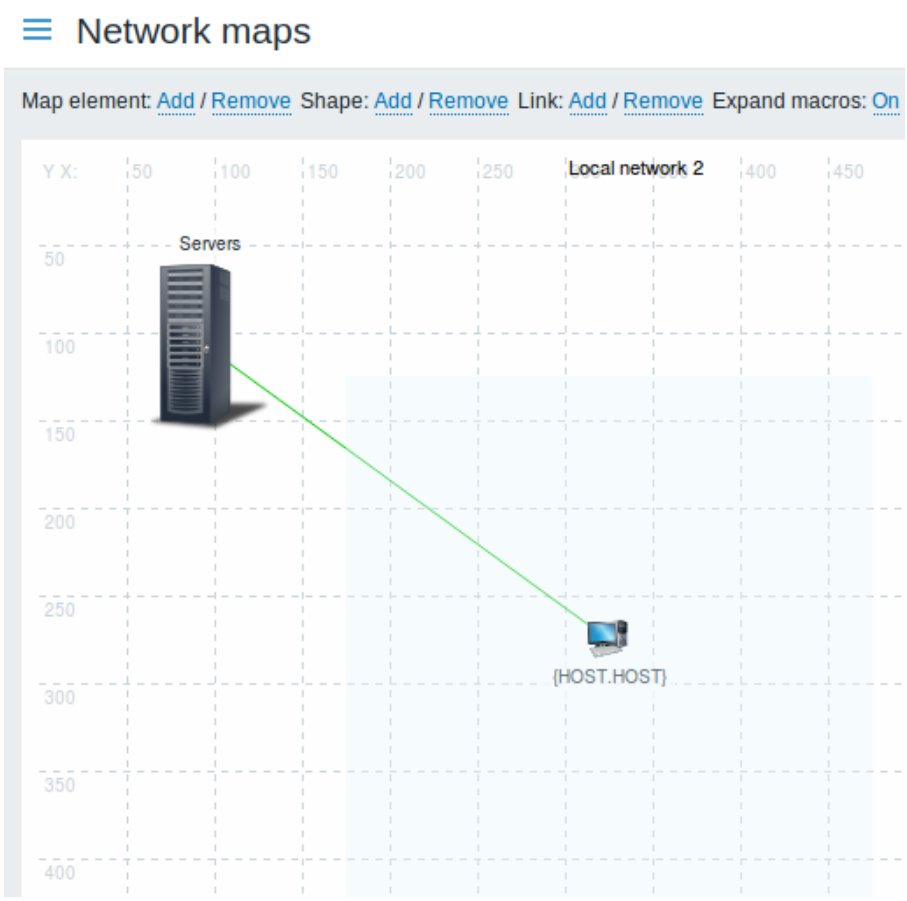
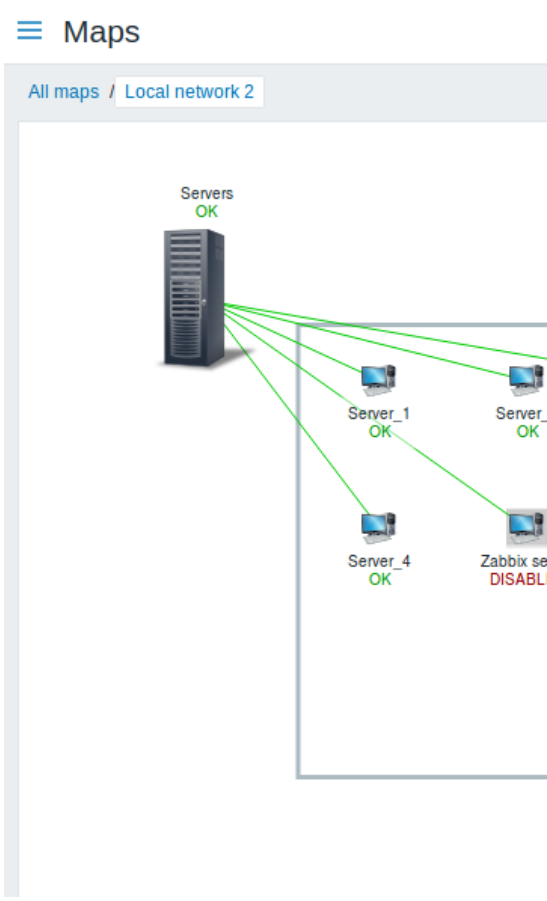
Ce tableau présente les paramètres propres au type d'élément *Groupe d'hôtes* :

Paramètre	Description
Type	Sélectionnez le type d'élément : **Groupe d'hôtes** - Icône représentant l'état de tous les déclencheurs de tous les hôtes appartenant au groupe sélectionné.
Afficher	Options d'affichage : **Groupe d'hôtes** - La sélection de cette option affiche une seule icône présentant les informations correspondantes sur le groupe d'hôtes concerné. **Éléments du groupe d'hôtes** - La sélection de cette option affiche plusieurs icônes présentant les informations correspondantes sur chaque élément (hôte) du groupe d'hôtes concerné.
Type de zone	Ce paramètre est disponible si *Afficher* est défini sur *Éléments du groupe d'hôtes* : **Adapter à la carte** - Tous les éléments du groupe d'hôtes sont répartis uniformément dans la carte. **Taille personnalisée** - Configuration manuelle de la zone de la carte destinée à l'affichage de tous les éléments du groupe d'hôtes.
Taille de la zone	Ce paramètre est disponible si *Type de zone* est défini sur *Taille personnalisée* : **Largeur** - Valeur en pixels indiquant la largeur de la zone de la carte. **Hauteur** - Valeur en pixels indiquant la hauteur de la zone de la carte.

Paramètre	Description
Algorithme de placement	**Grille** - Seule option disponible pour afficher tous les éléments du groupe d'hôtes.
Libellé	Libellé de l'icône, sous forme de chaîne quelconque. Les macros et les chaînes multilignes peuvent être utilisées dans les libellés. Si le type d'élément de la carte est *Groupe d'hôtes*, la spécification de certaines macros a un impact sur l'affichage de la carte, qui présente les informations correspondantes sur chaque hôte. Par exemple, si la macro `{HOST.IP}` est utilisée, la vue de modification de la carte affiche uniquement la macro `{HOST.IP}` elle-même, tandis que la vue de la carte inclut et affiche l'adresse IP unique de chaque hôte.

Affichage des éléments d'un groupe d'hôtes

Cette option est disponible si l'option d'affichage *Éléments du groupe d'hôtes* est sélectionnée.
Lorsque vous sélectionnez *Éléments du groupe d'hôtes* comme option *Afficher*, vous ne voyez d'abord qu'une seule icône pour le groupe d'hôtes.
Cependant, après avoir enregistré la carte et accédé à sa vue, vous constaterez que la carte inclut tous les éléments (hôtes) du groupe d'hôtes concerné :

Vue de modification de la carte	Vue de la carte
	

Notez l'utilisation de la macro `{HOST.HOST}`.
Lors de la modification de la carte, le nom de la macro n'est pas résolu, tandis que dans la vue de la carte, tous les noms uniques des hôtes sont affichés.

Indicateurs de lien

Vue d'ensemble

Vous pouvez attribuer des indicateurs à un lien entre des éléments dans une carte réseau.

Les indicateurs peuvent être basés sur des déclencheurs ou des valeurs d'élément. Il est possible d'afficher un style et une couleur de lien différents :

- lorsque des déclencheurs passent à l'état de problème ;
- lorsque la valeur d'un élément :
  - atteint un seuil (pour les éléments numériques) ;
  - correspond à une expression régulière (pour les éléments texte).

Lorsque vous configurez un lien, vous définissez le type et la couleur de lien par défaut. En attribuant des indicateurs à un lien, il devient possible de faire dépendre le style et la couleur du lien de l'état du déclencheur ou de la valeur de l'élément.

Par exemple, si l'un des déclencheurs attribués passe à l'état de problème, le style et la couleur du lien changeront pour le refléter. Ainsi, votre lien par défaut était peut-être une ligne verte. Désormais, avec le déclencheur à l'état de problème, votre lien peut devenir rouge et en gras (si vous l'avez défini ainsi).

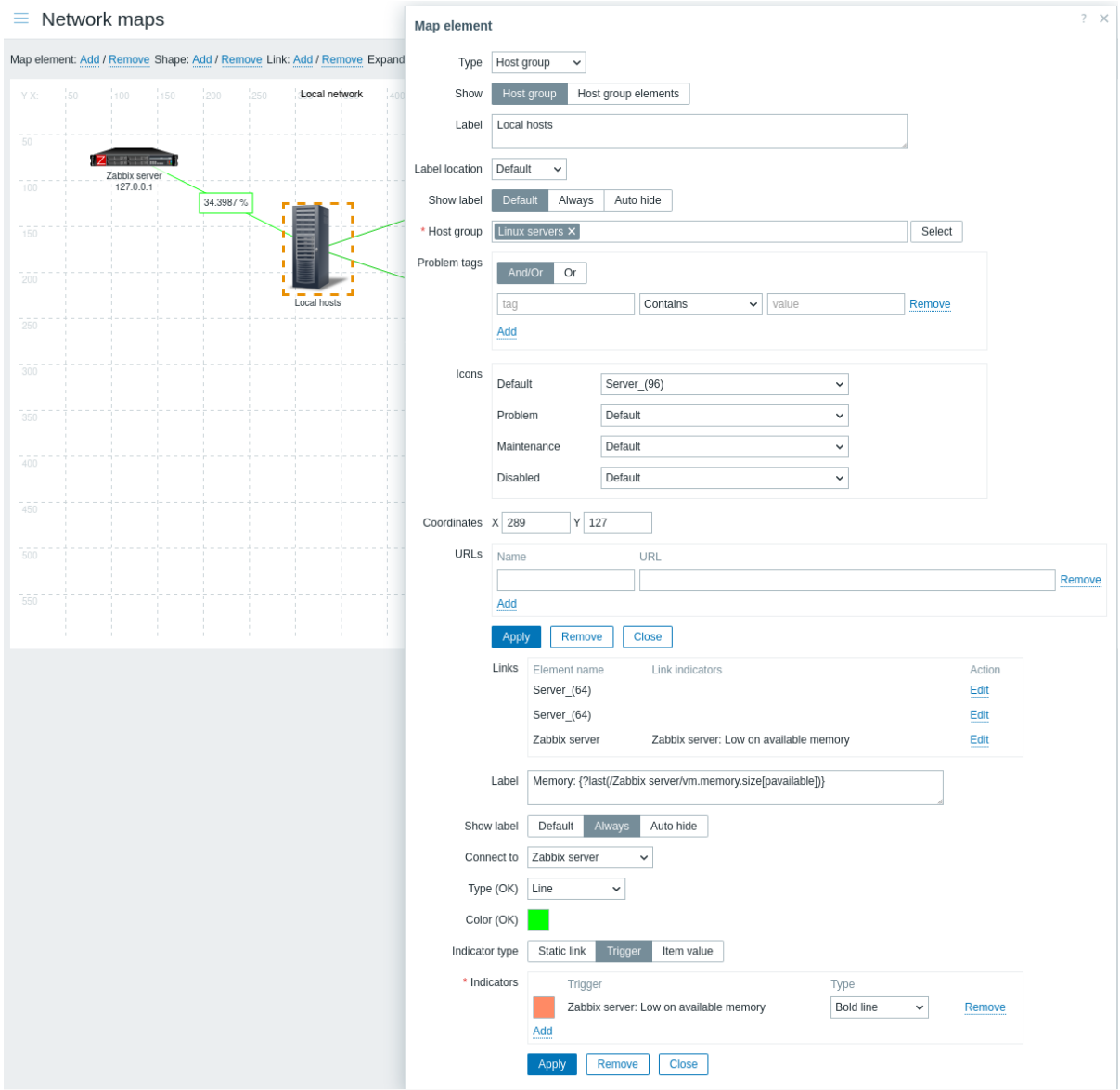
De même, si la valeur d'un élément atteint un seuil spécifié ou correspond à une expression régulière spécifiée, le style du lien peut le refléter.

Configuration

Déclencheurs

Pour attribuer des déclencheurs comme indicateurs de lien, procédez comme suit :

- sélectionnez un élément de carte
- cliquez sur *Edit* dans la section *Links* pour le lien approprié
- sélectionnez *Trigger* comme type d'indicateur
- cliquez sur *Add* dans le bloc *Indicators* et sélectionnez un ou plusieurs déclencheurs

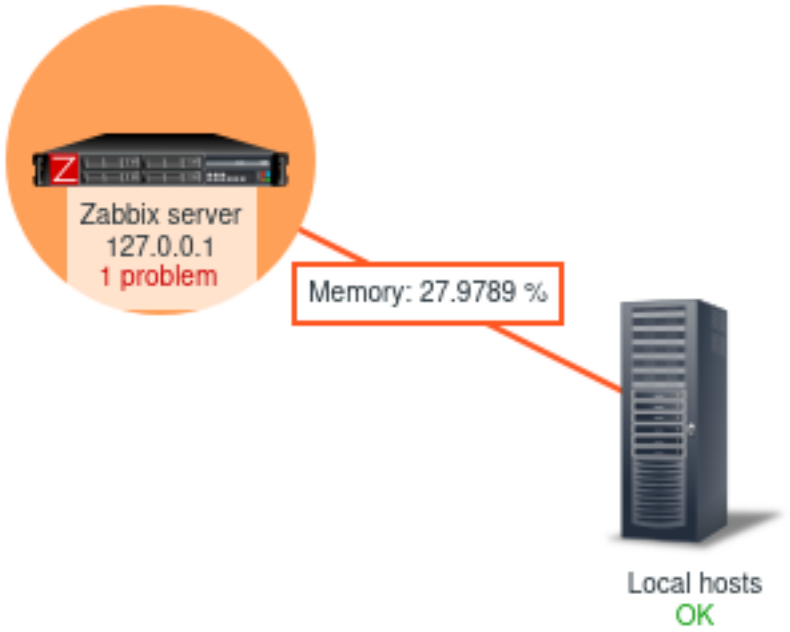


Tous les champs de saisie obligatoires sont marqués d'un astérisque rouge.

Les déclencheurs ajoutés peuvent être consultés dans la liste *Indicators*.

Vous pouvez définir le type de lien et la couleur pour chaque indicateur directement depuis la liste. Une fois terminé, cliquez sur *Apply*, fermez le formulaire et cliquez sur *Update* pour enregistrer les modifications de la carte.

Dans *Monitoring* > *Maps*, le type de lien et la couleur correspondants seront affichés si le **trigger** passe à l'état de problème.



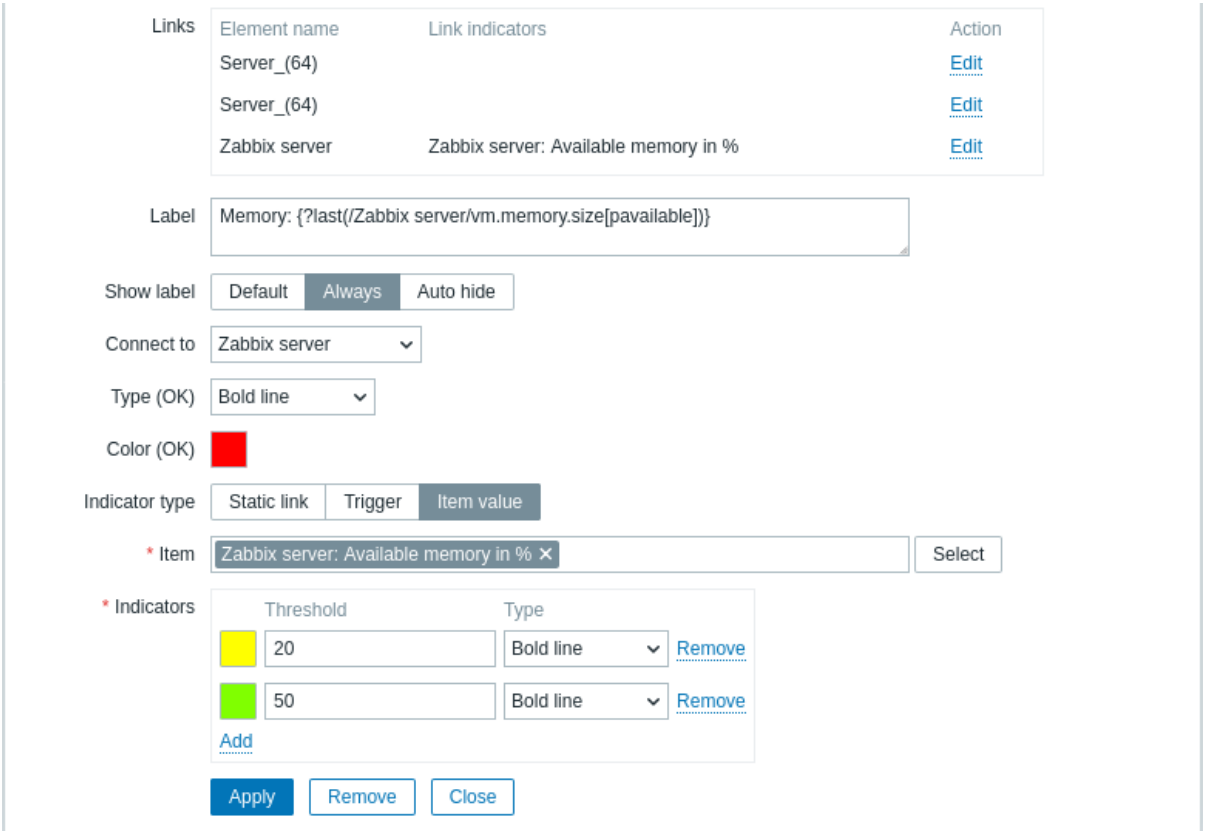
Note:
Si plusieurs déclencheurs passent à l'état de problème, le problème ayant la sévérité la plus élevée déterminera le style et la couleur du lien. Si plusieurs déclencheurs de même sévérité sont attribués au même lien de carte, celui ayant l'ID le plus faible est prioritaire. Notez également que :

1. Les paramètres *Minimum trigger severity* et *Show suppressed problem* de la configuration de la carte influencent les problèmes pris en compte.
2. Dans le cas de déclencheurs avec plusieurs problèmes (génération de plusieurs problèmes), chaque problème peut avoir une sévérité différente de celle du déclencheur (modifiée manuellement), peut avoir des tags différents (en raison des macros) et peut être supprimé.

Valeurs d'élément

Pour attribuer des valeurs d'élément comme indicateurs de lien, procédez comme suit :

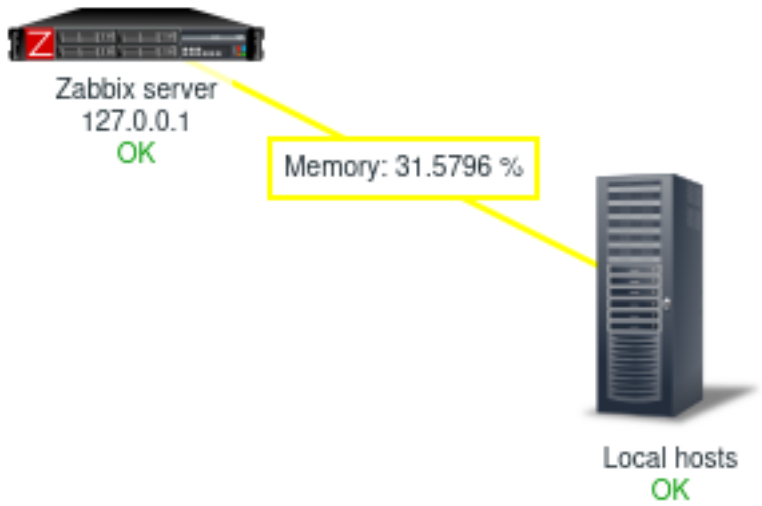
- sélectionnez un élément de carte
- cliquez sur *Edit* dans la section *Links* pour le lien approprié
- sélectionnez *Item value* comme type d'indicateur
- sélectionnez l'élément
- ajoutez un ou plusieurs seuils ou motifs de valeur d'élément dans le bloc *Indicators*



Les seuils/motifs d'élément ajoutés peuvent être consultés dans la liste *Indicators*.

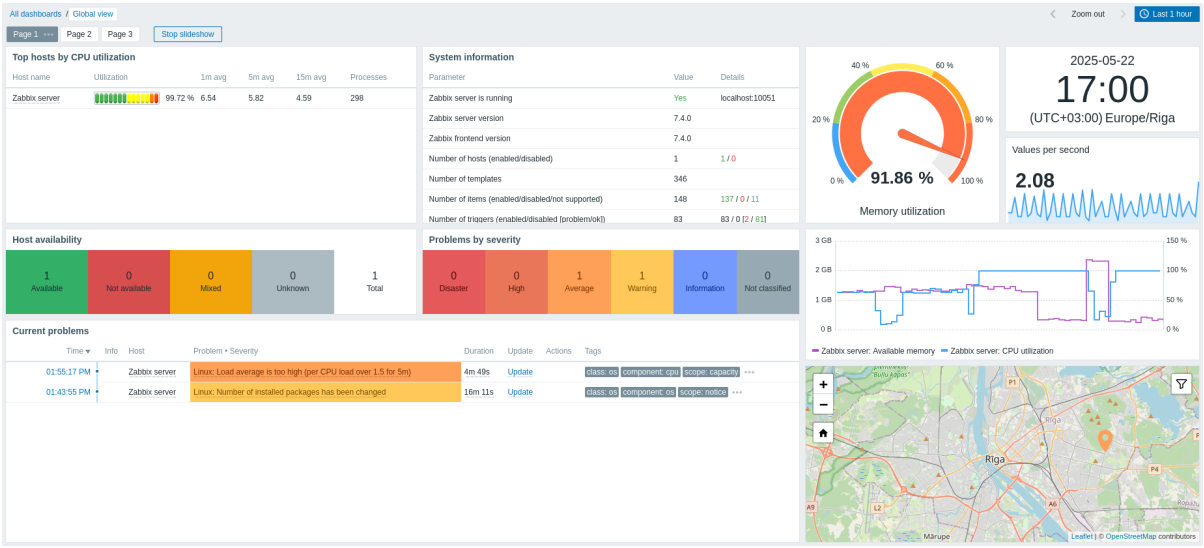
Vous pouvez définir le type de lien et la couleur pour chaque indicateur directement depuis la liste. Une fois terminé, cliquez sur *Apply*, fermez le formulaire et cliquez sur *Update* pour enregistrer les modifications de la carte.

Dans *Monitoring* > *Maps*, le type de lien et la couleur correspondants seront affichés si la **valeur d'élément** atteint le seuil défini (pour les types de données numériques) ou correspond au motif d'expression régulière (pour les types de données textuelles).



Tableaux de bord

Les tableaux de bord - à la fois les tableaux de bord globaux et les tableaux de bord d'hôte - offrent une puissante plateforme de visualisation avec des widgets et des outils tels que des graphiques modernes, des cartes, des diaporamas, et bien plus encore.



Modèles et groupes de modèles

Vue d'ensemble

L'utilisation de modèles est un excellent moyen de réduire sa charge de travail et de rationaliser la configuration de Zabbix. Un modèle est un ensemble d'entités qui peut être appliqué de manière pratique à plusieurs hôtes.

Les entités peuvent être :

- éléments
- déclencheurs
- graphiques
- tableaux de bord
- règles de découverte de bas niveau
- scénarios web

Comme, dans la pratique, de nombreux hôtes sont identiques ou assez similaires, il est naturel que l'ensemble des entités (éléments, déclencheurs, graphiques, ...) que vous avez créées pour un hôte puisse être utile pour beaucoup d'autres. Bien sûr, vous pourriez les copier vers chaque nouvel hôte, mais cela représenterait beaucoup de travail manuel. À la place, avec les modèles, vous pouvez les copier dans un modèle, puis appliquer ce modèle à autant d'hôtes que nécessaire.

Lorsqu'un modèle est lié à un hôte, toutes les entités (éléments, déclencheurs, graphiques, ...) du modèle sont ajoutées à l'hôte. Les modèles sont attribués directement à chaque hôte individuel (et non à un groupe d'hôtes).

Les modèles sont souvent utilisés pour regrouper des entités pour des services ou applications particuliers (comme Apache, MySQL, PostgreSQL, Postfix...) puis appliqués aux hôtes exécutant ces services.

Un autre avantage de l'utilisation des modèles apparaît lorsqu'il faut modifier quelque chose pour tous les hôtes. Une modification effectuée une seule fois au niveau du modèle sera propagée à tous les hôtes liés.

Les modèles sont organisés en groupes de modèles.

Passez à la création et la configuration d'un modèle.

Modèles prêts à l'emploi

Vue d'ensemble

Zabbix fournit un ensemble croissant de modèles préconfigurés afin de simplifier et d'accélérer la configuration des cibles de supervision.

Tous les modèles prêts à l'emploi sont disponibles dans *Collecte de données* > *Modèles*.

Lorsque vous mettez Zabbix à niveau, les modèles ne sont pas mis à niveau automatiquement. Cela évite d'écraser les modifications personnalisées que vous avez apportées. Pour mettre à niveau ou ajouter des modèles manuellement, consultez la section *Mise à niveau des modèles* ci-dessous.

Veuillez utiliser la barre latérale pour accéder aux informations relatives aux types de modèles spécifiques et aux exigences de fonctionnement.

Les tableaux de bord globaux conçus pour fonctionner avec les modèles Zabbix existants peuvent être téléchargés depuis le dépôt de tableaux de bord et importés dans votre environnement Zabbix. Une description détaillée de chaque tableau de bord, y compris les prérequis, est disponible dans le fichier `README` du tableau de bord.

Voir également :

- Lier un modèle
- Problèmes connus liés aux modèles

Mise à niveau des modèles

Pour mettre à niveau ou ajouter un **seul modèle** :

1. Ouvrez le répertoire `templates` dans le dépôt Git de Zabbix.

2. Choisissez la branche (sous l'étiquette *Source*) correspondant à votre version de Zabbix (par exemple, *master* pour la version en développement).

3. Recherchez le modèle que vous souhaitez ajouter ou mettre à niveau (par exemple, MySQL by Zabbix agent).

4. Copiez le contenu du fichier de modèle (par exemple, `template_db_mysql_agent.yaml`) et enregistrez-le dans un fichier local.

5. Dans l'interface utilisateur de Zabbix, accédez à *Collecte de données* > *Modèles* et importez le fichier local.

> **Note:**
> Les types de média webhook peuvent être mis à niveau de manière similaire : récupérez-les dans le répertoire `media`, puis importez-les dans *Alertes* > *Types de média*.

Pour mettre à niveau ou ajouter **plusieurs modèles** :

1. Téléchargez l'archive des ensembles de modèles correspondant à votre serveur web :

- Nginx - `templates_bundles-8.0-1M-nginx.zip`
- Apache - `templates_bundles-8.0-2M-apache.zip`

2. Extrayez l'archive. Elle contient des fichiers YAML, chacun comprenant plusieurs modèles, ainsi qu'un fichier d'index répertoriant les modèles présents dans chaque fichier YAML.

3. Dans l'interface utilisateur de Zabbix, accédez à *Collecte de données* > *Modèles* et importez les fichiers YAML un par un.

> **Note:**
> Les types de média webhook peuvent être mis à niveau de manière similaire : récupérez le fichier `media_bundle-8.0-000.yaml` dans l'archive des ensembles de modèles, puis importez-le dans *Alertes* > *Types de média*.

Pour mettre à niveau ou ajouter **tous les modèles** en une seule fois, vous pouvez utiliser le script `import_templates.sh` inclus dans l'archive des ensembles de modèles. Le script nécessite les utilitaires `curl` et `jq`, et utilise la méthode de l'API Zabbix `configuration.import`. Lors de l'exécution du script, indiquez le répertoire contenant les fichiers YAML, ainsi que l'URL de votre API Zabbix et votre jeton d'authentification.

> **Warning:**
> Le script `import_templates.sh` écrase tous les modèles existants, sans possibilité de réviser ou d'annuler les modifications apportées aux modèles.

Exemple :

```
/tmp/templates_bundles-8.0-2M-apache/import_templates.sh \
  /tmp/templates_bundles-8.0-2M-apache \
  https://example.com/zabbix/api_jsonrpc.php \
  d8d6b5c78ee2a8333db11bfcd3e448c49c239d8ce00702759e2b56f42e24dbe4
```

Compatibilité des modèles avec l'assistant de configuration des hôtes

L'assistant de configuration des hôtes est compatible avec tous les modèles prêts à l'emploi.

Vous pouvez également rendre vos modèles personnalisés compatibles avec l'assistant de configuration des hôtes :

1. Dans l'interface Zabbix, accédez à *Collecte de données* > *Modèles* et exportez le modèle personnalisé.

2. Modifiez le fichier exporté :

- Ajoutez l'élément `wizard_ready` du modèle pour indiquer la compatibilité avec l'assistant de configuration des hôtes.
- Si nécessaire, ajoutez également l'élément `readme` ou `config` du modèle.

3. Dans l'interface Zabbix, accédez à *Collecte de données* > *Modèles* et importez le fichier de modèle mis à jour.

Fonctionnement du modèle de l'agent Zabbix

Étapes à suivre pour garantir le bon fonctionnement des modèles qui collectent des métriques avec Zabbix agent :

1. Assurez-vous que Zabbix agent est installé sur l'hôte. Pour les vérifications actives, assurez-vous également que l'adresse du serveur/proxy Zabbix est ajoutée au paramètre 'ServerActive' du fichier de configuration de l'agent.
2. Liez le modèle à un hôte cible (si le modèle n'est pas disponible dans votre installation Zabbix, vous devrez peut-être d'abord l'importer).
3. Si nécessaire, ajustez les valeurs des macros du modèle.
4. Configurez l'instance surveillée afin d'autoriser le partage des données avec Zabbix.

Une description détaillée d'un modèle, comprenant la liste complète des macros, éléments et déclencheurs, est disponible dans le fichier README du modèle (accessible en cliquant sur le nom d'un modèle).

Les modèles suivants sont disponibles :

- Apache by Zabbix agent
- Apache by Zabbix agent active
- HAProxy by Zabbix agent
- IIS by Zabbix agent
- IIS by Zabbix agent active
- MariaDB by Zabbix agent
- MariaDB by Zabbix agent active
- Microsoft Exchange Server 2016 by Zabbix agent
- Microsoft Exchange Server 2016 by Zabbix agent active
- MySQL by Zabbix agent
- MySQL by Zabbix agent active
- Nginx by Zabbix agent
- Nginx by Zabbix agent active
- Percona by Zabbix agent
- Percona by Zabbix agent active
- PHP-FPM by Zabbix agent
- PHP-FPM by Zabbix agent active
- Podman by Zabbix agent
- Podman by Zabbix agent active
- PostgreSQL by Zabbix agent
- PostgreSQL by Zabbix agent active
- RabbitMQ cluster by Zabbix agent

Opération du modèle Zabbix agent 2

Étapes pour garantir le bon fonctionnement des modèles qui collectent des métriques avec Zabbix agent 2 :

1. Assurez-vous que agent 2 est installé sur l'hôte et que la version installée contient le plugin requis. Dans certains cas, vous devrez peut-être d'abord mettre à niveau agent 2.
2. Liez le modèle à un hôte cible (si le modèle n'est pas disponible dans votre installation Zabbix, vous devrez peut-être d'abord importer le modèle).
3. Si nécessaire, ajustez les valeurs des macros de modèle. Notez que les macros utilisateur peuvent être utilisées pour remplacer des paramètres de configuration.
4. Configurez l'instance surveillée pour autoriser le partage de données avec Zabbix.

Attention:
Les modèles Zabbix agent 2 fonctionnent en conjonction avec les plugins. Bien que la configuration de base puisse être effectuée simplement en ajustant les macros utilisateur, une personnalisation plus poussée peut être réalisée en configurant le plugin lui-même. Par exemple, si un plugin prend en charge les sessions nommées, il est possible de surveiller plusieurs entités du même type (par exemple, MySQL1 et MySQL2) en spécifiant une session nommée avec sa propre URI, son nom d'utilisateur et son mot de passe pour chaque entité dans le fichier de configuration.

Une description détaillée d'un modèle, y compris la liste complète des macros, éléments et déclencheurs, est disponible dans le fichier README du modèle (accessible en cliquant sur le nom d'un modèle).

Les modèles suivants sont disponibles :

- Ceph by Zabbix agent 2
- Docker
- Memcached
- MongoDB cluster by Zabbix agent 2
- MongoDB node by Zabbix agent 2
- MySQL by Zabbix agent 2
- MySQL by Zabbix agent 2 active
- Oracle by Zabbix agent 2
- PostgreSQL by Zabbix agent 2
- PostgreSQL by Zabbix agent 2 active
- Redis
- SMART by Zabbix agent 2
- SMART by Zabbix agent 2 active
- Systemd by Zabbix agent 2
- Website certificate by Zabbix agent 2

Opération du modèle HTTP

Étapes à suivre pour garantir le bon fonctionnement des modèles qui collectent des métriques avec l'agent HTTP :

1. Créez un hôte dans Zabbix correspondant à la cible de supervision. Indiquez l'adresse IP ou le nom DNS de la cible de supervision comme interface principale de l'hôte. Cette configuration est nécessaire pour que la macro `{HOST.CONN}` soit correctement résolue dans les éléments du modèle.
2. Liez le modèle à l'hôte créé à l'étape 1 (si le modèle n'est pas disponible dans votre installation Zabbix, vous devrez peut-être d'abord importer le modèle).
3. Si nécessaire, ajustez les valeurs des macros au niveau du modèle.
4. Configurez l'instance supervisée afin qu'elle autorise le partage de données avec Zabbix.

Une description détaillée d'un modèle, comprenant la liste complète des macros, des éléments et des déclencheurs, est disponible dans le fichier `README` du modèle (accessible en cliquant sur le nom d'un modèle).

Les modèles suivants sont disponibles :

- Acronis Cyber Protect Cloud via HTTP
- Apache via HTTP
- Asterisk via HTTP
- AWS via HTTP
- AWS Cost Explorer via HTTP
- AWS EC2 via HTTP
- AWS ECS Cluster via HTTP
- AWS ECS Serverless Cluster via HTTP
- AWS ELB Application Load Balancer via HTTP
- AWS ELB Network Load Balancer via HTTP
- AWS Lambda via HTTP
- AWS RDS instance via HTTP
- AWS S3 bucket via HTTP
- Azure via HTTP
- Cisco Meraki organization via HTTP
- Cisco SD-WAN via HTTP
- Cisco Secure Firewall Threat Defense via HTTP
- Claude API via HTTP
- ClickHouse via HTTP
- Cloudflare via HTTP
- CockroachDB via HTTP
- Control-M enterprise manager via HTTP
- Control-M server via HTTP
- Cradlepoint NCM v2 via HTTP
- DELL PowerEdge R720 via HTTP
- DELL PowerEdge R740 via HTTP
- DELL PowerEdge R820 via HTTP

- DELL PowerEdge R840 via HTTP
- Domain RDAP via HTTP
- Elasticsearch Cluster via HTTP
- Envoy Proxy via HTTP
- Etcd via HTTP
- FortiGate via HTTP
- GitHub repository via HTTP
- GitLab via HTTP
- GLPI via HTTP
- Google Cloud Platform (GCP) via HTTP
- Hadoop via HTTP
- HAProxy via HTTP
- HashiCorp Consul Cluster via HTTP
- HashiCorp Consul Node via HTTP
- HashiCorp Nomad via HTTP
- HashiCorp Vault via HTTP
- Hikvision camera via HTTP
- HPE iLO via HTTP
- HPE MSA 2040 Storage via HTTP
- HPE MSA 2060 Storage via HTTP
- HPE Primera via HTTP
- HPE Synergy via HTTP
- InfluxDB via HTTP
- Jenkins via HTTP
- Kubernetes API server via HTTP, Kubernetes cluster state via HTTP, Kubernetes Controller manager via HTTP, Kubernetes kubelet via HTTP, Kubernetes nodes via HTTP, Kubernetes Scheduler via HTTP sont obsolètes et ont été remplacés par les modèles *Kubernetes Cluster via HTTP* et *Kubernetes Node via HTTP*.
- Kubernetes Cluster via HTTP
- Kubernetes Node via HTTP
- MantisBT via HTTP
- Microsoft 365 reports via HTTP
- Microsoft SharePoint via HTTP
- NetApp AFF A700 via HTTP
- Nextcloud via HTTP
- NGINX via HTTP
- NGINX Plus via HTTP
- Nutanix Prism Element via HTTP
- OpenAI Platform via HTTP
- OpenStack via HTTP
- OpenWeatherMap via HTTP
- Oracle Cloud via HTTP
- Palo Alto PA-440 via HTTP
- PHP-FPM via HTTP
- Podman via HTTP
- Proxmox VE via HTTP
- Pure Storage FlashArray v1 and v2 via HTTP
- RabbitMQ cluster via HTTP
- Ribbon SBC Edge via HTTP
- Ribbon SBC SWe CE via HTTP
- Ribbon SBC SWe Core via HTTP
- TiDB via HTTP
- TiDB PD via HTTP
- TiDB TiKV via HTTP
- Travis CI via HTTP
- Veeam Backup Enterprise Manager via HTTP
- Veeam Backup and Replication via HTTP
- VeloCloud SD-WAN via HTTP
- VeloCloud SD-WAN Edge via HTTP
- YugabyteDB via HTTP
- ZooKeeper via HTTP

Opération du modèle IPMI

Les modèles IPMI ne nécessitent aucune configuration spécifique. Pour commencer la surveillance, associez le modèle à un hôte cible (si le modèle n'est pas disponible dans votre installation Zabbix, vous devrez peut-être d'abord l'importer).

Une description détaillée d'un modèle, y compris la liste complète des macros, des éléments et des déclencheurs, est disponible dans le fichier README du modèle (accessible en cliquant sur le nom du modèle).

Modèle disponible :

- Chassis by IPMI

Fonctionnement du modèle JMX

Étapes pour garantir le bon fonctionnement des modèles qui collectent des métriques via JMX :

1. Assurez-vous que le Java gateway de Zabbix est installé et correctement configuré.
2. Liez le modèle à l'hôte cible. L'hôte doit avoir l'interface JMX configurée. Si le modèle n'est pas disponible dans votre installation de Zabbix, vous devrez peut-être d'abord importer le modèle.
3. Si nécessaire, ajustez les valeurs des macros du modèle.
4. Configurez l'instance surveillée pour autoriser le partage des données avec Zabbix.

Une description détaillée d'un modèle, y compris la liste complète des macros, éléments et déclencheurs, est disponible dans le fichier README du modèle (accessible en cliquant sur le nom du modèle).

Les modèles suivants sont disponibles :

- Apache ActiveMQ by JMX
- Apache Cassandra by JMX
- Apache Kafka by JMX
- Apache Tomcat by JMX
- GridGain by JMX
- Ignite by JMX
- Jira Data Center by JMX
- WildFly Domain by JMX
- WildFly Server by JMX

Opération du modèle ODBC

Étapes pour garantir le bon fonctionnement des modèles qui collectent des métriques via la surveillance ODBC :

1. Assurez-vous que le pilote ODBC requis est installé sur le serveur Zabbix ou le proxy.
2. Liez le modèle à un hôte cible (si le modèle n'est pas disponible dans votre installation Zabbix, vous devrez peut-être d'abord l'importer).
3. Si nécessaire, ajustez les valeurs des macros du modèle.
4. Configurez l'instance surveillée pour autoriser le partage des données avec Zabbix.

Une description détaillée d'un modèle, y compris la liste complète des macros, éléments et déclencheurs, est disponible dans le fichier README du modèle (accessible en cliquant sur le nom du modèle).

Les modèles suivants sont disponibles :

- MariaDB by ODBC
- MSSQL by ODBC
- MySQL by ODBC
- Oracle by ODBC
- Percona by ODBC
- PostgreSQL by ODBC

Modèles standardisés pour les équipements réseau

Aperçu

Afin de fournir une supervision des périphériques réseau tels que les commutateurs et les routeurs, nous avons créé deux modèles dits: un pour le périphérique réseau lui-même (son châssis, en gros) et un pour l'interface réseau.

Des modèles pour de nombreuses familles de périphériques réseau sont fournis. Tous les modèles couvrent (lorsqu'il est possible d'obtenir ces éléments depuis le périphérique) :

- Supervision des pannes du châssis (alimentations, ventilateurs et température, état global)

- Supervision des performances du châssis (éléments CPU et mémoire)
- Collecte de l'inventaire du châssis (numéros de série, nom du modèle, version du micrologiciel)
- Supervision des interfaces réseau avec IF-MIB et EtherLike-MIB (état de l'interface, charge du trafic de l'interface, état duplex pour Ethernet)

Si le modèle n'est pas disponible dans votre installation Zabbix, vous devrez peut-être d'abord l'importer.

Si vous importez les nouveaux modèles fournis par défaut, vous pouvez également mettre à jour l'expression régulière globale `@Network interfaces for discovery` vers :

```
Does not match regular expression: ^Software Loopback Interface
Does not match regular expression: ^(In)?[lL]oop[bB]ack[0-9._]*$
Does not match regular expression: ^NULL[0-9.]*$
Does not match regular expression: ^[lL]o[0-9.]*$
Does not match regular expression: ^[sS]ystem$
Does not match regular expression: ^Nu[0-9.]*$
```

afin d'exclure les interfaces de bouclage et les interfaces null sur la plupart des systèmes.

Appareils

Liste des familles d'appareils pour lesquelles des modèles sont disponibles :

Nom du modèle	Fournisseur	Famille d'appareils	Modèles connus	OS	MIB utilisés	**Tags**
Alcatel Timetra TiMOS SNMP	Alcatel	Alcatel Timetra	ALCATEL SR 7750	TiMOS	TIMETRA-SYSTEM-MIB,TIMETRA-CHASSIS-MIB	Certified
Aruba CX 8300s by SNMP	HPE Aruba Networking	Série de commutateurs Aruba CX	Aruba JL636A 8325, Aruba JL717A 8360	Aruba AOS-CX	ARUBAWIRED-FAN-MIB, ARUBAWIRED-POWERSUPPLY-MIB, ARUBAWIRED-SYSTEMINFO-MIB, ARUBAWIRED-TEMPSENSOR-MIB, OSPF-MIB	Certified
Brocade FC SNMP	Brocade	Commutateurs FC Brocade	Brocade 300 SAN Switch-	-	SW-MIB,ENTITY-MIB	Performance Fault
Brocade_Foundry Stackable SNMP	Brocade	Brocade ICX	Brocade ICX6610, Brocade ICX7250-48, Brocade ICX7450-48F		FOUNDRY-SN-AGENT-MIB, FOUNDRY-SN-STACKING-MIB	Certified
Brocade_Foundry Nonstackable SNMP	Brocade, Foundry	Brocade MLX, Foundry	Brocade MLXe, Foundry FLS648, Foundry FWSX424		FOUNDRY-SN-AGENT-MIB	Performance Fault
Check Point Next Generation Firewall by SNMP	Check Point	Pare-feu nouvelle génération	-	Gaia	HOST-RESOURCES-MIB, CHECKPOINT-MIB, UCD-SNMP-MIB, SNMPv2-MIB, IF-MIB	Certified
Ciena 3906 by SNMP	Ciena	CPE Ciena	Ciena 3906	SAOS	WWP-LEOS-BLADE-MIB, WWP-LEOS-CHASSIS-MIB, WWP-LEOS-SYSTEM-CONFIG-MIB	Certified

Nom du modèle	Fournisseur	Famille d'appareils	Modèles connus	OS	MIB utilisés	**Tags**
Cisco Catalyst 3750<device model> SNMP	Cisco	Cisco Catalyst 3750	Cisco Catalyst 3750V2-24FS, Cisco Catalyst 3750V2-24PS, Cisco Catalyst 3750V2-24TS, Cisco Catalyst SNMP, Cisco Catalyst SNMP		CISCO-MEMORY-POOL-MIB, IF-MIB, EtherLike-MIB, SNMPv2-MIB, CISCO-PROCESS-MIB, CISCO-ENVMON-MIB, ENTITY-MIB	Certified
Cisco IOS SNMP	Cisco	Cisco IOS ver > 12.2 3.5	Cisco C2950	IOS	CISCO-PROCESS-MIB,CISCO-MEMORY-POOL-MIB,CISCO-ENVMON-MIB	Certified
Cisco IOS versions 12.0_3_T-12.2_3.5 SNMP	Cisco	Cisco IOS > 12.0 3 T et 12.2 3.5	-	IOS	CISCO-PROCESS-MIB,CISCO-MEMORY-POOL-MIB,CISCO-ENVMON-MIB	Certified
Cisco IOS prior to 12.0_3_T SNMP	Cisco	Cisco IOS 12.0 3 T	-	IOS	OLD-CISCO-CPU-MIB,CISCO-MEMORY-POOL-MIB	Certified
D-Link DES_DGS Switch SNMP	D-Link	Commutateurs DES/DGX	D-Link DES-xxxx/DGS-xxxx,DLINK DGS-3420-26SC	-	DLINK-AGENT-MIB,EQUIPMENT-MIB,ENTITY-MIB	Certified
D-Link DES 7200 SNMP	D-Link	DES-7xxx	D-Link DES 7206	-	ENTITY-MIB,MY-SYSTEM-MIB,MY-PROCESS-MIB,MY-MEMORY-MIB	Performance Fault Interfaces
Dell Force S-Series SNMP	Dell	Dell Force S-Series	S4810		F10-S-SERIES-CHASSIS-MIB	Certified
Extreme Exos SNMP	Extreme	Extreme EXOS	X670V-48x	EXOS	EXTREME-SYSTEM-MIB,EXTREME-SOFTWARE-MONITOR-MIB	Certified
FortiGate by SNMP	Fortinet	FortiGate (NGFW)	-	FortiOS	HOST-RESOURCES-MIB FORTINET-FORTIGATE-MIB FORTINET-CORE-MIB SNMPv2-MIB IF-MIB ENTITY-MIB	Performance Inventory
HP Comware HH3C SNMP	HP	HP (H3C) Comware	HP A5500-24G-4SFP HI Switch		HH3C-ENTITY-EXT-MIB,ENTITY-MIB	Certified
HP Enterprise Switch SNMP	HP	Commutateurs d'entreprise HP	HP ProCurve J4900B Switch 2626, HP J9728A 2920-48G Switch		STATISTICS-MIB,NETSWITCH-MIB,HP-ICF-CHASSIS,ENTITY-MIB,SEMI-MIB	Certified

Nom du modèle	Fournisseur	Famille d'appareils	Modèles connus	OS	MIB utilisés	**Tags**
Huawei AR600 by SNMP	Huawei	Routeurs Huawei NetEngine série AR600	Huawei AR600/AR6100/AR6200/AR6300 series routers with VRP software (V300R019/V300R021/V300R023 or later)		HUAWEI-CBQOS-MIB, HUAWEI-NQA-MIB, ENTITY-MIB, IF-MIB, HOST-RESOURCES-MIB, SNMPv2-MIB	Certified
Huawei OceanStor 5300 V5 by SNMP	Huawei	Huawei OceanStor Dorado	Huawei OceanStor Dorado with V5 software		HUAWEI-STORAGE-HARDWARE-MIB, HUAWEI-STORAGE-SPACE-MIB, ISM-PERFORMANCE-MIB, HOST-RESOURCES-MIB, SNMPv2-MIB	Certified
Huawei OceanStor Dorado by SNMP	Huawei	Huawei OceanStor, Huawei OceanStor Dorado	Huawei OceanStor Dorado 3000/5000/6000/8000/18000/5300V3/5500/5600/6800/18500/18800 with V6/V700 software		HUAWEI-STORAGE-HARDWARE-MIB, HUAWEI-STORAGE-SPACE-MIB, ISM-PERFORMANCE-MIB, HOST-RESOURCES-MIB, SNMPv2-MIB	Certified
Huawei VRP by SNMP	Huawei	Huawei VRP	S2352P-EI		ENTITY-MIB,HUAWEI-ENTITY-EXTENT-MIB	Certified
Intel_Qlogic Infiniband SNMP	Intel/QLogic	Appareils Infini-band In-tel/QLogic	Infiniband 12300		ICS-CHASSIS-MIB	Fault Inventory
Juniper SNMP	Juniper	Modèles MX,SRX,EX	Juniper MX240, Juniper EX4200-24F	JunOS	JUNIPER-MIB	Certified
Juniper MX NETCONF	Juniper	Modèles MX	Juniper MX204 Edge Router	JunOS 24.2R1-S1.10	*Not applicable (uses NETCONF over SSH)*	Certified
Juniper MX SNMP	Juniper	Modèles MX	Juniper MX204 Edge Router	JunOS 24.2R1-S1.10	OSPF-MIB, JUNIPER-DOM-MIB, JUNIPER-MIB, BGP4-V2-MIB-JUNIPER, OSPFV3-MIB-JUNIPER	Certified
Mellanox SNMP	Mellanox	Appareils Infini-band Mellanox	SX1036	MLNX-OS	HOST-RESOURCES-MIB,ENTITY-MIB,ENTITY-SENSOR-MIB,MELLANOX-MIB	Certified

Nom du modèle	Fournisseur	Famille d'appareils	Modèles connus	OS	MIB utilisés	**Tags**
MikroTik CCR<device model> SNMP	MikroTik	Routeurs MikroTik Cloud Core (série CCR)	Des modèles dédiés distincts sont disponibles pour MikroTik CCR1009-7G-1C-1S+, MikroTik CCR1009-7G-1C-1S+PC, MikroTik CCR1009-7G-1C-PC, MikroTik CCR1016-12G, MikroTik CCR1016-12S-1S+, MikroTik CCR1036-12G-4S-EM, MikroTik CCR1036-12G-4S, MikroTik CCR1036-8G-2S+, MikroTik CCR1036-8G-2S+EM, MikroTik CCR1072-1G-8S+, MikroTik CCR2004-16G-2S+, MikroTik CCR2004-1G-12S+2XS	RouterOS	MIKROTIK-MIB,HOST-RESOURCES-MIB	Certified

Nom du modèle	Fournisseur	Famille d'appareils	Modèles connus	OS	MIB utilisés	**Tags**
MikroTik CRS<device model> SNMP	MikroTik	Commutateurs MikroTik Cloud Router (série CRS)	Des modèles dédiés distincts sont disponibles pour MikroTik CRS106-1C-5S, MikroTik CRS109-8G-1S-2HnD-IN, MikroTik CRS112-8G-4S-IN, MikroTik CRS112-8P-4S-IN, MikroTik CRS125-24G-1S-2HnD-IN, MikroTik CRS212-1G-10S-1S+IN, MikroTik CRS305-1G-4S+IN, MikroTik CRS309-1G-8S+IN, MikroTik CRS312-4C+8XG-RM, MikroTik CRS317-1G-16S+RM, MikroTik CRS326-24G-2S+IN, MikroTik CRS326-24G-2S+RM, MikroTik CRS326-24S+2Q+RM, MikroTik CRS328-24P-4S+RM, MikroTik CRS328-4C-20S-4S+RM, MikroTik CRS354-48G-4S+2Q+RM, MikroTik CRS354-48P-4S+2Q+RM	RouterOS/SwitchOS	MIKROTIK-MIB,HOST-RESOURCES-MIB	Certified
MikroTik CSS<device model> SNMP	MikroTik	Commutateurs intelligents MikroTik Cloud (série CSS)	Des modèles dédiés distincts sont disponibles pour MikroTik CSS326-24G-2S+RM, MikroTik CSS610-8G-2S+IN	RouterOS	MIKROTIK-MIB,HOST-RESOURCES-MIB	Certified
MikroTik FiberBox SNMP	MikroTik	MikroTik FiberBox	MikroTik FiberBox	RouterOS	MIKROTIK-MIB,HOST-RESOURCES-MIB	Certified
MikroTik hEX <device model> SNMP	MikroTik	MikroTik hEX	Des modèles dédiés distincts sont disponibles pour MikroTik hEX, MikroTik hEX lite, MikroTik hEX PoE, MikroTik hEX PoE lite, MikroTik hEX S	RouterOS	MIKROTIK-MIB,HOST-RESOURCES-MIB	Certified

Nom du modèle	Fournisseur	Famille d'appareils	Modèles connus	OS	MIB utilisés	**Tags**
MikroTik netPower <device model> SNMP	MikroTik	MikroTik net-Power	Des modèles dédiés distincts sont disponibles pour MikroTik netPower 15FR, MikroTik netPower 16P SNMP, MikroTik netPower Lite 7R	RouterOS/SwitchOS, SwitchOS Lite	MIKROTIK-MIB,HOST-RESOURCES-MIB	Certified
MikroTik PowerBox <device model> SNMP	MikroTik	MikroTik Power-Box	Des modèles dédiés distincts sont disponibles pour MikroTik PowerBox, MikroTik PowerBox Pro	RouterOS	MIKROTIK-MIB,HOST-RESOURCES-MIB	Certified
MikroTik RB<device model> SNMP	MikroTik	Routeurs MikroTik série RB	Des modèles dédiés distincts sont disponibles pour MikroTik RB1100AHx4, MikroTik RB1100AHx4 Dude Edition, MikroTik RB2011iL-IN, MikroTik RB2011iL-RM, MikroTik RB2011iLS-IN, MikroTik RB2011UiAS-IN, MikroTik RB2011UiAS-RM, MikroTik RB260GS, MikroTik RB3011UiAS-RM, MikroTik RB4011iGS+RM, MikroTik RB5009UG+S+IN	RouterOS	MIKROTIK-MIB,HOST-RESOURCES-MIB	Certified
MikroTik SNMP	MikroTik	Appareils MikroTik RouterOS	MikroTik CCR1016-12G, MikroTik RB2011UAS-2HnD, MikroTik 912UAG-5HPnD, MikroTik 941-2nD, MikroTik 951G-2HnD, MikroTik 1100AHx2	RouterOS	MIKROTIK-MIB,HOST-RESOURCES-MIB	Certified
Netgear Fastpath SNMP	Netgear	Netgear Fastpath	M5300-28G		FASTPATH-SWITCHING-MIB,FASTPATH-BOXSERVICES-PRIVATE-MIB	Fault Inventory
QTech QSW SNMP	QTech	Appareils Qtech	Qtech QSW-2800-28T	-	QTECH-MIB,ENTITY-MIB	Performance Inventory

Nom du modèle	Fournisseur	Famille d'appareils	Modèles connus	OS	MIB utilisés	**Tags**
Stormshield SNS by SNMP	Stormshield	Pare-feu Stormshield Network Security (SNS)	SN3100	Stormshield SNS	HOST-RESOURCES-MIB, UCD-SNMP-MIB, STORMSHIELD-ASQ-STATS-MIB, STORMSHIELD-AUTOUPDATE-MIB, STORMSHIELD-HA-MIB, STORMSHIELD-PROPERTY-MIB, STORMSHIELD-HEALTH-MONITOR-MIB, STORMSHIELD-IF-MIB, STORMSHIELD-SYSTEM-MONITOR-MIB, STORMSHIELD-IPSEC-STATS-MIB	Certified
TP-LINK SNMP	TP-LINK	TP-LINK	T2600G-28TS v2.0		TPLINK-SYSMONITOR-MIB,TPLINK-SYSINFO-MIB	Performance Inventory
Ubiquiti AirOS SNMP	Ubiquiti	Appareils sans fil Ubiquiti AirOS	NanoBridge,NanoStation,Unifi	AirOS	FROGFOOT-RESOURCES-MIB,IEEE802dot11-MIB	Performance
Vyatta Virtual Router by SNMP	Ciena	Vyatta	Vyatta Virtual Router 1908e	Vyatta 1908e	SNMPv2-MIB, HOST-RESOURCES-MIB, UCD-SNMP-MIB, IF-MIB, DISMAN-EVENT-MIB	Performance Inventory

Conception des modèles

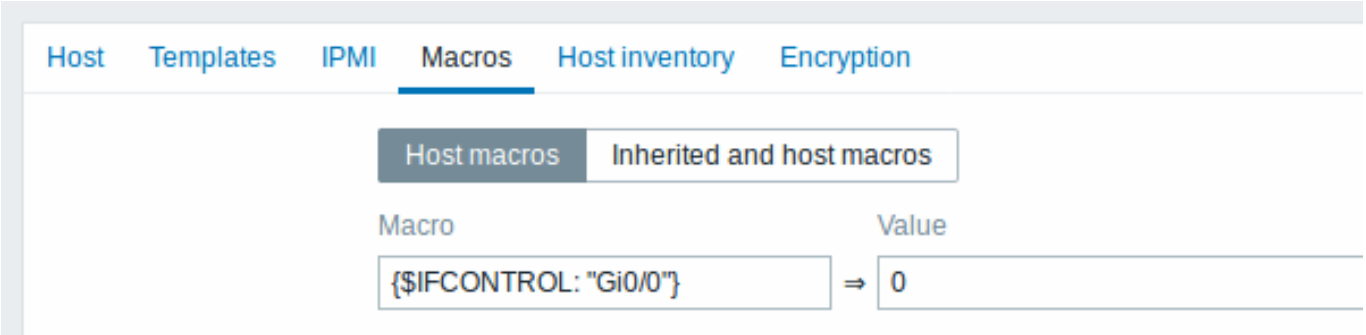
Les modèles ont été conçus en tenant compte des points suivants :

- Les macros utilisateur sont utilisées autant que possible afin que les déclencheurs puissent être ajustés par l'utilisateur ;
- La découverte de bas niveau est utilisée autant que possible afin de minimiser le nombre d'éléments non pris en charge ;
- Tous les modèles dépendent de Template ICMP Ping, de sorte que tous les périphériques sont également vérifiés par ICMP ;
- Les éléments n'utilisent aucun MIB - les OID SNMP sont utilisés dans les éléments et les découvertes de bas niveau. Il n'est donc pas nécessaire de charger des MIB dans Zabbix pour que les modèles fonctionnent ;
- Les interfaces réseau de bouclage sont filtrées lors de la découverte, ainsi que les interfaces avec ifAdminStatus = down(2)
- Des compteurs 64 bits provenant de IF-MIB::ifXTable sont utilisés lorsque cela est possible. Si cela n'est pas pris en charge, les compteurs 32 bits par défaut sont utilisés à la place.

Toutes les interfaces réseau découvertes disposent d'un déclencheur qui surveille leur état opérationnel (lien), par exemple :

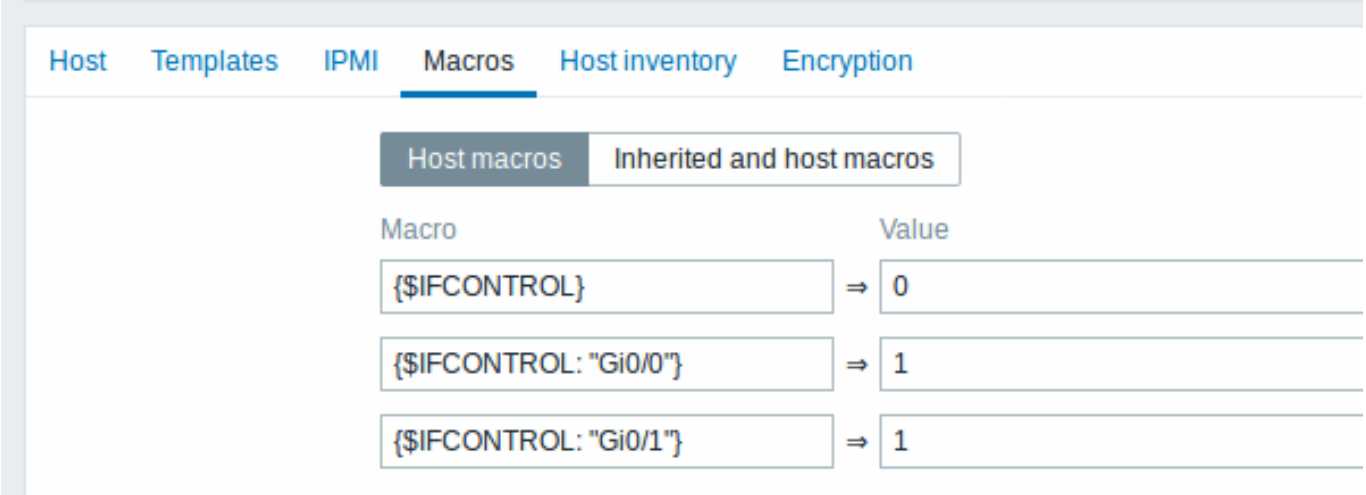
```
{$IFCONTROL:"{#IFNAME}"}=1 and last(/Alcatel Timetra TiMOS SNMP/net.if.status[ifOperStatus.{#SNMPINDEX}])
```

- Si vous ne souhaitez pas surveiller cette condition pour une interface spécifique, créez une macro utilisateur avec contexte ayant la valeur 0. Par exemple :



où Gi0/0 est {#IFNAME}. De cette façon, le déclencheur n'est plus utilisé pour cette interface spécifique.

- Vous pouvez également modifier le comportement par défaut afin que tous les déclencheurs ne se déclenchent pas, et activer ce déclencheur uniquement pour un nombre limité d'interfaces, comme les liaisons montantes :



Balises

- Performance - les MIB de la famille d'appareils offrent un moyen de surveiller les éléments de processeur et de mémoire ;
- Défaut - les MIB de la famille d'appareils offrent un moyen de surveiller au moins un capteur de température ;
- Inventaire - les MIB de la famille d'appareils offrent un moyen de collecter au moins le numéro de série et le nom du modèle de l'appareil ;
- Certifié - les trois catégories principales ci-dessus sont couvertes.

Opération du modèle VMware

Aperçu

Zabbix inclut une gamme de modèles prêts à l'emploi pour la surveillance des machines virtuelles, conçus pour VMware vCenter et les hyperviseurs ESXi. Les modèles disponibles sont regroupés en deux ensembles de modèles distincts :

- VMware - utilise les données UUID pour les macros correspondantes
- VMware FQDN - utilise les données FQDN pour les macros correspondantes

Les modèles conçus pour surveiller VMware vCenter ou les hyperviseurs ESXi contiennent des règles de découverte de bas niveau préconfigurées ainsi que divers contrôles intégrés pour surveiller les installations virtuelles.

Note:
Pour que le modèle *VMware FQDN* fonctionne correctement, chaque machine virtuelle surveillée doit avoir un nom de système d'exploitation unique conforme aux règles FQDN. De plus, VMware Tools/Open Virtual Machine Tools doivent être installés sur chaque machine. Si ces prérequis sont remplis, il est recommandé d'utiliser le modèle *VMware FQDN*. Le modèle *VMware FQDN* est disponible depuis Zabbix 5.2 avec l'introduction de la possibilité de créer des hôtes avec des interfaces personnalisées.

 Un modèle *VMware* classique est également disponible et peut être utilisé si les exigences FQDN ne sont pas respectées. Cependant, le modèle *VMware* présente un problème connu. Les hôtes des machines virtuelles découvertes sont créés avec les noms enregistrés dans vCenter (par exemple, "VM1", "VM2", etc.). Si Zabbix agent est installé sur ces hôtes et que l'autoréenregistrement actif de Zabbix agent est activé, le processus d'autoréenregistrement lira les noms d'hôte tels qu'ils ont été enregistrés au démarrage (par exemple, "vm1.example.com", "vm2.example.com", etc.). Cela peut entraîner la création de nouveaux hôtes pour des machines virtuelles existantes (puisqu'aucune correspondance de nom n'a été trouvée), ce qui entraîne des hôtes en double avec des noms différents.

Veuillez noter :

- Le modèle *VMware Hypervisor* peut être lié manuellement à un hôte ainsi qu'être utilisé dans la découverte ;
- Le modèle *VMware Guest* ne doit pas être lié manuellement à un hôte et ne peut être utilisé que dans la découverte.

Configuration des macros d'hôte

Pour utiliser les vérifications simples VMware, l'hôte doit avoir les macros utilisateur suivantes définies :

- {$VMWARE.URL} - URL du SDK du service VMware (vCenter ou hyperviseur ESXi) (https://servername/sdk)
- {$VMWARE.USERNAME} - nom d'utilisateur du service VMware
- {$VMWARE.PASSWORD} - mot de passe de l'utilisateur {$VMWARE.USERNAME} du service VMware

Fonctionnement du modèle OTLP

Zabbix fournit une gamme de modèles prêts à l'emploi pour collecter et surveiller les traces, les métriques et les journaux OpenTelemetry à l'aide d'éléments *Requête de télémétrie*.

Étapes à suivre pour garantir le bon fonctionnement des modèles :

1. Consultez le fichier `README` du modèle pour connaître les exigences de configuration spécifiques à celui-ci.
2. Créez un hôte dans Zabbix correspondant à la cible de supervision.
3. Vérifiez que le paramètre `TelemetryProvider` est configuré dans le fichier de configuration du serveur Zabbix ou du proxy Zabbix.
4. Liez le modèle à l'hôte créé (si le modèle n'est pas disponible dans votre installation Zabbix, vous devrez peut-être d'abord l'importer).
5. Si nécessaire, ajustez les valeurs des macros au niveau du modèle.

Une description détaillée des modèles, comprenant la liste complète des macros, des éléments et des déclencheurs, est disponible dans le fichier `README` du modèle, auquel vous pouvez accéder en cliquant sur le nom du modèle.

Les modèles suivants sont disponibles :

- Envoy Proxy by OTLP
- Generic OpenTelemetry by OTLP
- OpenRouter traces by OTLP
- Proxmox VE by OTLP

Notifications lors des événements

Aperçu

En supposant que nous ayons configuré certains éléments et déclencheurs et que certains événements se produisent maintenant à la suite du changement d'état des déclencheurs, il est temps d'envisager certaines actions.

Pour commencer, nous ne voudrions pas regarder les déclencheurs ou la liste des événements tout le temps. Il serait préférable de recevoir une notification si quelque chose d'important (comme un problème) s'est produit. Aussi, lorsque des problèmes surviennent, nous aimerions que toutes les personnes concernées soient informées.

C'est pourquoi l'envoi de notifications est l'une des principales actions proposées par Zabbix. Qui et quand doit être notifié lors d'un certain événement peut être défini.

Pour pouvoir envoyer et recevoir des notifications de Zabbix, vous devez :

- définir certains médias
- configurer une action qui envoie un message à l'un des médias définis

Les actions consistent en *conditions* et *opérations*. Fondamentalement, lorsque les conditions sont remplies, les opérations sont effectuées. Les deux opérations principales sont l'envoi d'un message (notification) et l'exécution d'une commande à distance.

Pour les événements créés par découverte et enregistrement automatique, certaines opérations supplémentaires sont disponibles. Ceux-ci incluent l'ajout ou la suppression d'un hôte, la liaison d'un modèle, etc.

Types de média

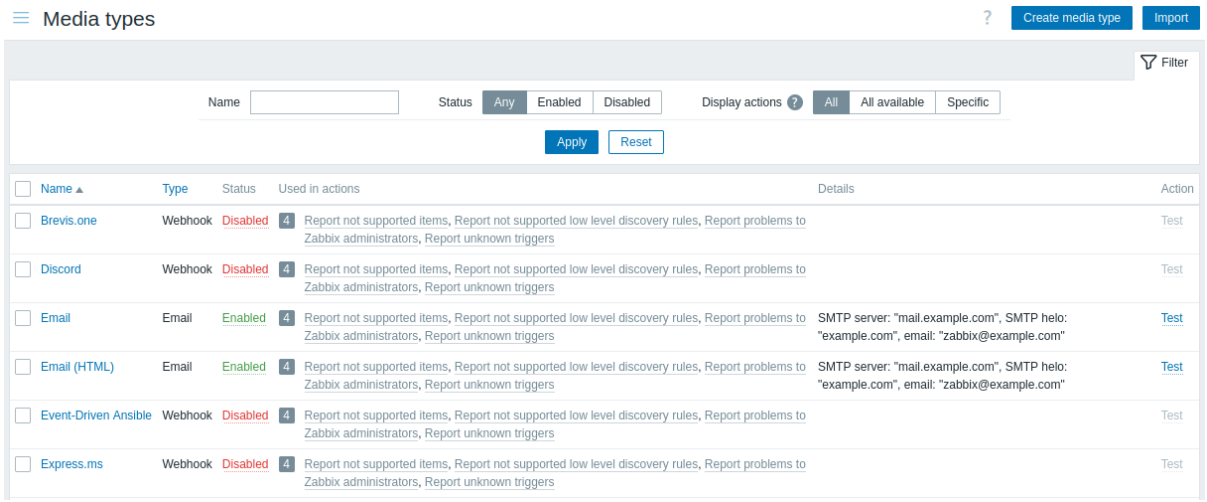
Présentation

Les types de média sont les canaux de diffusion utilisés pour envoyer des notifications et des alertes depuis Zabbix.

Les types de média prennent en charge les méthodes de diffusion suivantes :

- E-mail
- Notification push
- SMS
- Script personnalisé
- Webhook

Les types de média sont gérés dans *Alertes* > *Types de média*. Certains types de média sont prédéfinis dans le jeu de données par défaut. Il vous suffit d'ajuster leurs paramètres pour les rendre opérationnels.



Pour comprendre comment les types de média s'intègrent au processus d'alerte, examinons les trois conditions requises pour transmettre des notifications de Zabbix aux utilisateurs finaux :

1. Une opération d'action qui envoie les notifications doit être définie.
2. Un type de média fonctionnel doit être défini (par exemple *E-mail*, qui envoie des alertes à l'aide de SMTP).
3. Les informations de diffusion au niveau utilisateur (telles que les adresses e-mail, les numéros de téléphone, etc.) doivent être définies dans les médias utilisateur.

Test des types de média

Pour vérifier si un type de média configuré fonctionne, cliquez sur *Tester* dans la liste des types de média.

La demande de test sera envoyée au serveur Zabbix. Le serveur Zabbix tentera d'envoyer une alerte à l'aide du type de média spécifié et renverra le résultat à l'interface. L'interface attendra que le serveur renvoie les résultats. Le test du type de média dispose par défaut d'un délai d'expiration de 65 secondes (configurable dans *Administration* > *Général* > *Délais d'expiration*).

Voir également les détails des tests pour :

- E-mail
- Notification push
- Webhook
- Script

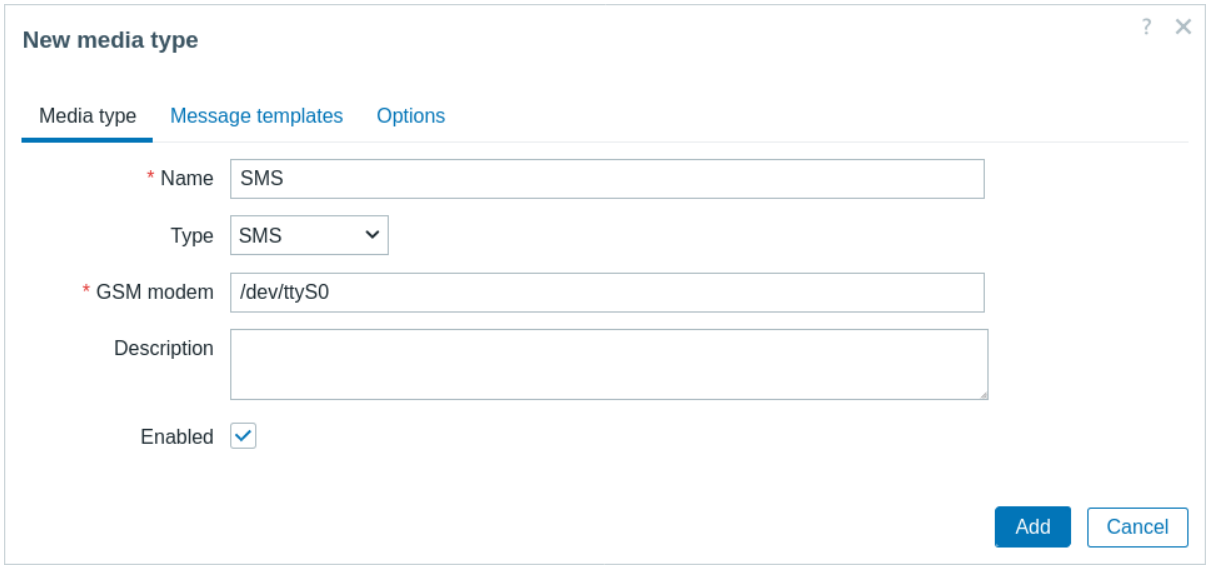
Configuration

Pour créer un type de média dans l'interface Zabbix :

1. Accédez à *Alertes* > *Types de média*.
2. Cliquez sur *Créer un type de média*.
3. Saisissez les paramètres du type de média dans le formulaire.

Certains paramètres sont **communs** à toutes les méthodes de diffusion.

Note:
Si une ou plusieurs méthodes de diffusion sont spécifiées dans l'indicateur de fonctionnalité `$ZBX_FEATURE_FLAGS['media_type_denylist']` du fichier de configuration de l'interface (`zabbix.conf.php`), ces méthodes de diffusion ne sont pas disponibles dans la liste *Type*.

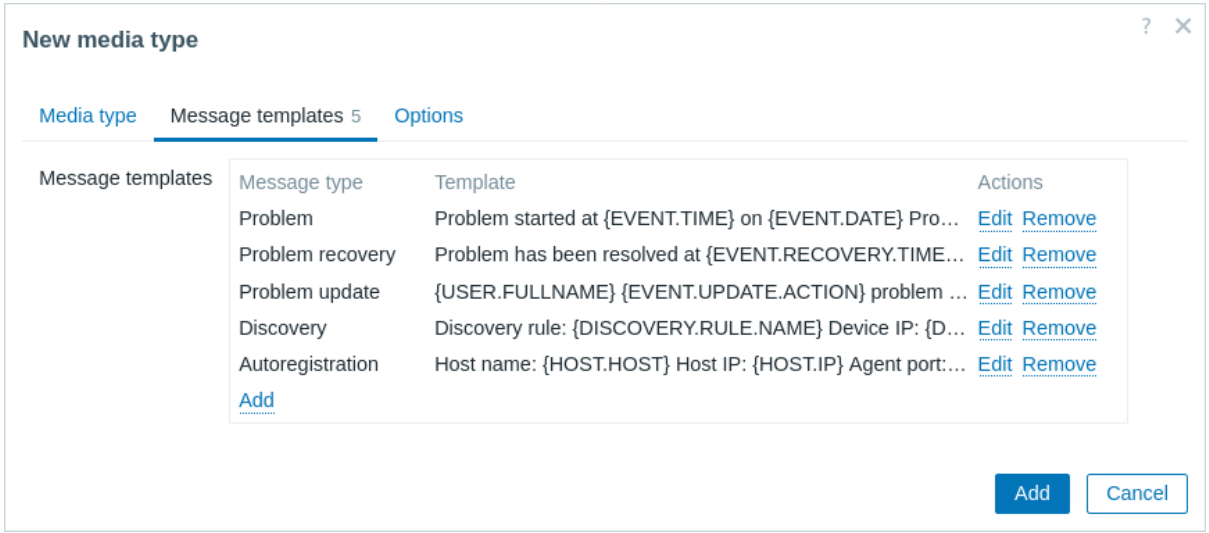


Paramètre	Description
Nom	Nom du type de média.
Type	Sélectionnez la méthode de diffusion pour le type de média.
Description	Saisissez une description du type de média.
Activé	Cochez la case pour activer le type de média.

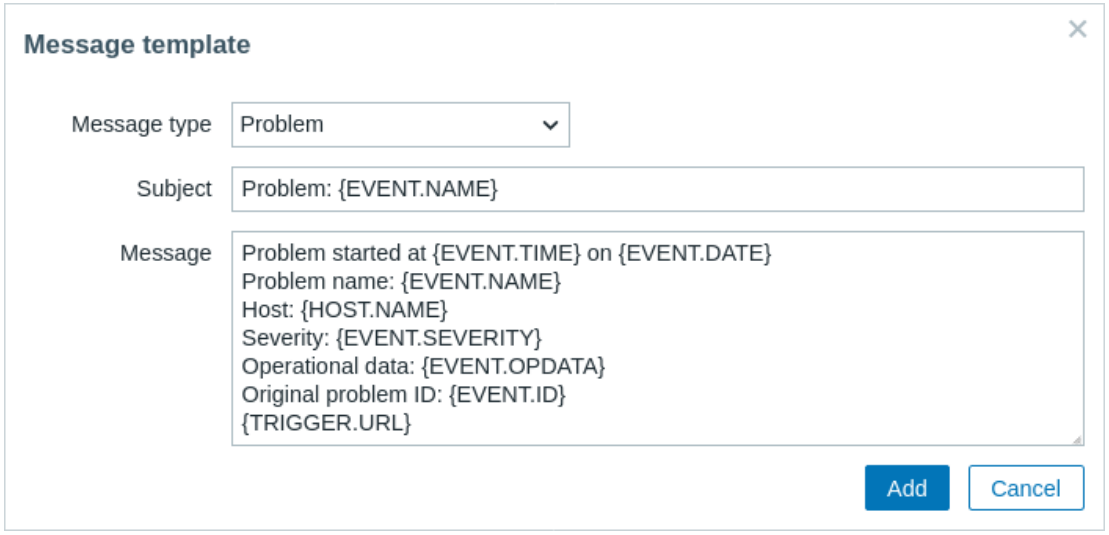
Pour connaître les paramètres spécifiques à chaque méthode, consultez les pages e-mail, notification push, SMS, script d'alerte personnalisé ou webhook.

Modèles de message

L'onglet **Modèles de message** contient les messages par défaut selon le type d'événement (problème, récupération de problème, découverte, etc).



Cliquez sur **Ajouter** pour définir un message par défaut (ou sur **Modifier** pour mettre à jour un message existant) :



Parameter	Description
Message type	Type d'un événement pour lequel le message par défaut doit être utilisé. Un seul message par défaut peut être défini pour chaque type d'événement.
Subject	Objet du message par défaut. L'objet peut contenir des macros. Il est limité à 255 caractères. L'objet n'est pas disponible pour le type de média SMS.
Message	Le message par défaut. Il est limité à un certain nombre de caractères selon le type de base de données (voir Sending messages pour plus d'informations). Le message peut contenir des macros prises en charge. Dans les messages de problème et de mise à jour de problème, les macros d'expression sont prises en charge (par exemple, `{?avg(/host/key,1h)}`).

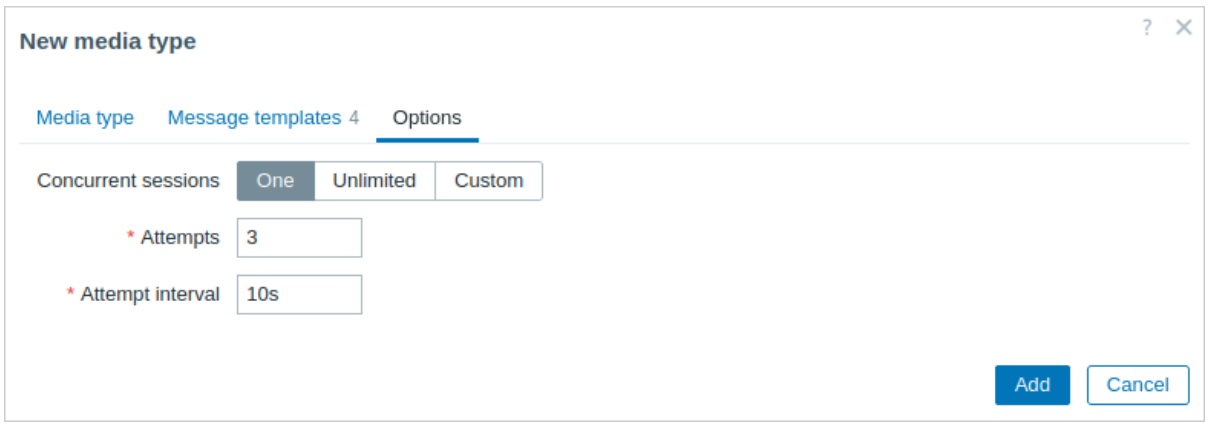
Notez que les messages par défaut sont remplacés par des messages personnalisés, s'ils sont définis dans les opérations d'action.

Warning:
La définition de modèles de message est obligatoire pour toutes les méthodes de livraison, y compris les webhooks ou les scripts d'alerte personnalisés qui n'utilisent pas de messages par défaut pour les notifications. Par exemple, l'action "Send message to Pushover webhook" échouera à envoyer des notifications de problème si le message de problème pour le webhook Pushover n'est pas défini.

Options

L'onglet **Options** contient les paramètres de traitement des alertes.
Le même ensemble d'options peut être configuré pour chaque type de média.

Tous les types de média sont traités en parallèle.
Bien que le nombre maximal de sessions simultanées soit configurable pour chaque type de média,
le nombre total de processus alerter sur le serveur ne peut être limité que par le paramètre StartAlerters.
Les alertes générées par un déclencheur sont traitées séquentiellement.
Ainsi, plusieurs notifications ne peuvent être traitées simultanément que si elles sont générées par plusieurs déclencheurs.



Paramètre	Description
Sessions simultanées	Sélectionnez le nombre de sessions alerter parallèles pour le type de média : **Une** - une session **Illimité** - nombre illimité de sessions **Personnalisé** - sélectionnez un nombre personnalisé de sessions Des valeurs illimitées ou élevées signifient davantage de sessions parallèles et une capacité accrue d'envoi des notifications. Des valeurs illimitées ou élevées doivent être utilisées dans les grands environnements où de nombreuses notifications peuvent devoir être envoyées simultanément. Si le nombre de notifications à envoyer est supérieur au nombre de sessions simultanées, les notifications restantes sont mises en file d'attente ; elles ne sont pas perdues.
Tentatives	Nombre de tentatives d'envoi d'une notification. Jusqu'à 100 tentatives peuvent être spécifiées ; la valeur par défaut est « 3 ». Si « 1 » est spécifié, Zabbix n'enverra la notification qu'une seule fois et ne réessaiera pas en cas d'échec de l'envoi.
Intervalle entre les tentatives	Fréquence des tentatives de renvoi d'une notification en cas d'échec de l'envoi, en secondes (0-3600). Si « 0 » est spécifié, Zabbix réessaiera immédiatement. Les suffixes de temps sont pris en charge (par exemple, `30s`, `1m`, `1h`).

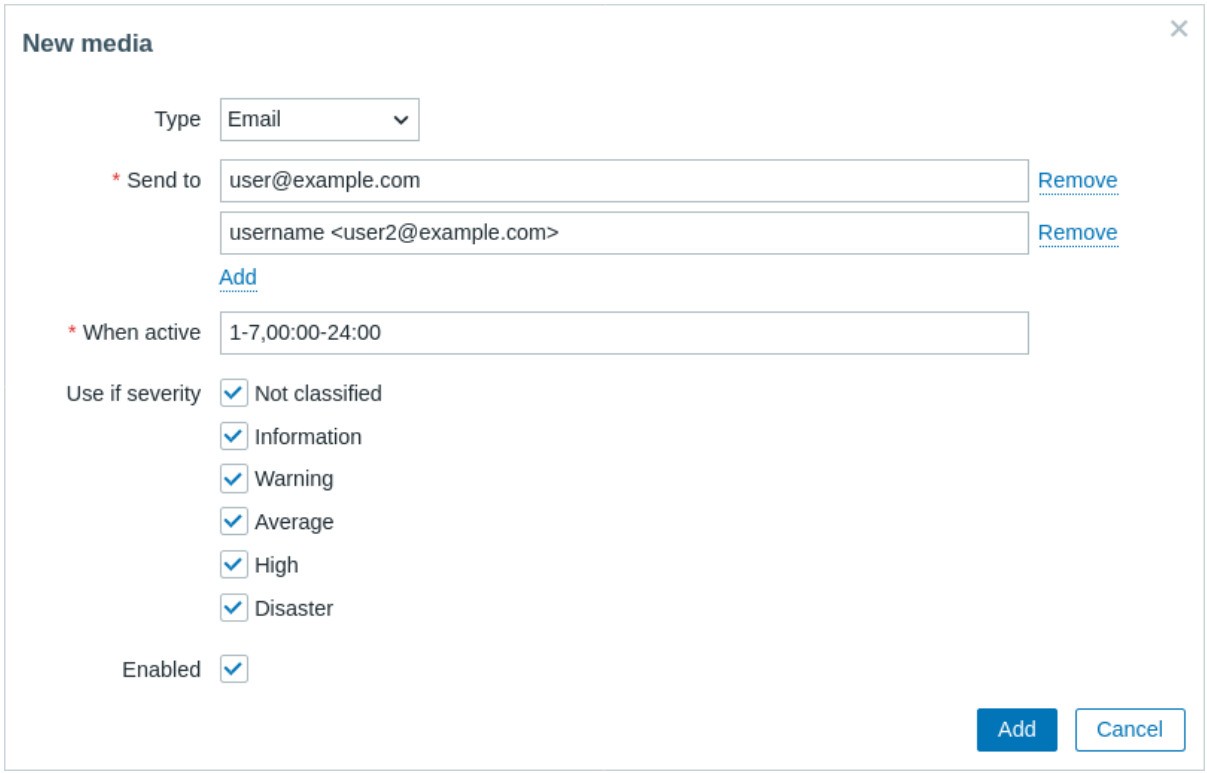
Médias utilisateur

Alors que les types de médias définissent **comment** une notification sera envoyée, les médias utilisateur définissent **où** la notification doit être envoyée.

Les médias utilisateur (par exemple, une adresse e-mail, un identifiant utilisateur de webhook, etc.) doivent être définis dans le profil utilisateur, quelle que soit la méthode d'envoi. Une action envoyant des messages à l'utilisateur *Admin* à l'aide du webhook *X* ne pourra pas les envoyer si les détails d'envoi du webhook *X* ne sont pas définis dans le profil de l'utilisateur Admin.

Pour définir les médias utilisateur :

1. Accédez à *Utilisateurs* > *Utilisateurs* et ouvrez le formulaire des propriétés de l'utilisateur. Vous pouvez également accéder à *Paramètres utilisateur* > *Notifications* dans votre propre profil utilisateur.
2. Cliquez sur *Ajouter* dans l'onglet *Média*.
3. Saisissez les paramètres du média utilisateur dans le formulaire.



Paramètre	Description
Type	La liste déroulante contient les noms des types de médias activés. Notez que lors de la modification d'un média dont le type est désactivé, le type est affiché en rouge.

Paramètre	Description
Envoyer à	Saisissez les coordonnées auxquelles les messages doivent être envoyés. Pour le type de média e-mail, plusieurs adresses peuvent être ajoutées en cliquant sur le bouton Add situé sous le champ d'adresse. Dans ce cas, les notifications seront envoyées à toutes les adresses indiquées. Pour obtenir des exemples d'adresses, consultez la description du paramètre *E-mail* du type de média e-mail. Pour le type de média de notification push, vous pouvez sélectionner les appareils mobiles actifs de l'utilisateur ou ajouter un ou plusieurs appareils spécifiques.
Actif pendant	Vous pouvez limiter la période pendant laquelle les messages sont envoyés, par exemple aux seuls jours ouvrés (1-5,09:00-18:00). Notez que cette limite est basée sur le fuseau horaire de l'utilisateur. Si le fuseau horaire de l'utilisateur est modifié et diffère de celui du système, cette limite devra peut-être être ajustée en conséquence afin de ne pas manquer de messages importants. Consultez la page Spécification des périodes pour obtenir une description du format. Les macros utilisateur sont prises en charge.
Utiliser si la sévérité est	Cochez les cases correspondant aux sévérités des déclencheurs pour lesquelles vous souhaitez recevoir des notifications. *Remarque* : la sévérité par défaut (« Non classé ») **doit être** cochée si vous souhaitez recevoir des notifications pour les événements qui ne sont pas liés à un déclencheur. Après l'enregistrement, les sévérités des déclencheurs sélectionnées seront affichées dans les couleurs de sévérité correspondantes, tandis que les sévérités non sélectionnées seront grisées.
Statut	Statut du média utilisateur. **Activé** - est utilisé. **Désactivé** - n'est pas utilisé.

E-mail

Vue d'ensemble

Pour configurer l'e-mail comme canal d'envoi des messages, vous devez configurer l'e-mail comme type de média et attribuer des adresses spécifiques aux utilisateurs.

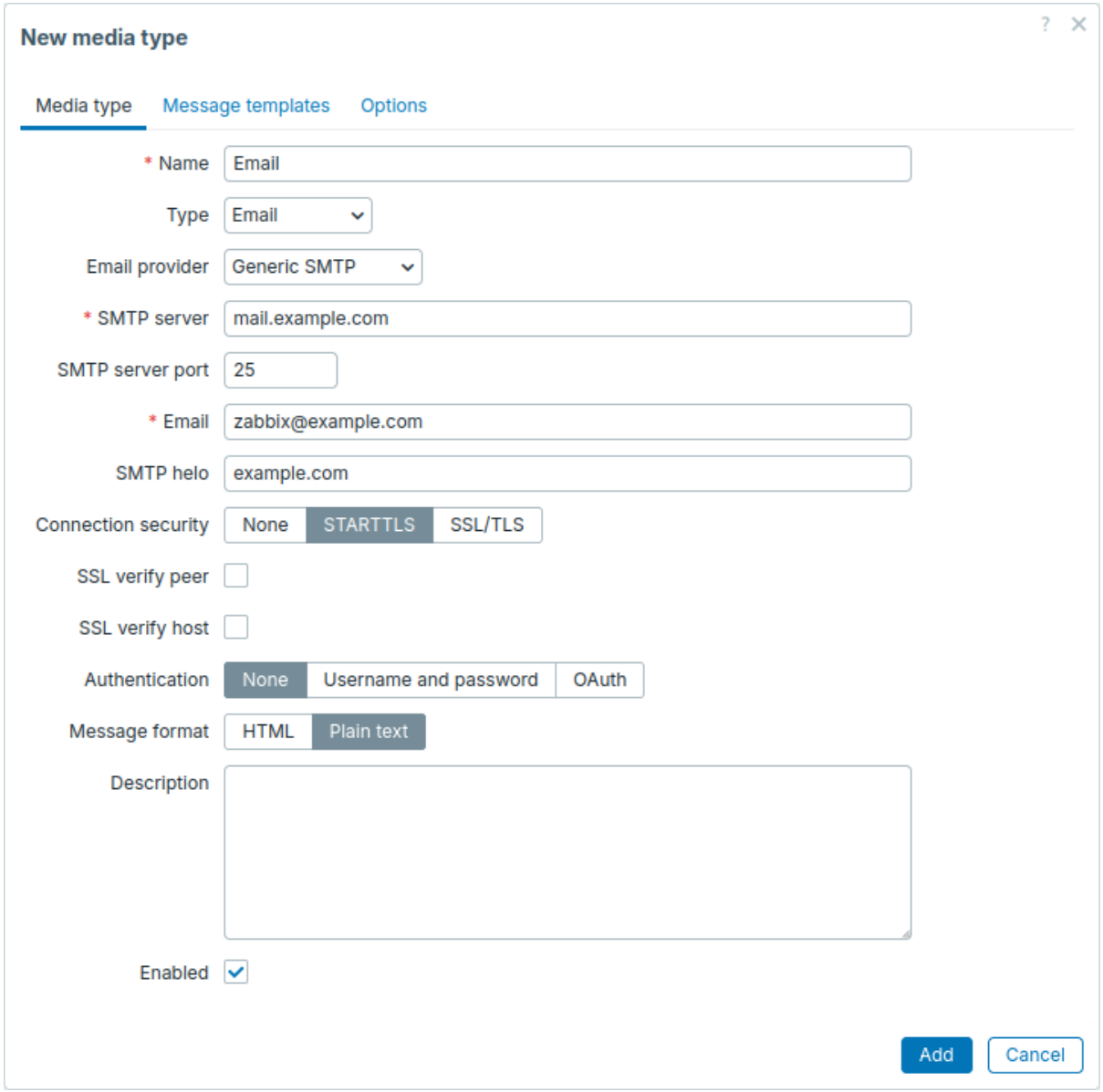
> **Note:**
> Plusieurs notifications pour un même événement seront regroupées dans le même fil de discussion e-mail.

Configuration

Pour configurer l'e-mail comme type de média :

1. Accédez à *Alertes* > *Types de média*.
2. Cliquez sur *Créer un type de média* (ou cliquez sur *E-mail* dans la liste des types de média prédéfinis).
3. Saisissez les paramètres du type de média e-mail dans le formulaire.

L'onglet **Type de média** contient les attributs généraux du type de média :



Tous les champs de saisie obligatoires sont marqués d'un astérisque rouge.

Warning:
Le mot de passe est exporté en clair lors de l'exportation de la configuration du type de média e-mail.

Les paramètres suivants sont spécifiques au type de média e-mail :

Paramètre	Description
Fournisseur d'e-mail	Sélectionnez le fournisseur d'e-mail : *Generic SMTP*, *Gmail*, *Gmail relay*, *Office365* ou *Office365 relay*. Si vous sélectionnez une option liée à Gmail ou Office365, vous devrez uniquement fournir l'adresse e-mail de l'expéditeur et le mot de passe ; les options telles que *Serveur SMTP*, *Port du serveur SMTP*, *SMTP helo* et *Sécurité de la connexion* seront automatiquement renseignées par Zabbix. Voir également : Types de média Gmail/Office365 automatisés.
Serveur SMTP	Définissez un serveur SMTP pour gérer les messages sortants. Ce champ est disponible si *Generic SMTP* est sélectionné comme fournisseur d'e-mail.
Port du serveur SMTP	Définissez le port du serveur SMTP pour gérer les messages sortants. Ce champ est disponible si *Generic SMTP* est sélectionné comme fournisseur d'e-mail.

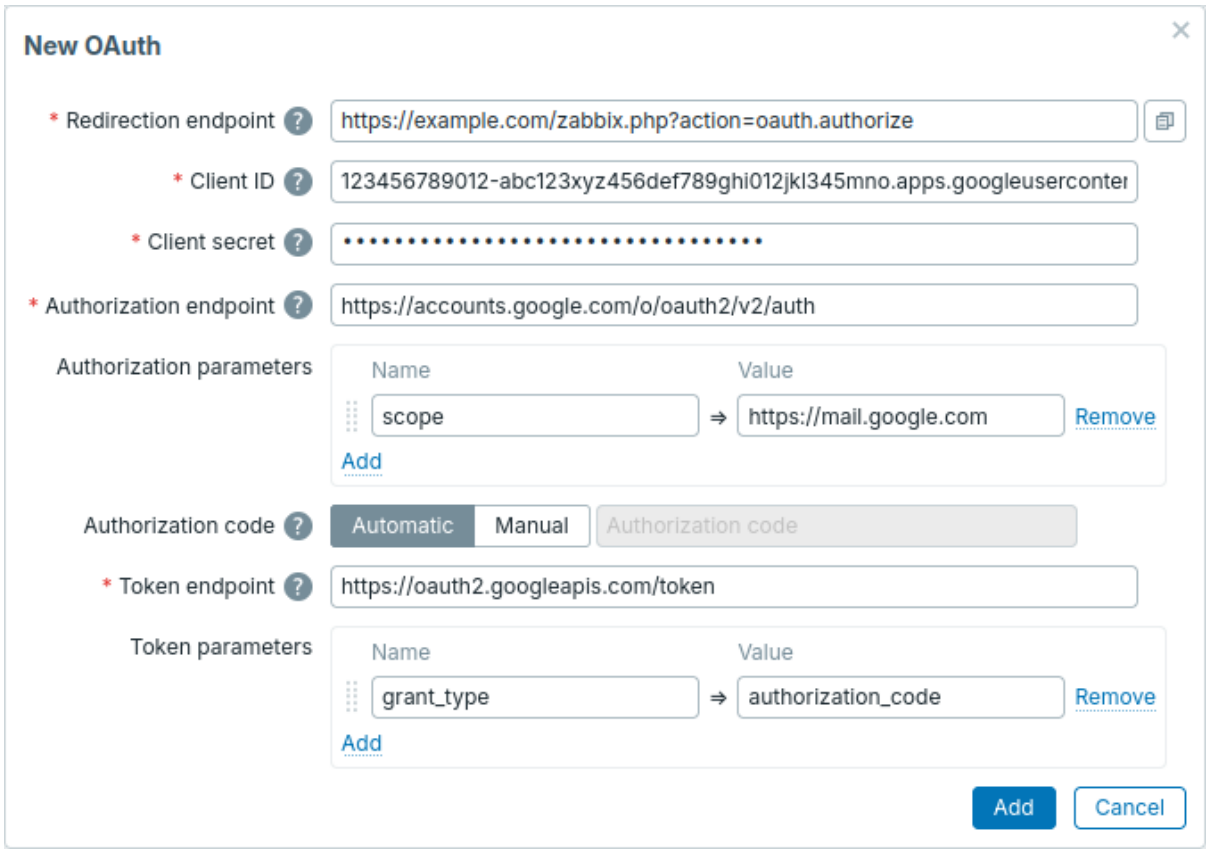
Paramètre	Description
E-mail	L'adresse saisie ici sera utilisée comme adresse **From** pour les messages envoyés. L'ajout d'un nom d'affichage de l'expéditeur avec l'adresse e-mail réelle (par exemple, `Zabbix_info <zabbix@company.com>`) est pris en charge. Certaines restrictions s'appliquent aux noms d'affichage dans les e-mails Zabbix par rapport à ceux autorisés par la RFC 5322, comme l'illustrent les exemples suivants. Exemples valides : `zabbix@company.com` (uniquement une adresse e-mail ; les chevrons ne sont pas obligatoires) `Zabbix_info <zabbix@company.com>` (un nom d'affichage suivi de l'adresse e-mail entre chevrons) `Ω-monitoring <zabbix@company.com>` (caractères UTF-8 dans le nom d'affichage) Exemples invalides : `Zabbix HQ zabbix@company.com` (un nom d'affichage est présent, mais l'adresse e-mail n'est pas entourée de chevrons) `"Zabbix@<H(comment)Q>" <zabbix@company.com>` (bien que valide selon la RFC 5322, les paires entre guillemets et les commentaires ne sont pas pris en charge dans les e-mails Zabbix).
SMTP helo	Définissez une valeur SMTP helo correcte, généralement un nom de domaine. Si ce champ est vide, le nom de domaine de l'e-mail sera envoyé (c'est-à-dire ce qui suit `@` dans le champ *E-mail*). S'il est impossible de récupérer le nom de domaine, un avertissement de niveau debug sera enregistré et le nom d'hôte du serveur sera envoyé comme domaine pour la commande HELO. Ce champ est disponible si *Generic SMTP* est sélectionné comme fournisseur d'e-mail.
Sécurité de la connexion	Sélectionnez le niveau de sécurité de la connexion : **Aucune** - ne pas utiliser l'option CURLOPT_USE_SSL **STARTTLS** - utiliser l'option CURLOPT_USE_SSL avec la valeur CURLUSESSL_ALL **SSL/TLS** - l'utilisation de CURLOPT_USE_SSL est facultative.
Vérifier le pair SSL	Cochez la case pour vérifier le certificat SSL du serveur SMTP. La valeur de la directive de configuration du serveur « SSLCALocation » doit être placée dans CURLOPT_CAPATH pour la validation du certificat. Cette option définit l'option cURL CURLOPT_SSL_VERIFYPEER.
Vérifier l'hôte SSL	Cochez la case pour vérifier que le champ *Common Name* ou le champ *Subject Alternate Name* du certificat du serveur SMTP correspond. Cette option définit l'option cURL CURLOPT_SSL_VERIFYHOST.
Authentification	Sélectionnez le niveau d'authentification : **Aucune** - aucune option cURL n'est définie **Nom d'utilisateur et mot de passe** - implique « AUTH=* », laissant à cURL le choix du mécanisme d'authentification **OAuth** - authentification OAuth L'authentification OAuth n'est pas prise en charge pour le fournisseur d'e-mail *Office365 relay*.
Nom d'utilisateur	Nom d'utilisateur à utiliser pour l'authentification. Cette option définit la valeur de CURLOPT_USERNAME. Les macros utilisateur sont prises en charge.
Mot de passe	Mot de passe à utiliser pour l'authentification. Cette option définit la valeur de CURLOPT_PASSWORD. Les macros utilisateur sont prises en charge. La valeur saisie ici est exportée telle quelle lors de l'exportation de la configuration du type de média. Pour éviter d'exposer des identifiants sensibles dans les fichiers exportés, utilisez une macro utilisateur (par exemple, `{$EMAIL_SMTP_PASSWORD}`) au lieu d'un mot de passe littéral : la référence à la macro sera exportée, tandis que la valeur secrète devra être (re)configurée sur le système de destination.
Jetons OAuth	Cliquez sur *Configurer* pour définir les paramètres de récupération des jetons OAuth dans une nouvelle fenêtre. Ce champ est disponible uniquement si « OAuth » est sélectionné dans le champ *Authentification*.
Format du message	Sélectionnez le format du message : **HTML** - envoyer au format HTML ; **Texte brut** - envoyer au format texte brut.

Attention:
Pour activer l'authentification SMTP, le serveur Zabbix doit être compilé avec l'option `--with-libcurl` de compilation, qui nécessite libcurl (voir les exigences du serveur pour connaître les détails de version).

Voir également les paramètres communs des types de média pour plus d'informations sur la configuration des messages par défaut et des options de traitement des alertes.

Jetons OAuth

Les paramètres suivants sont requis pour récupérer des jetons OAuth :



Warning:
Le secret client est exporté en texte clair lors de l'export de la configuration du type de média email.

Parameter	Description
Redirection endpoint	Saisissez l'URL de l'interface Zabbix vers laquelle le service OAuth redirigera la demande d'autorisation OAuth (au format `https://<zabbix-frontend-url>/zabbix.php?action=oauth.authorize`). Elle est définie automatiquement pour un nouveau type de média email lorsque l'URL de l'interface est définie.
Client ID	Saisissez l'identifiant unique de l'application cliente enregistrée auprès du serveur d'autorisation OAuth.
Client secret	Saisissez le secret privé de l'application cliente enregistrée auprès du serveur d'autorisation OAuth.
Authorization endpoint	Saisissez l'URL du serveur d'autorisation OAuth pour demander l'autorisation de l'utilisateur.
Authorization parameters	Saisissez les paramètres du point de terminaison d'autorisation.
Authorization code	Saisissez le code d'autorisation : **Automatic** - le code sera récupéré automatiquement via une page de redirection ; **Manual** - saisissez le code manuellement si la récupération automatique échoue.
Token endpoint	Saisissez l'URL du serveur d'autorisation OAuth pour échanger le code d'autorisation contre des jetons d'accès et de rafraîchissement.
Token parameters	Saisissez les paramètres du jeton d'accès.

Note:
La récupération des jetons OAuth dispose de fonctionnalités automatisées pour les fournisseurs de messagerie **Gmail**, **Gmail relay** et **Office365**. Il suffit de renseigner les valeurs des paramètres *Redirection endpoint*, *Client ID* et *Client secret*. Zabbix remplira automatiquement les autres valeurs requises (voir Valeurs par défaut des URL OAuth selon le fournisseur).

Récupération des jetons d'accès et d'actualisation OAuth

Après avoir soumis les paramètres OAuth :

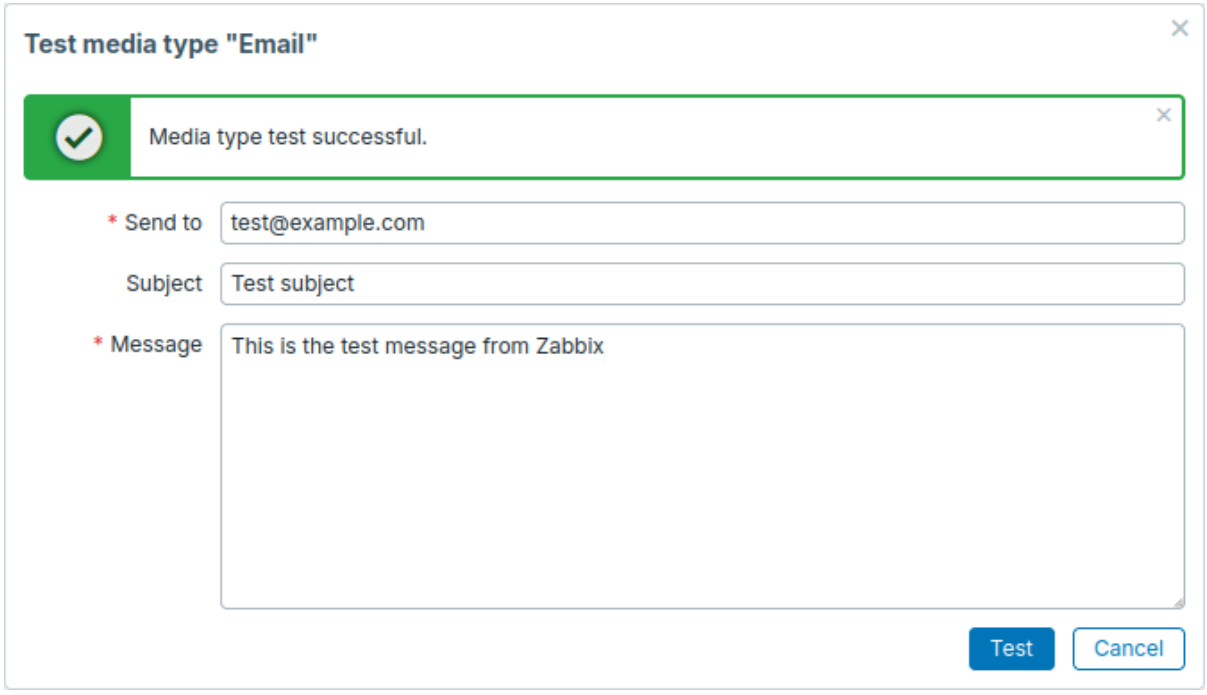
1. Une fenêtre contextuelle du navigateur s'ouvre et redirige l'utilisateur vers le *point de terminaison d'autorisation*.
2. L'utilisateur autorise Zabbix dans le service OAuth.
3. Le service OAuth redirige l'utilisateur vers l'action du frontend Zabbix `oauth.authorize` avec le code d'autorisation et la valeur de portée.
4. En réponse, Zabbix enverra une requête au *point de terminaison de jeton* afin d'échanger le code d'autorisation contre des jetons d'accès et d'actualisation.

Test

Pour vérifier qu'un type de média e-mail configuré fonctionne correctement :

1. Repérez l'e-mail concerné dans la liste des types de média.
2. Cliquez sur *Test* dans la dernière colonne de la liste (une fenêtre de test s'ouvrira).
3. Saisissez une adresse de destinataire dans *Send to*, le corps du message et, éventuellement, l'objet.
4. Cliquez sur *Test* pour envoyer un message de test.

Un message indiquant la réussite ou l'échec du test s'affichera dans la même fenêtre :



Média utilisateur

Une fois le type de média e-mail configuré, accédez à la section *Users* > *Users* et modifiez le profil de l'utilisateur pour lui attribuer un média e-mail. Les étapes de configuration du média utilisateur, communes à tous les types de médias, sont décrites sur la page Media types.

Types de média Gmail/Office365 automatisés

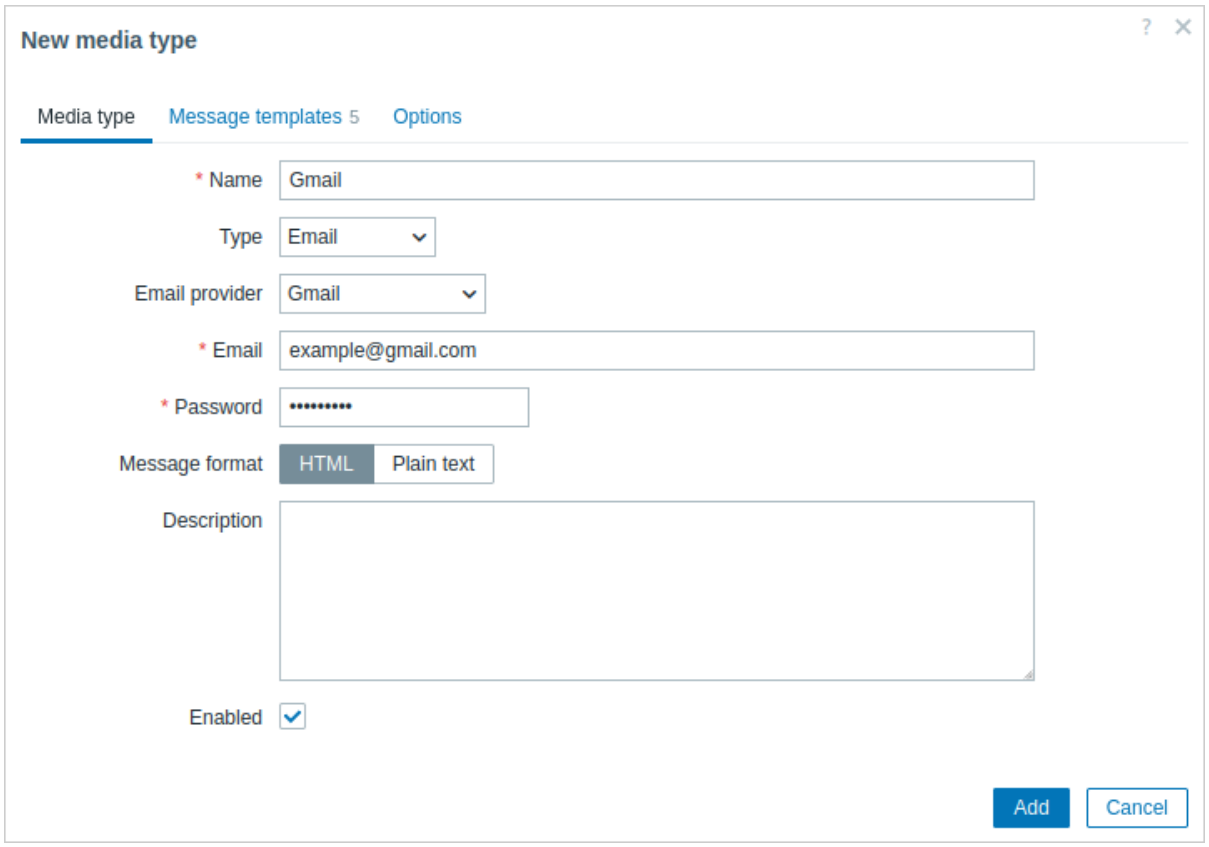
Vue d'ensemble

Les utilisateurs de Gmail ou d'Office365 peuvent bénéficier de fonctionnalités automatisées dans la configuration du type de média.

Configuration

Le champ *Fournisseur de messagerie* dans la configuration du type de média de messagerie permet de sélectionner des options préconfigurées pour Gmail et Office 365.

Lors de la sélection des options associées à Gmail/Office365, il suffit de fournir l'adresse e-mail et le mot de passe de l'expéditeur pour créer un type de média fonctionnel.



Dès que l'adresse e-mail et le mot de passe sont fournis, Zabbix peut automatiquement renseigner tous les paramètres requis pour les types de média Gmail/Office365 avec les valeurs réelles/recommandées, à savoir *Serveur SMTP*, *Port du serveur SMTP*, *HELO SMTP* et *Sécurité de la connexion*. Grâce à cette automatisation, ces champs ne sont même pas affichés. Il est toutefois possible de consulter les informations relatives au serveur SMTP et à l'e-mail dans la liste des types de média (voir la colonne *Détails*).

Notez également que :

- Le mot de passe n'est pas requis pour les options de relais.
- Pour le relais Office365, le nom de domaine de l'adresse e-mail fournie sera utilisé pour renseigner dynamiquement le serveur SMTP (c'est-à-dire pour remplacer « example.com » dans `example-com.mail.protection.outlook.com` par la valeur réelle).

Jetons OAuth

La récupération des jetons OAuth dispose de fonctionnalités automatisées pour les fournisseurs de messagerie *Gmail*, *Gmail relay* et *Office365*.



Il suffit de fournir un ensemble limité de paramètres pour l'accès OAuth et la récupération du jeton d'accès et du jeton d'actualisation : *Redirection endpoint*, *Client ID* et *Client secret*. Zabbix renseignera automatiquement les autres valeurs requises (voir Valeurs par défaut des URL OAuth par fournisseur).

Notez qu'il est également possible d'utiliser le formulaire Generic SMTP pour configurer l'autorisation OAuth pour ces fournisseurs.

Note:
SmtpClientAuthentication doit être activé soit pour chaque utilisateur/boîte aux lettres partagée, soit pour l'ensemble du locataire dans Office365 (désactivé par défaut).

Valeurs par défaut des URL OAuth par fournisseur

Le tableau suivant répertorie les valeurs et paramètres par défaut des URL OAuth par fournisseur utilisés par Zabbix.

Paramètre	Gmail	Office365	SMTP générique
Point de terminaison d'autorisation	`https://accounts.google.com/o/oauth2/auth`	`https://login.microsoftonline.com/common/oauth2/v2.0/authori`	Aucune valeur par défaut
scope	`https://mail.google.com`	`https://outlook.office.com/SMTP.Send offline_access`	Aucune valeur par défaut
access_type	`offline`	Non utilisé	Aucune valeur par défaut
prompt	`consent`	Non utilisé	Aucune valeur par défaut
redirect_uri	La valeur du champ *Point de redirection* est ajoutée automatiquement par le code. Elle n'est pas répertoriée dans les paramètres du *Point de terminaison d'autorisation*.		
state	Un hachage unique permettant d'identifier le média mis à jour avec le jeton OAuth est ajouté automatiquement par le code. Il n'est pas répertorié dans les paramètres du *Point de terminaison d'autorisation*.		
Point de terminaison de jeton	`https://oauth2.googleapis.com/token`	`https://login.microsoftonline.com/common/oauth2/v2.0/token`	Aucune valeur par défaut
grant_type	`authorization_code`	`authorization_code`	Aucune valeur par défaut
redirect_uri	La valeur du champ *Point de redirection* est ajoutée automatiquement par le code. Elle n'est pas répertoriée dans les paramètres du *Point de terminaison de jeton*.		
state	Un hachage unique permettant d'identifier le média mis à jour avec le jeton OAuth est ajouté automatiquement par le code. Il n'est pas répertorié dans les paramètres du *Point de terminaison de jeton*.		

Solution de contournement Office365 pour les comptes personnels

Les comptes personnels Office365 ne prennent pas en charge les chaînes de requête dans l'URL de redirection.

Pour contourner ce problème, l'administrateur du serveur web Apache peut ajouter la règle de réécriture suivante au fichier `.htaccess` :

```
RewriteEngine On
RewriteRule ^/oauth\.authorize$ /zabbix.php?action=oauth.authorize [QSA,L,PT]
```

Cela permet d'utiliser une URL de redirection simplifiée comme :

```
http://server-name/zabbix/oauth.authorize
```

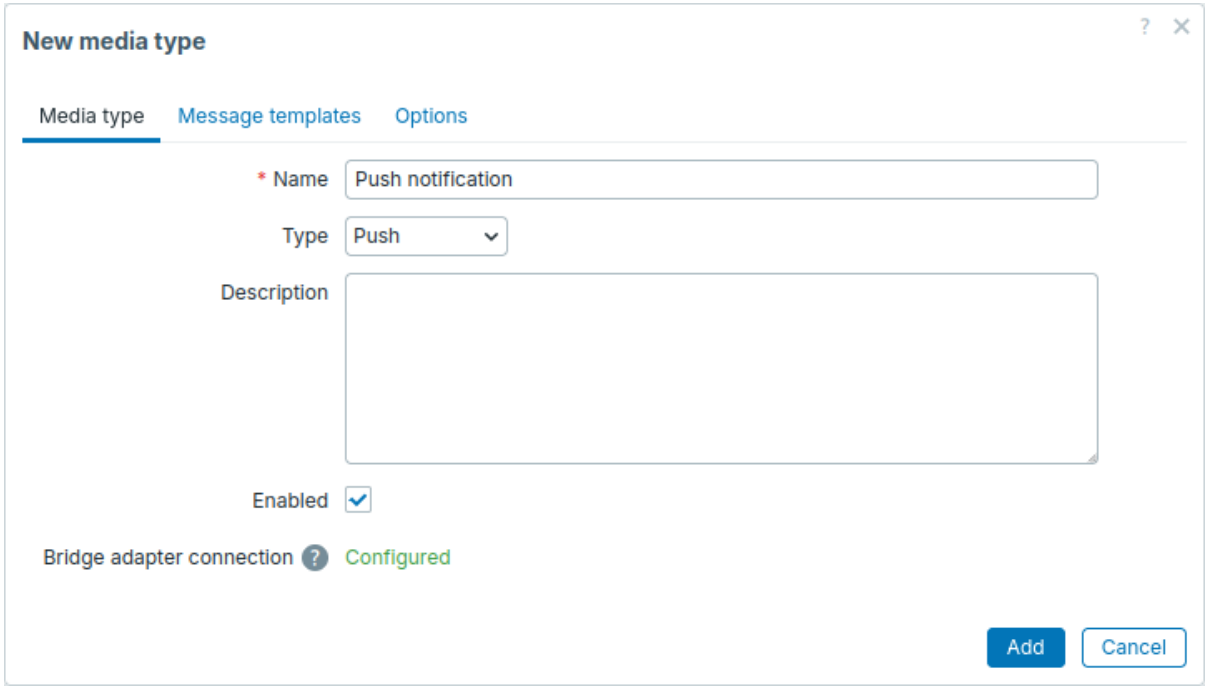
au lieu d'une URL avec des paramètres de requête, garantissant ainsi la compatibilité avec les comptes personnels Office365.

Notification push

Ce type de média permet au serveur Zabbix d'envoyer des notifications push à l'application mobile Zabbix sur votre appareil mobile.

Configuration

Lors de la création d'un type de média, sélectionnez *Push* comme *Type* :



Le formulaire de configuration contient les paramètres, les modèles de message et les options communs aux types de média.

Le formulaire indique également si l'adaptateur Zabbix bridge est configuré.
 Si l'adaptateur n'est pas configuré, vous pouvez tout de même créer le type de média, mais celui-ci ne fonctionnera pas.

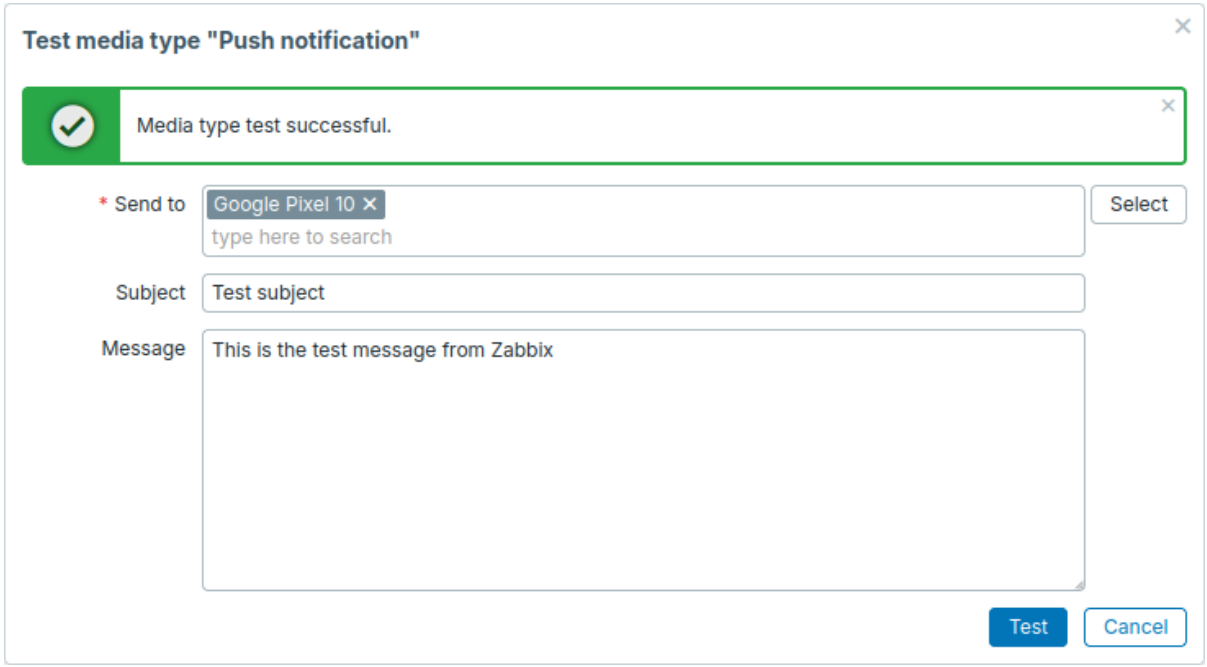
Après avoir créé et activé le type de média, ajoutez un appareil pour un utilisateur afin d'ajouter automatiquement le type de média à cet utilisateur (ou ajoutez manuellement un média utilisateur). Cela lui permettra de recevoir des notifications push sur son appareil mobile.

Test

Pour vérifier que ce type de média fonctionne :

1. Recherchez le type de média dans la liste des types de média.
2. Cliquez sur *Test* dans la dernière colonne de la liste pour ouvrir le formulaire de test.
3. Dans le champ *Envoyer à*, sélectionnez un appareil.
4. Cliquez sur *Test* pour envoyer une notification push de test à l'appareil sélectionné.

Un message de réussite ou d'échec s'affichera dans la même fenêtre :



SMS

Aperçu

Zabbix prend en charge l'envoi de messages SMS à l'aide d'un modem GSM série connecté au port série du serveur Zabbix.

Assurez-vous que :

- La vitesse du périphérique série (généralement /dev/ttyS0 sous Linux) correspond à celle du modem GSM. Zabbix ne définit pas la vitesse de la liaison série. Il utilise les paramètres par défaut.
- L'utilisateur 'zabbix' dispose d'un accès en lecture/écriture au périphérique série. Exécutez la commande ls -l /dev/ttyS0 pour voir les autorisations actuelles du périphérique série.
- Le modem GSM a son code PIN saisi et le conserve après une remise sous tension. Vous pouvez également désactiver le code PIN sur la carte SIM. Le code PIN peut être saisi en exécutant la commande AT+CPIN="NNNN" (NNNN est votre code PIN, les guillemets doivent être présents) dans un logiciel de terminal, tel que Unix minicom ou Windows HyperTerminal.

Zabbix a été testé avec les modems GSM suivants :

- Siemens MC35
- Teltonika ModemCOM/G10

Pour configurer les SMS comme canal de distribution des messages, vous devez également configurer les SMS comme type de média et saisir les numéros de téléphone correspondants pour les utilisateurs.

Configuration

Pour configurer les SMS comme type de média :

1. Accédez à *Alertes* > *Types de média*.
2. Cliquez sur *Créer un type de média* (ou cliquez sur *SMS* dans la liste des types de média prédéfinis).
3. Saisissez les paramètres du type de média SMS dans le formulaire.

Les paramètres suivants sont spécifiques au type de média SMS :

Paramètre	Description
Modem GSM	Définissez le nom du périphérique série du modem GSM. Le chemin saisi ici sera validé par rapport au paramètre du serveur SMSDevices (s'il est spécifié).

Consultez les paramètres communs des types de média pour plus d'informations sur la configuration des messages par défaut et des options de traitement des alertes. Notez que le traitement parallèle de l'envoi des notifications SMS n'est pas possible.

Médias utilisateur

Une fois le type de média SMS configuré, accédez à la section *Utilisateurs* > *Utilisateurs* et modifiez le profil utilisateur pour attribuer le média SMS à l'utilisateur. Les étapes de configuration des médias utilisateur, communes à tous les types de médias, sont décrites sur la page Types de médias.

Scripts d'alerte personnalisés

Vue d'ensemble

Si vous n'êtes pas satisfait des types de média existants pour l'envoi des alertes, il existe une autre façon de le faire. Vous pouvez créer un script qui gérera la notification à votre manière.

Les scripts d'alerte personnalisés sont exécutés sur le serveur Zabbix. Ces scripts doivent se trouver dans le répertoire spécifié par le paramètre AlertScriptsPath du fichier de configuration du serveur.

Voici un exemple de script d'alerte personnalisé :

```
#####!/bin/bash

to=$1
subject=$2
body=$3
host=$4
value=$5

cat <<EOF | mail -s "$subject" "$to"
$body

Host: $host
Value: $value
EOF
```

Attention:
Zabbix vérifie le code de sortie des commandes et scripts exécutés. Tout code de sortie différent de **0** est considéré comme une erreur de command execution. Dans de tels cas, Zabbix tentera de répéter l'exécution ayant échoué.

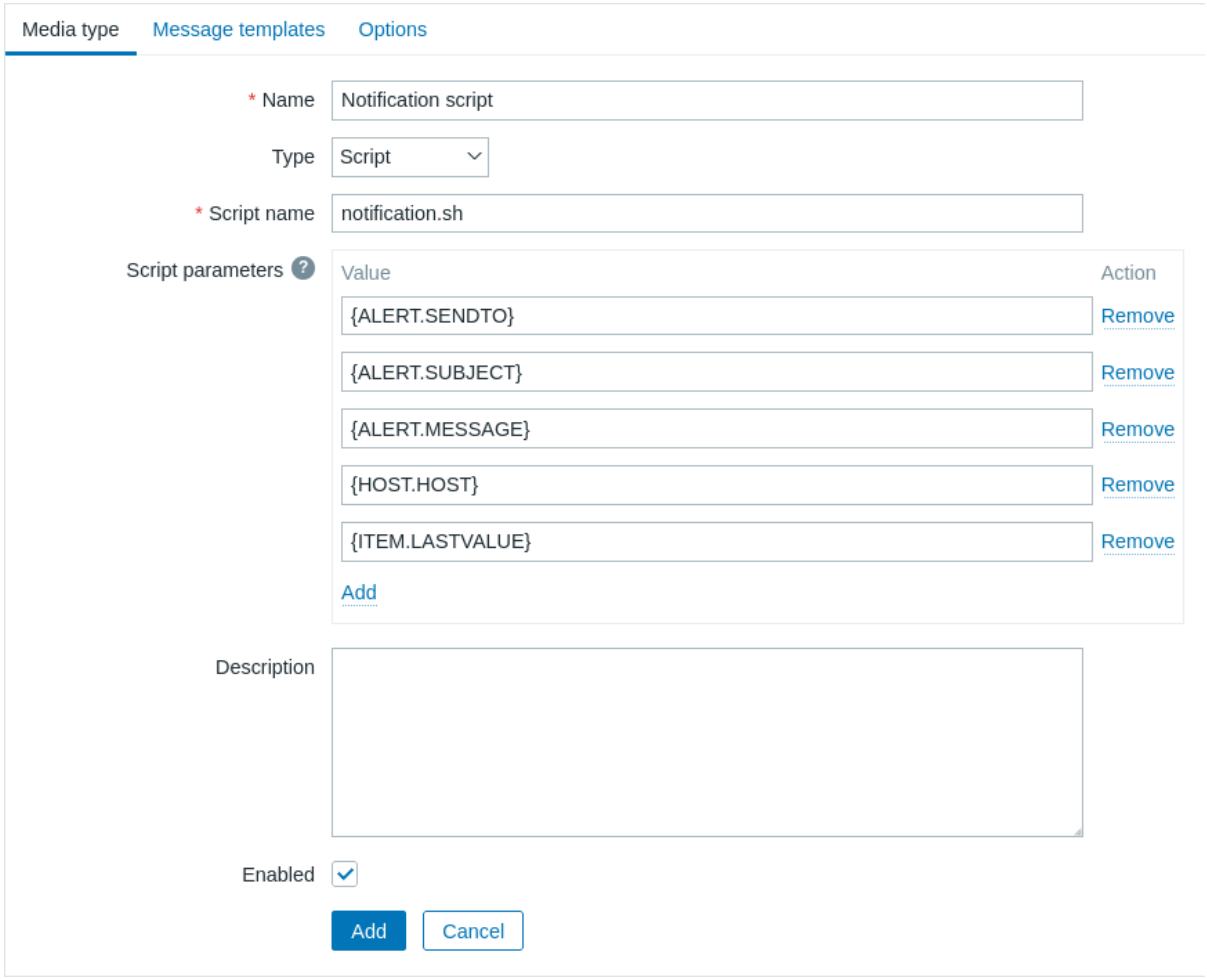
Les variables d'environnement ne sont ni conservées ni créées pour le script, elles doivent donc être gérées explicitement.

Configuration

Pour configurer des scripts d'alerte personnalisés en tant que type de média :

1. Accédez à *Alertes* > *Types de média*.
2. Cliquez sur *Créer un type de média*.
3. Saisissez les paramètres du type de média de script dans le formulaire.

L'onglet **Type de média** contient les attributs généraux du type de média :



Tous les champs de saisie obligatoires sont marqués d'un astérisque rouge.

Les paramètres suivants sont spécifiques au type de média de script :

Paramètre	Description
Nom du script	Saisissez le nom du fichier de script (par exemple, notification.sh) situé dans le répertoire spécifié dans le paramètre de configuration du serveur AlertScriptsPath.
Paramètres du script	Ajoutez des paramètres de script facultatifs qui seront transmis au script en tant qu'arguments de ligne de commande, dans l'ordre dans lequel ils sont définis. Les paramètres du script prennent en charge les macros {ALERT.SENDTO}, {ALERT.SUBJECT}, {ALERT.MESSAGE} et toutes les macros prises en charge dans les notifications, ainsi que les macros utilisateur.

Consultez les paramètres communs des types de média pour plus d'informations sur la configuration des messages par défaut et des options de traitement des alertes.

Warning:
Même si un script d'alerte n'utilise pas les messages par défaut, les modèles de message correspondant aux types d'opération utilisés par ce type de média doivent tout de même être définis. Sinon, une notification ne sera pas envoyée.

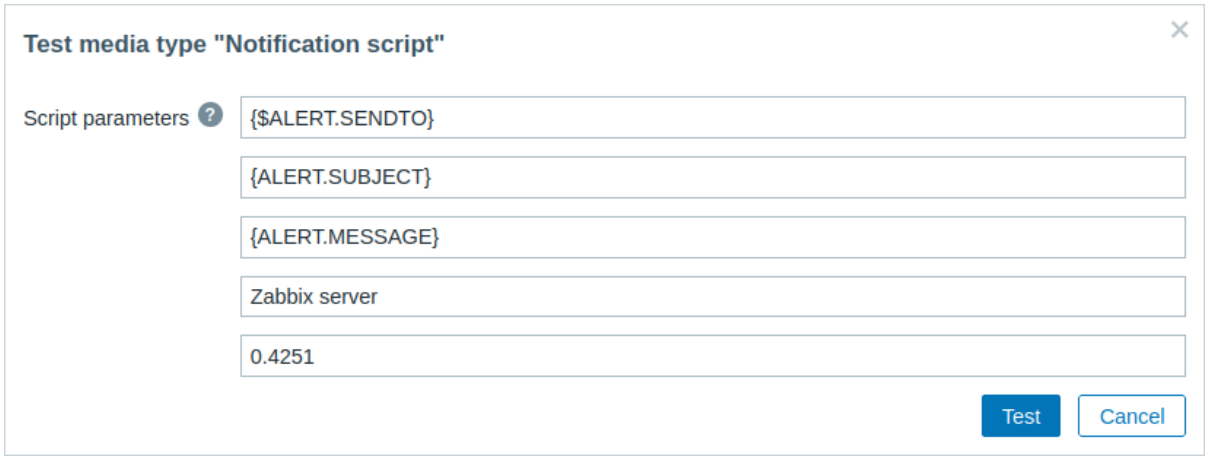
Attention:
Si plusieurs types de média de script sont configurés, ces scripts peuvent être traités en parallèle par les processus d'alerte. Le nombre total de processus d'alerte est limité par le paramètre `StartAlerters` du fichier de configuration du serveur.

Test

Pour tester un type de média de script configuré :

1. Localisez le script concerné dans la liste des types de média.

2. Cliquez sur *Test* dans la dernière colonne de la liste ; un formulaire de test s'ouvrira dans une fenêtre contextuelle. Le formulaire de test contiendra le même nombre de paramètres que ceux configurés pour le type de média de script.
3. Modifiez les valeurs des paramètres du script si nécessaire. La modification n'affecte que la procédure de test ; les valeurs réelles ne seront pas modifiées.
4. Cliquez sur *Test*.



Note:
Lors du test d'un type de média de script configuré, {ALERT.SENDTO}, {ALERT.SUBJECT}, {ALERT.MESSAGE} et les macros utilisateur prendront leurs valeurs, mais les macros liées aux événements (par exemple, {HOST.HOST}, {ITEM.LASTVALUE}, etc.) ne seront pas résolues, car pendant le test il n'existe aucun événement associé à partir duquel obtenir les détails. Notez que les macros contenues dans les macros {ALERT.SUBJECT} et {ALERT.MESSAGE} ne seront pas non plus résolues. Par exemple, si la valeur de {ALERT.SUBJECT} est composée de "Problem: {EVENT.NAME}", alors la macro {EVENT.NAME} ne sera pas résolue.

Média utilisateur

Une fois le type de média configuré, allez dans la section *Users* > *Users* et modifiez un profil utilisateur en attribuant ce type de média à l'utilisateur. Les étapes de configuration du média utilisateur, communes à tous les types de média, sont décrites sur la page Media types.

Notez que lors de la définition du média utilisateur, le champ *Send to* ne peut pas être vide. Si ce champ n'est pas utilisé dans le script d'alerte, saisissez n'importe quelle combinaison de caractères pris en charge afin de contourner les exigences de validation.

Webhook

Vue d'ensemble

Le type de média webhook est utile pour effectuer des appels HTTP à l'aide d'un code JavaScript personnalisé afin de permettre une intégration simple avec des logiciels externes tels que des systèmes de helpdesk, des chats ou des messageries. Vous pouvez choisir d'importer une intégration fournie par Zabbix ou de créer une intégration personnalisée à partir de zéro.

Intégrations

Les intégrations suivantes sont disponibles et permettent d'utiliser des types de média webhook prédéfinis pour envoyer les notifications Zabbix vers :

- brevis.one
- Discord
- Event-Driven Ansible
- messagerie Express.ms
- GitHub
- GLPI
- IBM Maximo Service Request
- iLert
- iTop
- Jira
- Jira Service Management
- ManageEngine ServiceDesk
- Mantis Bug Tracker

- Mattermost
- MS Teams Workflows
- LINE
- Opsgenie
- OTRS CE
- Pagerduty
- Pushover
- Redmine
- Rocket.Chat
- ServiceNow
- SIGNL4
- Slack
- SolarWinds
- SysAid
- Telegram
- TOPdesk
- VictorOps
- Zammad
- Zendesk

Note:
En plus des services répertoriés ici, Zabbix peut être intégré à **Spiceworks** (aucun webhook n'est requis). Pour convertir les notifications Zabbix en tickets Spiceworks, créez un type de média e-mail et saisissez l'adresse e-mail du service d'assistance Spiceworks (par exemple, help@zabbix.on.spiceworks.com) dans les paramètres du profil d'un utilisateur Zabbix désigné.

Configuration

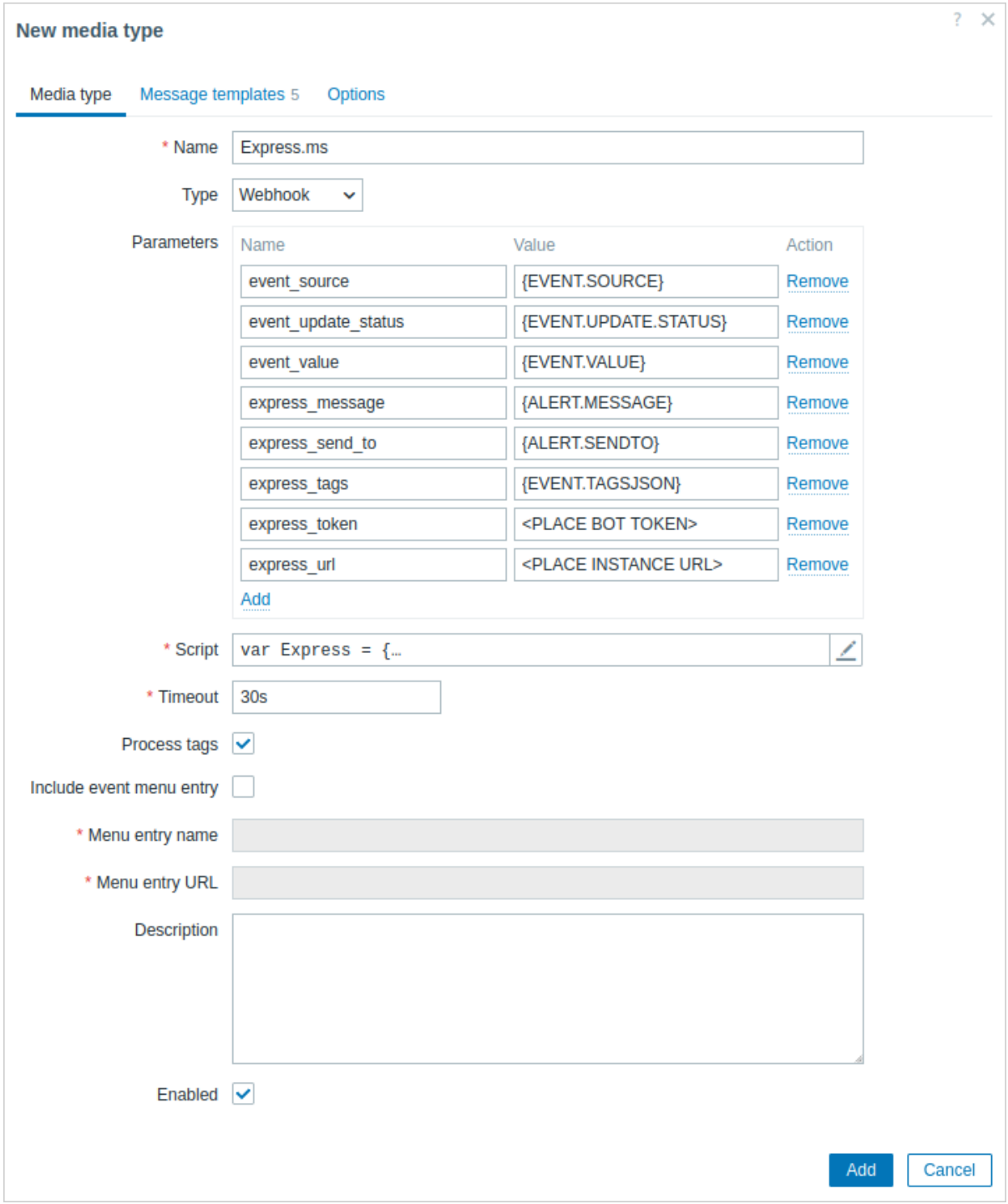
Pour commencer à utiliser une intégration webhook :

1. Localisez le fichier `.yaml` requis dans le répertoire `templates/media` de la version de Zabbix téléchargée ou téléchargez-le depuis le dépôt git de Zabbix.
2. Importez le fichier dans votre installation Zabbix. Le webhook apparaîtra dans la liste des types de média.
3. Configurez le webhook conformément aux instructions du fichier `Readme.md` (vous pouvez cliquer sur le nom d'un webhook ci-dessus pour accéder rapidement au fichier `Readme.md`).

Pour créer un webhook personnalisé à partir de zéro :

1. Accédez à *Alertes* > *Types de média*.
2. Cliquez sur *Créer un type de média*.
3. Saisissez les paramètres du type de média webhook dans le formulaire.

L'onglet **Type de média** contient différents attributs spécifiques à ce type de média :



Tous les champs de saisie obligatoires sont signalés par un astérisque rouge.

Les paramètres suivants sont spécifiques au type de média webhook :

Paramètre	Description
Paramètres	Spécifiez les variables du webhook sous forme de paires attribut-valeur. Pour les webhooks préconfigurés, la liste des paramètres varie selon le service. Consultez le fichier *Readme.md* du webhook pour obtenir la description des paramètres. Pour les nouveaux webhooks, plusieurs variables courantes sont incluses par défaut (URL:<empty>, HTTPProxy:<empty>, To:{ALERT.SENDTO}, Subject:{ALERT.SUBJECT}, Message:{ALERT.MESSAGE}) ; vous pouvez les conserver ou les supprimer. Les paramètres des webhooks prennent en charge les macros utilisateur, toutes les macros prises en charge dans les notifications de problèmes et, en outre, les macros {ALERT.SENDTO}, {ALERT.SUBJECT} et {ALERT.MESSAGE}. Si vous spécifiez un proxy HTTP, le champ prend en charge les mêmes fonctionnalités que le champ Proxy HTTP de la configuration de l'élément. La chaîne du proxy peut être précédée de `[scheme]://` afin de spécifier le type de proxy utilisé (par exemple, https, socks4, socks5 ; consultez la documentation).
Script	Saisissez le code JavaScript dans l'éditeur modal qui s'ouvre lorsque vous cliquez dans le champ du paramètre ou sur l'icône en forme de crayon située à côté. Ce code exécutera l'opération du webhook. Le script est une fonction qui accepte des paires paramètre-valeur. Les valeurs doivent être converties en objets JSON à l'aide de la méthode JSON.parse(), par exemple : `var params = JSON.parse(value);`. Le code a accès à tous les paramètres ; il peut effectuer des requêtes HTTP GET, POST, PUT et DELETE, prendre en charge des méthodes supplémentaires telles que CONNECT, PATCH, HEAD, OPTIONS et TRACE, et contrôler les en-têtes HTTP ainsi que le corps de la requête. Le script doit contenir un opérateur return, sinon il ne sera pas valide. Il peut renvoyer un état OK accompagné d'une liste facultative de tags et de valeurs de tags (voir l'option *Traiter les tags*) ou une chaîne d'erreur. Les événements de rétablissement (qu'ils soient générés automatiquement ou à la suite d'une fermeture manuelle) sont créés par le serveur et incluent les tags de l'événement résolu (y compris les tags hérités des modèles, des hôtes et des déclencheurs). Les scripts webhook sont exécutés après la création de l'alerte ; par conséquent, les tags renvoyés par un script webhook sont ajoutés uniquement après la création initiale de l'alerte et ne seront pas présents dans les macros `{EVENT.TAGS}` et `{EVENT.RECOVERY.TAGS}` du message initial du problème ou du message de rétablissement immédiat. *Remarque* : Il est recommandé d'utiliser des variables locales (par exemple, `var local = 1`) plutôt qu'une variable globale (par exemple, `global = 1`) afin de garantir que chaque script fonctionne avec ses propres données et d'éviter les collisions entre appels simultanés (voir les problèmes connus). Voir également : Directives de développement des webhooks, Exemples de scripts webhook, Objets JavaScript supplémentaires.
Délai d'expiration	Délai d'expiration de l'exécution JavaScript (1 à 60 s, 30 s par défaut). Les suffixes de temps sont pris en charge (par exemple, `30s`, `1m`).
Traiter les tags	Cochez la case pour traiter les valeurs des propriétés JSON renvoyées comme des tags. Ces tags sont ajoutés aux tags existants du problème. Notez que lors de l'utilisation des tags webhook, le webhook doit renvoyer un objet JSON contenant au moins un objet de tags vide : `var result = {tags: {}};` Exemples de tags pouvant être renvoyés : *jira-id:prod-1234*, *responsible:John Smith*, *processed:<no value>*
Inclure une entrée dans le menu des événements	Cochez la case pour inclure dans le menu des événements une entrée contenant un lien vers un ticket externe créé. Une entrée sera incluse pour chaque webhook activé dont cette case est cochée. Notez que si les paramètres *Nom de l'entrée de menu* et *URL de l'entrée de menu* contiennent des macros {EVENT.TAGS.<tag name>}, une entrée ne sera incluse que si ces macros peuvent être résolues (c'est-à-dire si l'événement possède les tags correspondants). Si cette option est sélectionnée, le webhook ne doit pas être utilisé pour envoyer des notifications à différents utilisateurs (envisagez plutôt de créer un utilisateur dédié) et ne doit pas être utilisé dans plusieurs actions d'alerte pour un seul événement de problème.

Paramètre	Description
Nom de l'entrée de menu	Spécifiez le nom de l'entrée de menu. La macro {EVENT.TAGS.<tag name>} est prise en charge. Ce champ est obligatoire uniquement si l'option *Inclure une entrée dans le menu des événements* est sélectionnée.
URL de l'entrée de menu	Spécifiez l'URL sous-jacente de l'entrée de menu. La macro {EVENT.TAGS.<tag name>} est prise en charge. Ce champ est obligatoire uniquement si l'option *Inclure une entrée dans le menu des événements* est sélectionnée.

Consultez les paramètres communs des types de média pour plus d'informations sur la configuration des messages par défaut et des options de traitement des alertes.

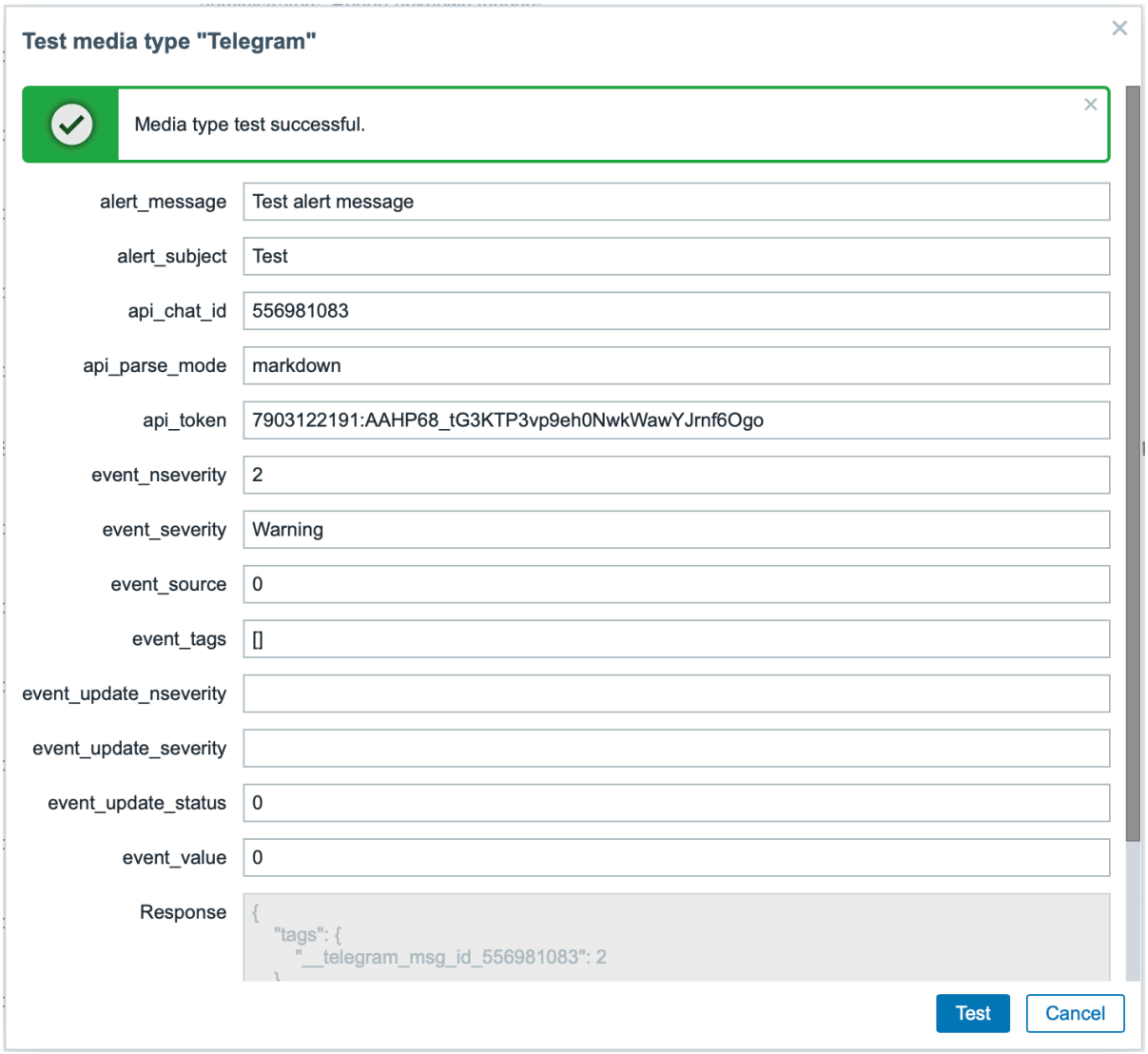
Warning:
Même si un webhook n'utilise pas de messages par défaut, les modèles de messages correspondant aux types d'opération utilisés par ce webhook doivent tout de même être définis.

Test

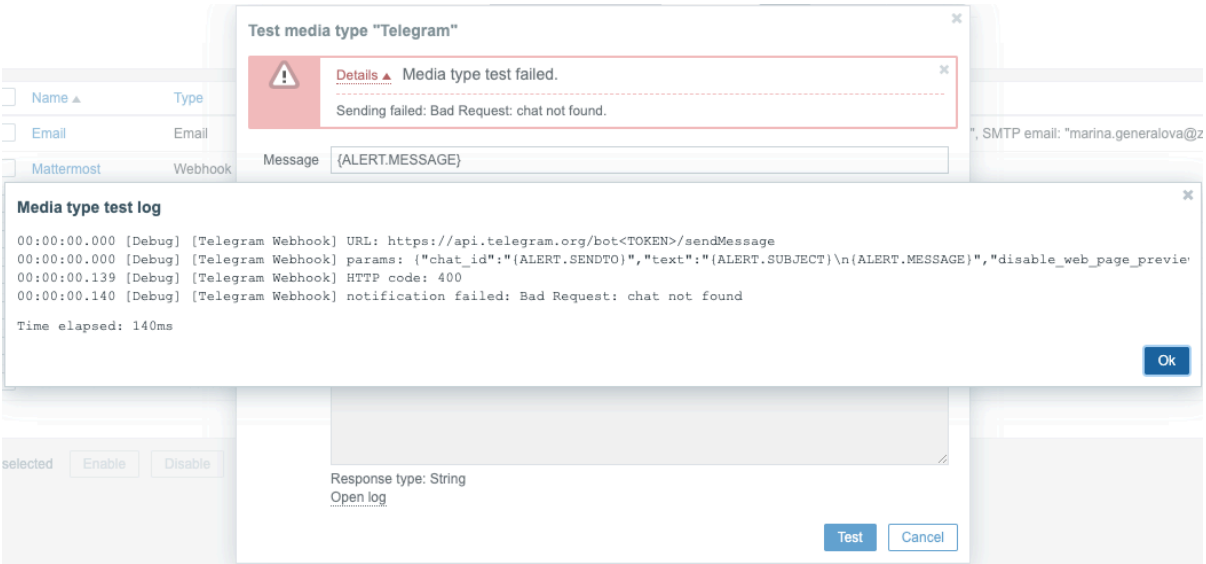
Pour tester un type de média webhook configuré :

1. Localisez le webhook concerné dans la liste des types de média.
2. Cliquez sur *Tester* dans la dernière colonne de la liste (une fenêtre de test s'ouvrira).
3. Modifiez les valeurs des paramètres du webhook si nécessaire.
   Remplacez les macros par des valeurs d'exemple ; sinon, les macros ne seront pas résolues et le test échouera.
4. Cliquez sur *Tester*.

Le remplacement ou la suppression de valeurs dans la fenêtre de test affecte uniquement la procédure de test ; les valeurs réelles des attributs du webhook restent inchangées.



Pour consulter les entrées du journal de test du type de média sans quitter la fenêtre de test, cliquez sur *Ouvrir le journal* (une nouvelle fenêtre contextuelle s'ouvrira).



Si le test du webhook réussit :

- Le message « *Media type test successful.* » s'affiche.
- La réponse du serveur apparaît dans le champ gris *Response*.
- Le type de réponse (JSON ou String) est indiqué sous le champ *Response*.

Si le test du webhook échoue :

- Le message « *Media type test failed.* » s'affiche, suivi de détails supplémentaires sur l'échec.

Média utilisateur

Une fois le type de média configuré, accédez à la section *Utilisateurs* > *Utilisateurs* et affectez le média webhook à un utilisateur existant, ou créez un nouvel utilisateur pour représenter le webhook. Les étapes de configuration des médias utilisateur pour un utilisateur existant, communes à tous les types de médias, sont décrites sur la page Types de médias.

Si un webhook utilise des étiquettes pour stocker l'ID du ticket\message, évitez d'affecter le même webhook comme média à différents utilisateurs, car cela peut entraîner des erreurs du webhook (cela concerne la majorité des webhooks qui utilisent l'option *Inclure l'entrée de menu de l'événement*). Dans ce cas, il est recommandé de créer un utilisateur dédié pour représenter le webhook :

1. Après avoir configuré le type de média webhook, accédez à la section *Utilisateurs* > *Utilisateurs* et créez un utilisateur Zabbix dédié pour représenter le webhook, par exemple avec le nom d'utilisateur *Slack* pour le webhook Slack. Tous les paramètres, à l'exception du média, peuvent conserver leurs valeurs par défaut, car cet utilisateur ne se connectera pas à Zabbix.
2. Dans le profil utilisateur, accédez à l'onglet *Média* et ajoutez un webhook avec les coordonnées requises. Si le webhook n'utilise pas de champ *Envoyer à*, saisissez une combinaison quelconque de caractères pris en charge afin de satisfaire aux exigences de validation.
3. Accordez à cet utilisateur au moins les permissions de lecture sur tous les hôtes pour lesquels il doit envoyer les alertes.

Lors de la configuration d'une action d'alerte, ajoutez cet utilisateur dans le champ *Envoyer aux utilisateurs* des détails de l'opération. Zabbix utilisera ainsi le webhook pour les notifications de cette action.

Configuration des actions d'alerte

Les actions déterminent quelles notifications doivent être envoyées via le webhook. Les étapes de configuration des actions impliquant des webhooks sont les mêmes que pour tous les autres types de média, à l'exception des cas suivants :

- Si un webhook utilise des balises de webhook pour stocker l'ID du ticket\message et gérer les opérations de mise à jour\résolution, évitez d'utiliser le même webhook dans plusieurs actions d'alerte pour un seul événement de problème. Si {EVENT.TAGS.<tag name>} existe et est mis à jour dans le webhook, sa valeur résultante sera indéfinie. Pour éviter cela, utilisez un nouveau nom de balise dans le webhook pour stocker les valeurs mises à jour. Cela s'applique aux webhooks Jira, Jira Service Desk, Mattermost, Opsgenie, OTRS, Redmine, ServiceNow, Slack, Zammad et Zendesk fournis par Zabbix, ainsi qu'à la plupart des webhooks utilisant l'option *Include event menu entry*. Notez toutefois qu'un seul webhook peut être utilisé dans plusieurs opérations ou étapes d'escalade de la même action, ainsi que dans différentes actions qui ne seront pas déclenchées par le même événement de problème en raison de conditions différentes.
- Lors de l'utilisation d'un webhook dans des actions pour des événements internes, veillez à cocher la case *Custom message* et à définir un message personnalisé dans la configuration de l'opération de l'action. Sinon, aucune notification ne sera envoyée.

Exemples de script webhook

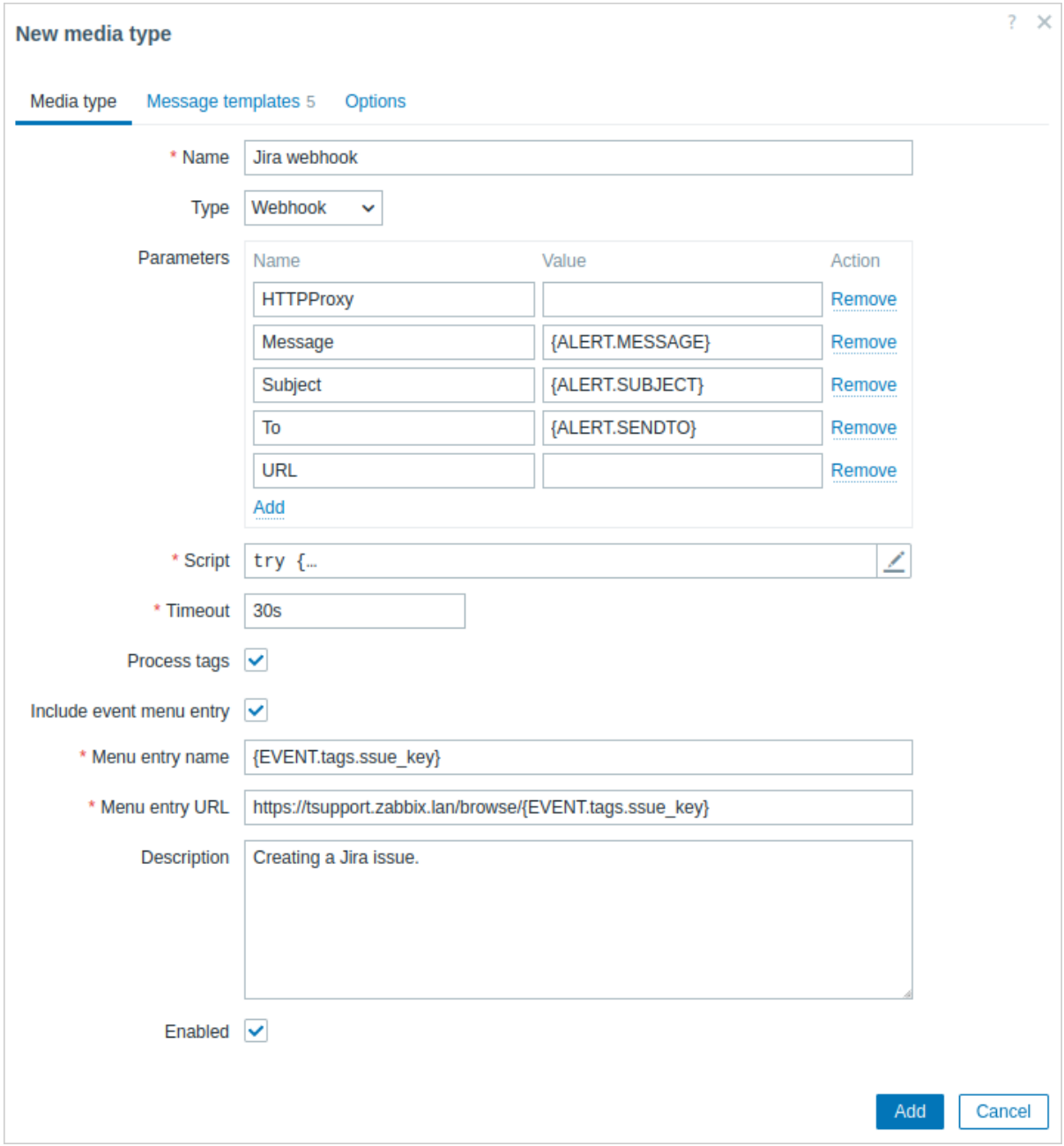
Aperçu

Bien que Zabbix propose un grand nombre d'intégrations webhook disponibles prêtes à l'emploi, vous pouvez souhaiter créer vos propres webhooks à la place. Cette section fournit des exemples de scripts webhook personnalisés (utilisés dans le paramètre *Script*). Voir la section webhook pour la description des autres paramètres de webhook.

> **Attention:**
> N'utilisez pas d'affectations non déclarées dans le JavaScript de prétraitement. Utilisez `var` pour déclarer les variables locales.

Les événements de récupération (qu'ils soient générés automatiquement ou à la suite d'une fermeture manuelle) incluent les tags d'événement résolus (y compris les tags hérités des modèles, des hôtes et des déclencheurs). Les scripts webhook sont exécutés après la création de l'alerte ; les tags renvoyés par un script webhook sont donc appliqués après la création initiale de l'alerte et peuvent ne pas être présents dans l'alerte qui a déclenché le webhook. Si une intégration nécessite que les tags produits par un webhook soient présents dans la notification initiale, récupérez les tags de l'événement depuis le serveur (par exemple, à l'aide de l'API Event) ou stockez les tags produits par le webhook dans un stockage persistant externe et faites la corrélation à cet endroit.

Webhook Jira (personnalisé)



Ce script créera un ticket JIRA et renverra certaines informations sur le ticket créé.

```
try {
    Zabbix.log(4, '[ Jira webhook ] Started with params: ' + value);

    var result = {
            'tags': {
                'endpoint': 'jira'
            }
        },
        params = JSON.parse(value),
        req = new HttpRequest(),
        fields = {},
        resp;

    if (params.HTTPProxy) {
        req.setProxy(params.HTTPProxy);
    }

    req.addHeader('Content-Type: application/json');
    req.addHeader('Authorization: Basic ' + params.authentication);
```

```
    fields.summary = params.summary;
    fields.description = params.description;
    fields.project = {key: params.project_key};
    fields.issuetype = {id: params.issue_id};

    resp = req.post('https://jira.example.com/rest/api/2/issue/',
        JSON.stringify({"fields": fields})
    );

    if (req.getStatus() != 201) {
        throw 'Response code: ' + req.getStatus();
    }

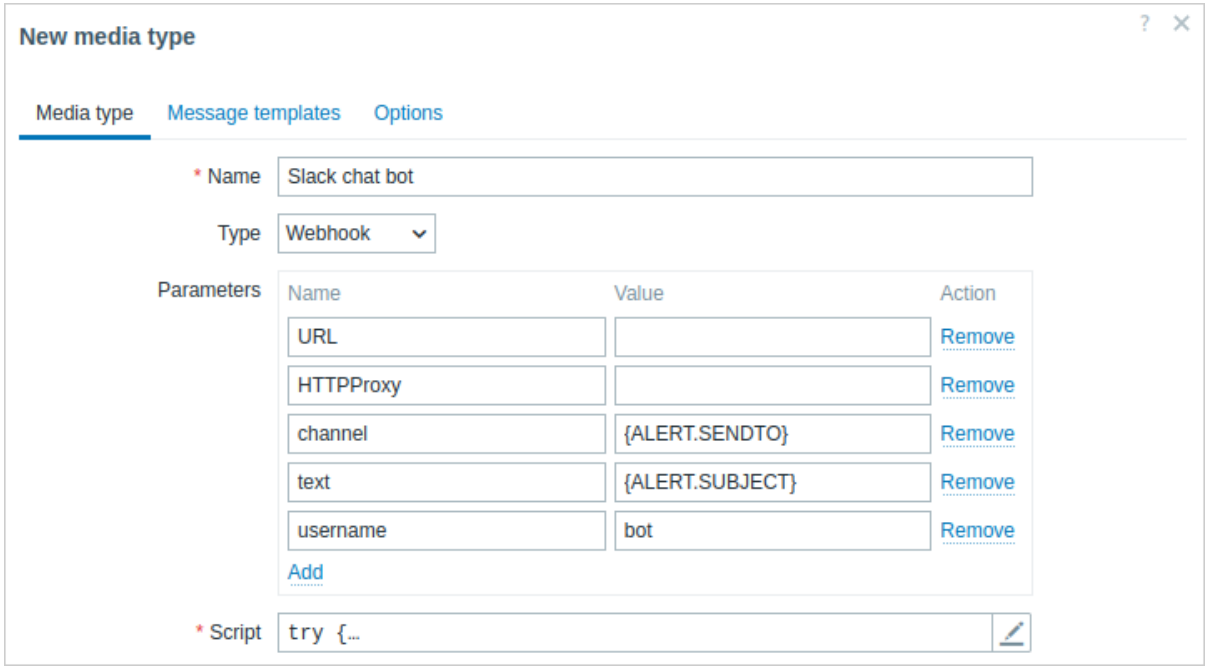
    resp = JSON.parse(resp);
    result.tags.issue_id = resp.id;
    result.tags.issue_key = resp.key;

    return JSON.stringify(result);
}
catch (error) {
    Zabbix.log(4, '[ Jira webhook ] Issue creation failed json : ' + JSON.stringify({"fields": fields}));
    Zabbix.log(3, '[ Jira webhook ] issue creation failed : ' + error);

    throw 'Failed with error: ' + error;
}
```

Webhook Slack (personnalisé)

Ce webhook transmettra les notifications de Zabbix vers un canal Slack.




```
try {
    var params = JSON.parse(value),
        req = new HttpRequest(),
        response;

    if (params.HTTPProxy) {
        req.setProxy(params.HTTPProxy);
    }

    req.addHeader('Content-Type: application/x-www-form-urlencoded');

    Zabbix.log(4, '[ Slack webhook ] Webhook request with value=' + value);
```

```
    response = req.post(params.URL, 'payload=' + encodeURIComponent(value));
    Zabbix.log(4, '[ Slack webhook ] Responded with code: ' + req.getStatus() + '. Response: ' + response)

    try {
        response = JSON.parse(response);
    }
    catch (error) {
        if (req.getStatus() < 200 || req.getStatus() >= 300) {
            throw 'Request failed with status code ' + req.getStatus();
        }
        else {
            throw 'Request success, but response parsing failed.';
        }
    }

    if (req.getStatus() !== 200 || !response.ok || response.ok === 'false') {
        throw response.error;
    }

    return 'OK';
}
catch (error) {
    Zabbix.log(3, '[ Slack webhook ] Sending failed. Error: ' + error);

    throw 'Failed with error: ' + error;
}
```

Actions

Vue d'ensemble

Si vous souhaitez que certaines opérations aient lieu à la suite d'événements (par exemple, l'envoi de notifications), vous devez configurer des actions.

Les actions peuvent être définies en réponse à des événements de tous les types pris en charge :

- Actions de déclencheur - pour les événements lorsque l'état du déclencheur passe entre `OK` et `Problem`
- Actions de service - pour les événements lorsque l'état du service passe entre `OK` et `Problem`
- Actions de découverte - pour les événements lorsque la découverte réseau a lieu
- Actions d'auto-enregistrement - pour les événements lorsque de nouveaux agents actifs s'auto-enregistrent (ou lorsque les métadonnées d'hôte changent pour ceux déjà enregistrés)
- Actions internes - pour les événements lorsque des éléments deviennent non pris en charge ou que des déclencheurs passent dans un état inconnu

Les principales différences des actions de service sont les suivantes :

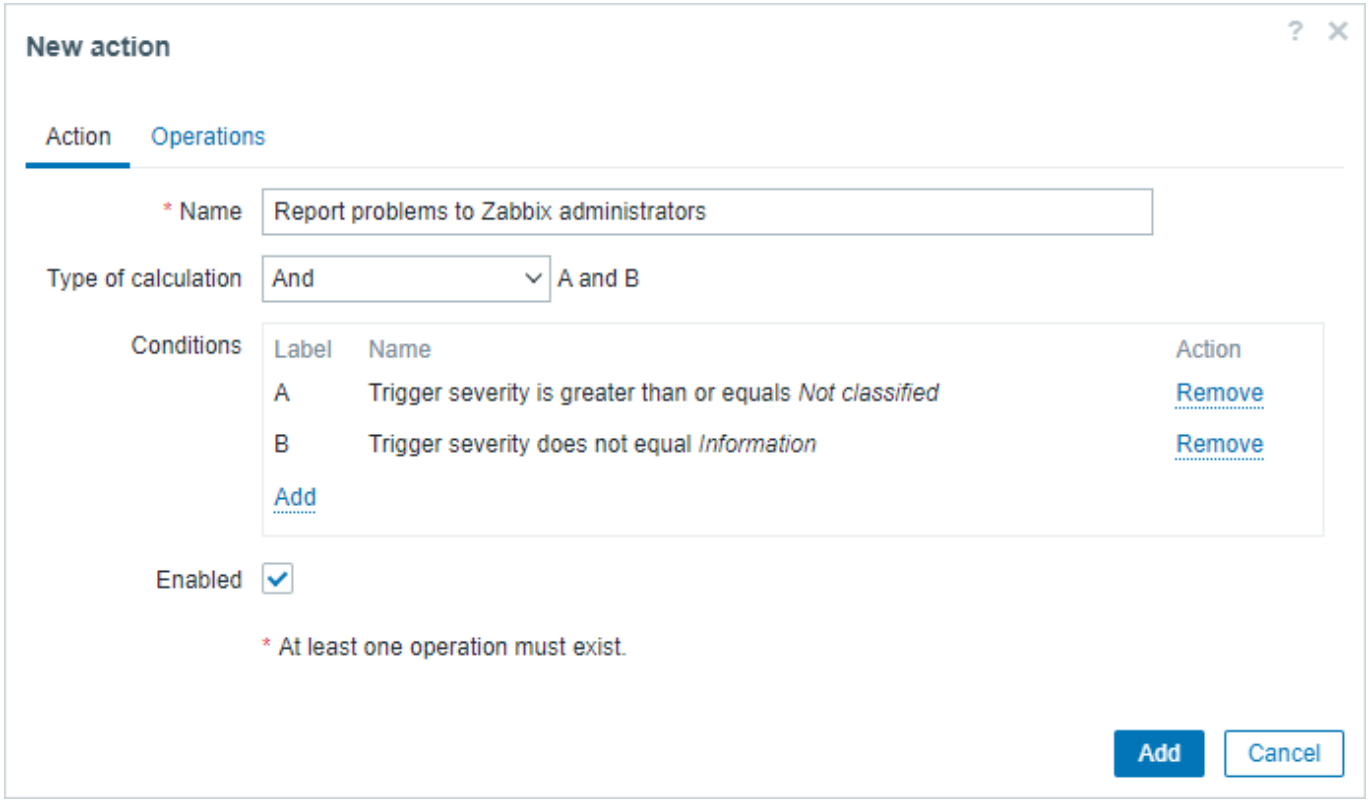
- L'accès des utilisateurs aux actions de service dépend des droits d'accès aux services accordés par le rôle de l'utilisateur
- Les actions de service prennent en charge un ensemble différent de conditions

Configuration d'une action

Pour configurer une action, procédez comme suit :

1. Accédez à *Alerts* > *Actions* et sélectionnez le type d'action requis dans le sous-menu (vous pouvez passer à un autre type plus tard, à l'aide de la liste déroulante du titre).
2. Cliquez sur *Create action*.
3. Nommez l'action.
4. Choisissez les conditions selon lesquelles les opérations sont exécutées.
5. Choisissez les opérations à exécuter.

Attributs généraux de l'action :



Tous les champs de saisie obligatoires sont marqués d'un astérisque rouge.

Parameter	Description
Name	Nom unique de l'action.
Type of calculation	Sélectionnez l'option d'évaluation des conditions de l'action (avec plus d'une condition) : **And** - toutes les conditions doivent être remplies. **Or** - il suffit qu'une condition soit remplie. **And/Or** - combinaison des deux : AND avec des types de conditions différents et OR avec le même type de condition. **Custom expression** - une formule de calcul définie par l'utilisateur pour évaluer les conditions de l'action.
Conditions	Liste des conditions de l'action. Cliquez sur *Add* pour ajouter une nouvelle condition. Si aucune condition n'est configurée, l'action s'exécutera pour chaque événement correspondant au type d'action configuré.
Enabled	Cochez la case pour activer l'action. Sinon, elle sera désactivée.

Conditions

Vue d'ensemble

Il est possible de définir qu'une action est exécutée uniquement si l'événement correspond à un ensemble défini de conditions. Les conditions sont définies lors de la configuration de l'action.

La correspondance des conditions est sensible à la casse.

Actions de déclencheur

Les conditions suivantes peuvent être utilisées dans les actions basées sur les déclencheurs :

Type de condition	Opérateurs pris en charge	Description
Groupe d'hôtes	égal à différent de	Spécifiez des groupes d'hôtes ou des groupes d'hôtes à exclure. **égal à** - l'événement appartient à ce groupe d'hôtes. **différent de** - l'événement n'appartient pas à ce groupe d'hôtes. La spécification d'un groupe d'hôtes parent sélectionne implicitement tous les groupes d'hôtes imbriqués. Pour spécifier uniquement le groupe parent, tous les groupes imbriqués doivent également être définis avec l'opérateur **différent de**.
Modèle	égal à différent de	Spécifiez des modèles ou des modèles à exclure. **égal à** - l'événement appartient à un déclencheur hérité de ce modèle. **différent de** - l'événement n'appartient pas à un déclencheur hérité de ce modèle.
Hôte	égal à différent de	Spécifiez des hôtes ou des hôtes à exclure. **égal à** - l'événement appartient à cet hôte. **différent de** - l'événement n'appartient pas à cet hôte.
Nom du tag	égal à différent de contient ne contient pas	Spécifiez un tag d'événement ou un tag d'événement à exclure. **égal à** - l'événement possède ce tag. **différent de** - l'événement ne possède pas ce tag. **contient** - l'événement possède un tag contenant cette chaîne. **ne contient pas** - l'événement ne possède pas de tag contenant cette chaîne.
Valeur du tag	égal à différent de contient ne contient pas	Spécifiez une combinaison tag/valeur d'événement ou une combinaison tag/valeur à exclure. **égal à** - l'événement possède ce tag et cette valeur. **différent de** - l'événement ne possède pas ce tag et cette valeur. **contient** - l'événement possède un tag et une valeur contenant ces chaînes. **ne contient pas** - l'événement ne possède pas de tag et de valeur contenant ces chaînes.
Déclencheur	égal à différent de	Spécifiez des déclencheurs ou des déclencheurs à exclure. **égal à** - l'événement est généré par ce déclencheur. **différent de** - l'événement est généré par n'importe quel autre déclencheur, sauf celui-ci.
Nom de l'événement	contient ne contient pas	Spécifiez une chaîne dans le nom de l'événement généré par le déclencheur ou une chaîne à exclure. Par défaut, le nom de l'événement correspond au nom du déclencheur, sauf si un nom d'événement personnalisé est spécifié dans la configuration du déclencheur. **contient** - le nom de l'événement contient cette chaîne. **ne contient pas** - cette chaîne est exclue du nom de l'événement. Remarque : la valeur saisie sera comparée au nom de l'événement avec toutes les macros développées.
Gravité du déclencheur	égal à différent de est supérieur ou égal à est inférieur ou égal à	Spécifiez la gravité du déclencheur. **égal à** - égal à la gravité du déclencheur. **différent de** - différent de la gravité du déclencheur. **est supérieur ou égal à** - supérieur ou égal à la gravité du déclencheur. **est inférieur ou égal à** - inférieur ou égal à la gravité du déclencheur.
Période de temps	dans pas dans	Spécifiez une période de temps ou une période de temps à exclure. **dans** - l'heure de l'événement se situe dans la période de temps. **pas dans** - l'heure de l'événement ne se situe pas dans la période de temps. Consultez la page spécification de la période de temps pour une description du format. Les macros utilisateur sont prises en charge.
Le problème est supprimé	non oui	Indiquez si le problème est supprimé (non affiché) en raison de la maintenance de l'hôte. **non** - le problème n'est pas supprimé. **oui** - le problème est supprimé.

Actions de service

Les conditions suivantes peuvent être utilisées dans les actions de service :

Type de condition	Opérateurs pris en charge	Description
Service	equals does not equal	Spécifiez un service ou un service à exclure. **equals** - l'événement appartient à ce service. **does not equal** - l'événement n'appartient pas à ce service. La spécification d'un service parent sélectionne implicitement tous les services enfants. Pour spécifier uniquement le service parent, tous les services imbriqués doivent également être définis avec l'opérateur **does not equal**.
Nom du service	contains does not contain	Spécifiez une chaîne dans le nom du service ou une chaîne à exclure. **contains** - l'événement est généré par un service contenant cette chaîne dans son nom. **does not contain** - cette chaîne ne peut pas être trouvée dans le nom du service.
Nom de la balise de service	equals does not equal contains does not contain	Spécifiez une balise d'événement ou une balise d'événement à exclure. Les balises d'événement de service peuvent être définies dans la section de configuration du service *Tags*. **equals** - l'événement possède cette balise. **does not equal** - l'événement ne possède pas cette balise. **contains** - l'événement possède une balise contenant cette chaîne. **does not contain** - l'événement ne possède pas une balise contenant cette chaîne.
Valeur de la balise de service	equals does not equal contains does not contain	Spécifiez une combinaison de balise d'événement et de valeur, ou une combinaison de balise et de valeur à exclure. Les balises d'événement de service peuvent être définies dans la section de configuration du service *Tags*. **equals** - l'événement possède cette balise et cette valeur. **does not equal** - l'événement ne possède pas cette balise et cette valeur. **contains** - l'événement possède une balise et une valeur contenant ces chaînes. **does not contain** - l'événement ne possède pas une balise et une valeur contenant ces chaînes.

Attention:
Assurez-vous de définir des modèles de message pour les actions de service dans le menu *Alerts* > *Media types*. Sinon, les notifications ne seront pas envoyées.

Actions de découverte

Les conditions suivantes peuvent être utilisées dans les événements basés sur la découverte :

Type de condition	Opérateurs pris en charge	Description
Adresse IP de l'hôte	est égal à n'est pas égal à	Indiquez une plage d'adresses IP ou une plage à exclure pour un hôte découvert. **est égal à** - l'adresse IP de l'hôte se trouve dans la plage. **n'est pas égal à** - l'adresse IP de l'hôte ne se trouve pas dans la plage. Les formats suivants sont possibles : Adresse IP unique : 192.168.1.33 Plage d'adresses IP : 192.168.1-10.1-254 Masque IP : 192.168.4.0/24 Liste : 192.168.1.1-254, 192.168.2.1-100, 192.168.2.200, 192.168.4.0/24 Les espaces dans le format de liste sont pris en charge.

Type de condition	Opérateurs pris en charge	Description
Type de service	est égal à n'est pas égal à	Indiquez un type de service découvert ou un type de service à exclure. **est égal à** - correspond au service découvert. **n'est pas égal à** - ne correspond pas au service découvert. Types de service disponibles : SSH, LDAP, SMTP, FTP, HTTP, HTTPS, POP, NNTP, IMAP, TCP, agent Zabbix, agent SNMPv1, agent SNMPv2, agent SNMPv3, ping ICMP, telnet.
Port du service	est égal à n'est pas égal à	Indiquez une plage de ports TCP d'un service découvert ou une plage à exclure. **est égal à** - le port du service se trouve dans la plage. **n'est pas égal à** - le port du service ne se trouve pas dans la plage.
Règle de découverte	est égal à n'est pas égal à	Indiquez une règle de découverte ou une règle de découverte à exclure. **est égal à** - utilise cette règle de découverte. **n'est pas égal à** - utilise toute autre règle de découverte, à l'exception de celle-ci.
Vérification de découverte	est égal à n'est pas égal à	Indiquez une vérification de découverte ou une vérification de découverte à exclure. **est égal à** - utilise cette vérification de découverte. **n'est pas égal à** - utilise toute autre vérification de découverte, à l'exception de celle-ci.
Objet découvert	est égal à	Indiquez l'objet découvert. **est égal à** - égal à l'objet découvert (un périphérique ou un service).
État de la découverte	est égal à	**Up** - correspond aux événements « Host Up » et « Service Up ». **Down** - correspond aux événements « Host Down » et « Service Down ». **Discovered** - correspond aux événements « Host Discovered » et « Service Discovered ». **Lost** - correspond aux événements « Host Lost » et « Service Lost ».
Temps de fonctionnement/Temps d'arrêt	est supérieur ou égal à est inférieur ou égal à	Temps de fonctionnement pour les événements « Host Up » et « Service Up ». Temps d'arrêt pour les événements « Host Down » et « Service Down ». **est supérieur ou égal à** - est supérieur ou égal à. Le paramètre est indiqué en secondes. **est inférieur ou égal à** - est inférieur ou égal à. Le paramètre est indiqué en secondes.
Valeur reçue	est égal à n'est pas égal à est supérieur ou égal à est inférieur ou égal à contient ne contient pas	Indiquez la valeur reçue d'une vérification d'agent (Zabbix, SNMP) dans une règle de découverte. Comparaison de chaînes. Si plusieurs vérifications d'agent Zabbix ou SNMP sont configurées pour une règle, les valeurs reçues pour chacune d'elles sont vérifiées (chaque vérification génère un nouvel événement qui est comparé à toutes les conditions). **est égal à** - égal à la valeur. **n'est pas égal à** - différent de la valeur. **est supérieur ou égal à** - supérieur ou égal à la valeur. **est inférieur ou égal à** - inférieur ou égal à la valeur. **contient** - contient la sous-chaîne. Le paramètre est indiqué sous forme de chaîne. **ne contient pas** - ne contient pas la sous-chaîne. Le paramètre est indiqué sous forme de chaîne.
Proxy	est égal à n'est pas égal à	Indiquez un proxy ou un proxy à exclure. **est égal à** - utilise ce proxy. **n'est pas égal à** - utilise tout autre proxy, à l'exception de celui-ci.
Groupe de proxies	est égal à n'est pas égal à	Indiquez un groupe de proxies ou un groupe de proxies à exclure. **est égal à** - utilise ce groupe de proxies. **n'est pas égal à** - utilise tout autre groupe de proxies, à l'exception de celui-ci.

Note:
Les vérifications de service d'une règle de découverte, qui génèrent des événements de découverte, ne sont pas effectuées simultanément. Par conséquent, si des valeurs **multiples** sont configurées pour les conditions `Type de service`, `Port du service` ou `Valeur reçue` dans l'action, elles seront comparées à un événement de découverte à la fois, mais **pas** à plusieurs événements simultanément. Par conséquent, les actions comportant plusieurs valeurs pour les mêmes types de vérification peuvent ne pas être exécutées correctement.

Actions d'enregistrement automatique

Les conditions suivantes peuvent être utilisées dans les actions basées sur l'enregistrement automatique d'un agent actif :

Type de condition	Opérateurs pris en charge	Description
Métadonnées de l'hôte	contient ne contient pas correspond à ne correspond pas à	Indiquez les métadonnées de l'hôte ou les métadonnées de l'hôte à exclure. **contient** - les métadonnées de l'hôte contiennent la chaîne. **ne contient pas** - les métadonnées de l'hôte ne contiennent pas la chaîne. Les métadonnées de l'hôte peuvent être indiquées dans un fichier de configuration de l'agent. **correspond à** - les métadonnées de l'hôte correspondent à l'expression régulière. **ne correspond pas à** - les métadonnées de l'hôte ne correspondent pas à l'expression régulière.
Nom de l'hôte	contient ne contient pas correspond à ne correspond pas à	Indiquez un nom d'hôte ou un nom d'hôte à exclure. **contient** - le nom de l'hôte contient la chaîne. **ne contient pas** - le nom de l'hôte ne contient pas la chaîne. **correspond à** - le nom de l'hôte correspond à l'expression régulière. **ne correspond pas à** - le nom de l'hôte ne correspond pas à l'expression régulière.
Proxy	est égal à n'est pas égal à	Indiquez un proxy ou un proxy à exclure. **est égal à** - utilise ce proxy. **n'est pas égal à** - utilise tout autre proxy (ou serveur), à l'exception de ce proxy.
Groupe de proxies	est égal à n'est pas égal à	Indiquez un groupe de proxies ou un groupe de proxies à exclure. **est égal à** - utilise ce groupe de proxies. **n'est pas égal à** - utilise tout autre groupe de proxies, à l'exception de celui-ci.

Actions d'événements internes

Les conditions suivantes peuvent être définies pour les actions basées sur des événements:

Type de condition	Opérateurs pris en charge	Description
Type d'événement	égal	**Élément dans l'état "non supporté"** - correspond aux événements où un élément passe d'un état 'normal' à 'non supporté' **Règle de découverte bas niveau dans l'état "non supporté"** : correspond aux événements où une règle de découverte de bas niveau passe d'un état 'normal' à 'non supporté' **Déclencheur dans l'état "inconnu"** : correspond aux événements où un déclencheur passe d'un état 'normal' à 'inconnu'
Groupe d'hôtes	égal n'est pas égal	Spécifiez les groupes d'hôtes ou les groupes d'hôtes à exclure. **égal** - l'événement appartient à ce groupe d'hôtes. **n'est pas égal** - l'événement n'appartient pas à ce groupe d'hôtes.
Nom de tag	égal n'est pas égal contient ne contient pas	Spécifiez le tag d'événement ou le tag d'événement à exclure. **égal** - l'événement a ce tag * *n'est pas égal **- l'événement n'a pas ce tag **contient **- l'événement a un tag contenant cette chaîne **ne contient pas** - l'événement n'a pas de tag contenant cette chaîne

Type de condition	Opérateurs pris en charge	Description
Valeur de tag	égal n'est pas égal contient ne contient pas	Spécifiez la combinaison de tag et de valeur de tag d'événement ou la combinaison de tag et de valeur de tag à exclure. **égal** - l'événement a ce tag et cette valeur **n'est pas égal** - l'événement n'a pas ce tag et cette valeur **contient** - l'événement a un tag et une valeur contenant ces chaînes **ne contient pas** - l'événement n'a pas de tag et de valeur contenant ces chaînes
Modèle	égal n'est pas égal	Spécifiez les modèles ou les modèles à exclure. **égal** - l'événement appartient à un élément/déclencheur/règle de découverte de bas niveau héritée de ce modèle. **n'est pas égal** - l'événement n'appartient pas à un élément/déclencheur/règle de découverte de bas niveau héritée de ce modèle.
Hôte	égal n'est pas égal	Spécifiez les hôtes ou les hôtes à exclure. **égal** - l'événement appartient à cet hôte. **n'est pas égal** - l'événement n'appartient pas à cet hôte.

Type de calcul

Les options suivantes de calcul des conditions sont disponibles :

- **And** - toutes les conditions doivent être remplies

Notez que l'utilisation du calcul "And" est interdite entre plusieurs déclencheurs lorsqu'ils sont sélectionnés comme condition `Trigger=`. Les actions ne peuvent être exécutées que sur la base de l'événement d'un seul déclencheur.

- **Or** - il suffit qu'une seule condition soit remplie
- **And/Or** - combinaison des deux : AND avec différents types de conditions et OR avec le même type de condition, par exemple :

Groupe d'hôtes égal à Oracle servers
 Groupe d'hôtes égal à MySQL servers
 Nom d'événement contient 'Database is down'
 Nom d'événement contient 'Database is unavailable'

est évalué comme

(Groupe d'hôtes égal à Oracle servers **or** Groupe d'hôtes égal à MySQL servers**) and (**Nom d'événement contient 'Database is down' **or** Nom d'événement contient 'Database is unavailable'**)**

- **Expression personnalisée** - une formule de calcul définie par l'utilisateur pour évaluer les conditions d'action. Elle doit inclure toutes les conditions (représentées par des lettres majuscules A, B, C, ...) et peut inclure des espaces, des tabulations, des parenthèses ( ), **and** (sensible à la casse), **or** (sensible à la casse), **not** (sensible à la casse).

Alors que l'exemple précédent avec `And/Or` serait représenté par (A or B) and (C or D), dans une expression personnalisée, vous pouvez aussi avoir plusieurs autres modes de calcul :

(A and B) and (C or D)
 (A and B) or (C and D)
 ((A or B) and C) or D
 (not (A or B) and C) or not D
 etc.

Suppression d'objets utilisés dans des actions

Lorsqu'un utilisateur tente de supprimer des objets utilisés dans une condition/opération d'action, il reçoit un message d'erreur correspondant.

Les actions ne sont pas désactivées si les conditions ou opérations d'action :

- comportent des objets supprimés ;
- comportent des objets qui ne sont plus découverts par la découverte de bas niveau et qui ont été supprimés (après la période *Delete lost resources*).

Les conditions/opérations d'action ne sont pas supprimées si des objets sont manquants. À la place, les objets supprimés sont affichés comme *Deleted host group*, *Deleted host*, *Deleted trigger*, etc. Une icône d'avertissement accompagnée du message « This action has conditions or operations referencing deleted object(s). » est affichée dans la colonne « Info » de la liste des actions.

Opérations

Aperçu

Vous pouvez définir les opérations suivantes pour tous les événements :

- Envoyer un message
- Exécuter une commande à distance

Attention:
Le serveur Zabbix ne crée pas d'alertes si l'accès à l'hôte est explicitement « refusé » pour l'utilisateur défini comme destinataire de l'opération d'action, ou si aucun droit n'est défini pour cet hôte pour cet utilisateur.

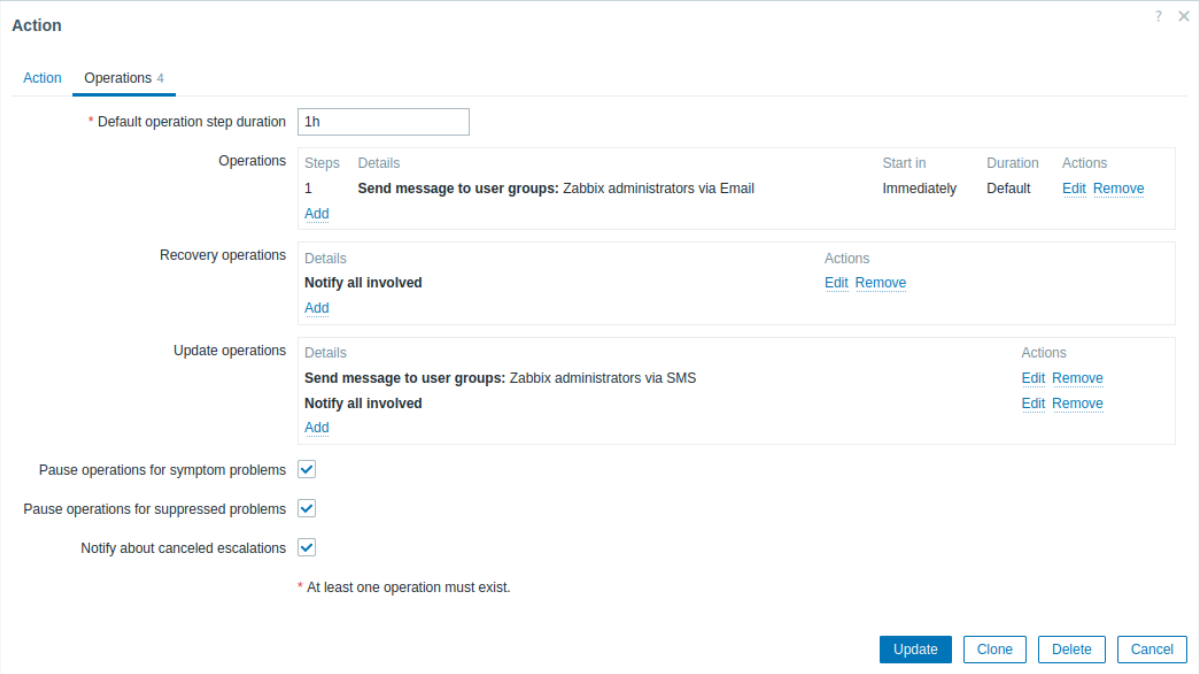
Pour les événements de découverte et d'auto-enregistrement, des opérations supplémentaires sont disponibles :

- Ajouter un hôte
- Supprimer l'hôte
- Activer l'hôte
- Désactiver l'hôte
- Ajouter à un groupe d'hôtes
- Supprimer d'un groupe d'hôtes
- Ajouter des tags d'hôte
- Supprimer des tags d'hôte
- Lier un modèle
- Dissocier un modèle
- Définir le mode d'inventaire de l'hôte

Configuration d'une opération

Pour configurer une opération :

1. Accédez à l'onglet *Opérations* dans la configuration de l'action.
2. Configurez les détails de l'opération.



Attributs généraux des opérations :

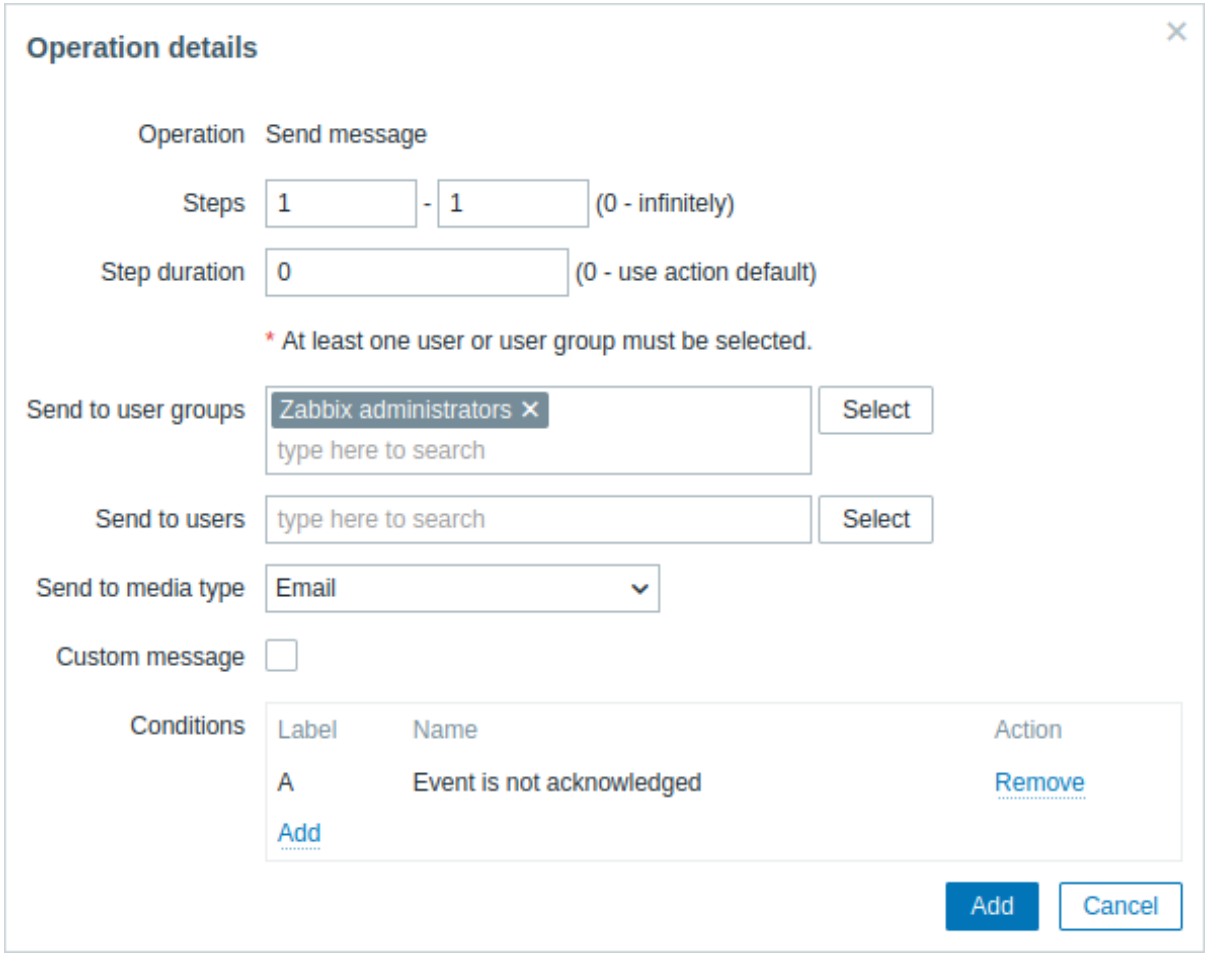
Paramètre	Description
Durée d'étape d'opération par défaut	Durée d'une étape d'opération par défaut (de 60 secondes à 1 semaine). Par exemple, une durée d'étape d'une heure signifie que, si une opération est exécutée, une heure s'écoulera avant l'étape suivante. Les suffixes de temps sont pris en charge, par exemple : 60s, 1m, 2h, 1d. Les macros utilisateur sont prises en charge.

Paramètre	Description
Opérations	Les opérations d'action (le cas échéant) sont affichées avec les détails suivants : **Étapes** - étape(s) d'escalade auxquelles l'opération est affectée. **Détails** - type d'opération et son destinataire/cible. La liste des opérations affiche également le type de média (e-mail, SMS ou script) utilisé, ainsi que le nom et le prénom (entre parenthèses après le nom d'utilisateur) du destinataire d'une notification. **Commence dans** - délai après un événement au terme duquel l'opération est exécutée. **Durée (sec)** - la durée de l'étape est affichée. *Par défaut* est affiché si l'étape utilise la durée par défaut, et une durée est affichée si une durée personnalisée est utilisée. **Actions** - les liens permettant de modifier et de supprimer une opération sont affichés.
Opérations de rétablissement	Les opérations d'action (le cas échéant) sont affichées avec les détails suivants : **Détails** - type d'opération et son destinataire/cible. La liste des opérations affiche également le type de média (e-mail, SMS ou script) utilisé, ainsi que le nom et le prénom (entre parenthèses après le nom d'utilisateur) du destinataire d'une notification. **Actions** - les liens permettant de modifier et de supprimer une opération sont affichés.
Opérations de mise à jour	Les opérations d'action (le cas échéant) sont affichées avec les détails suivants : **Détails** - type d'opération et son destinataire/cible. La liste des opérations affiche également le type de média (e-mail, SMS ou script) utilisé, ainsi que le nom et le prénom (entre parenthèses après le nom d'utilisateur) du destinataire d'une notification. **Actions** - les liens permettant de modifier et de supprimer une opération sont affichés.
Mettre en pause les opérations pour les problèmes symptomatiques	Cochez cette case pour mettre en pause les opérations (après la première opération) pour les problèmes symptomatiques. Notez que ce paramètre affecte uniquement les escalades de problèmes ; les opérations de rétablissement et de mise à jour ne seront pas affectées. Cette option est disponible uniquement pour les *actions de déclencheur*.
Mettre en pause les opérations pour les problèmes supprimés	Cochez cette case pour retarder le démarrage des opérations pendant toute la durée d'une période de maintenance. Lorsque les opérations démarrent, après la maintenance, toutes les opérations sont exécutées, y compris celles correspondant aux événements survenus pendant la maintenance. Notez que ce paramètre affecte uniquement les escalades de problèmes ; les opérations de rétablissement et de mise à jour ne seront pas affectées. Si vous décochez cette case, les opérations seront exécutées sans délai, même pendant une période de maintenance. Cette option n'est pas disponible pour les *actions de service*.
Notifier au sujet des escalades annulées	Décochez cette case pour désactiver les notifications concernant les escalades annulées (lorsqu'un hôte, un élément, un déclencheur ou une action est désactivé).

Tous les champs de saisie obligatoires sont marqués d'un astérisque rouge.

Pour configurer les détails d'une nouvelle opération, cliquez sur Add dans le bloc *Opérations*. Pour modifier une opération existante, cliquez sur Edit à côté de l'opération. Une fenêtre contextuelle s'ouvrira, dans laquelle vous pourrez modifier les détails de l'étape de l'opération.

Détails de l'opération



Paramètre	Description
Opération	Sélectionnez l'opération : **Envoyer un message** - envoyer un message à l'utilisateur. **<nom de la commande distante>** - exécuter une commande distante. Les commandes peuvent être exécutées si elles ont été préalablement définies dans les scripts globaux, avec *Opération d'action* sélectionné comme portée. D'autres opérations sont disponibles pour les événements de découverte et d'autoregistration (voir ci-dessus).
Étapes	Sélectionnez la ou les étapes auxquelles affecter l'opération dans un calendrier d'escalade : **De** - commencer l'exécution à partir de cette étape. **À** - exécuter jusqu'à cette étape (0 = infini, l'exécution ne sera pas limitée).
Durée de l'étape	Durée personnalisée pour ces étapes (0 = utiliser la durée d'étape par défaut). Les suffixes temporels sont pris en charge, par exemple 60s, 1m, 2h, 1d. Les macros utilisateur sont prises en charge. Plusieurs opérations peuvent être affectées à la même étape. Si ces opérations ont des durées d'étape différentes, la plus courte est prise en compte et appliquée à l'étape.
Type d'opération : envoyer un message	
Envoyer aux groupes d'utilisateurs	Sélectionnez les groupes d'utilisateurs auxquels envoyer le message. Le groupe d'utilisateurs doit disposer d'au moins des permissions de « lecture » sur le hôte pour être notifié.

Paramètre	Description
Envoyer aux utilisateurs	Sélectionnez les utilisateurs auxquels envoyer le message. L'utilisateur doit disposer d'au moins des permissions de « lecture » sur le hôte pour être notifié.
Envoyer au type de média	Envoyer le message à tous les types de média disponibles (configurés et activés), ou uniquement à un type spécifique.
Message personnalisé	Si cette option est sélectionnée, le message personnalisé peut être configuré. Pour les notifications concernant les événements internes via des webhooks, un message personnalisé est obligatoire.
Sujet	Sujet du message personnalisé. Le sujet peut contenir des macros. Il est limité à 255 caractères.
Message	Message personnalisé. Le message peut contenir des macros. Il est limité à un certain nombre de caractères selon le type de base de données (voir Envoyer un message pour plus d'informations).
Type d'opération : commande distante	
Liste des cibles	Sélectionnez les cibles sur lesquelles exécuter la commande : **Hôte actuel** - la commande est exécutée sur le hôte du déclencheur à l'origine de l'événement de problème. Cette option ne fonctionnera pas si le déclencheur concerne plusieurs hôtes. **Hôte** - sélectionnez le ou les hôtes sur lesquels exécuter la commande. **Groupe d'hôtes** - sélectionnez le ou les groupes d'hôtes sur lesquels exécuter la commande. La spécification d'un groupe d'hôtes parent sélectionne implicitement tous les groupes d'hôtes imbriqués. Ainsi, la commande distante sera également exécutée sur les hôtes des groupes imbriqués. Une commande sur un hôte n'est exécutée qu'une seule fois, même si l'hôte correspond plusieurs fois (par exemple, depuis plusieurs groupes d'hôtes, individuellement et depuis un groupe d'hôtes). La liste des cibles n'a pas de sens si un script personnalisé est exécuté sur le serveur Zabbix. Dans ce cas, sélectionner davantage de cibles entraîne uniquement l'exécution du script plusieurs fois sur le serveur. Notez que pour les scripts globaux, la sélection des cibles dépend également du paramètre *Groupe d'hôtes* dans la configuration du script global. L'option *Liste des cibles* n'est pas disponible pour les *actions de service*, car les commandes distantes sont toujours exécutées sur le serveur Zabbix dans ce cas.
Conditions	Condition d'exécution de l'opération : **L'événement n'est pas reconnu** - uniquement lorsque l'événement n'est pas reconnu. **L'événement est reconnu** - uniquement lorsque l'événement est reconnu. L'option *Conditions* est disponible uniquement pour les *actions de déclencheur*.

Lorsque vous avez terminé, cliquez sur *Ajouter* pour ajouter l'opération à la liste des *Opérations*.

Envoi du message

Aperçu

L'envoi d'un message est l'un des meilleurs moyens d'informer les personnes d'un problème. C'est pourquoi c'est l'une des principales actions proposées par Zabbix.

Configuration

Pour pouvoir envoyer et recevoir des notifications de Zabbix, vous devez :

- Définir le média auquel envoyer un message.

Si l'opération a lieu en dehors de la période Actif définie pour le média sélectionné dans la configuration de l'utilisateur, le message ne sera pas envoyé.

La gravité de déclencheur par défaut (« Non classé ») **doit être** cochée dans la configuration du média utilisateur si vous souhaitez recevoir des notifications pour des événements autres que ceux liés aux déclencheurs, tels que la découverte, l'auto-enregistrement d'un agent actif ou les événements internes.

- Configurer une opération d'action qui envoie un message à l'un des médias définis.

> **Attention:**
> Zabbix envoie des notifications uniquement aux utilisateurs disposant au minimum d'autorisations de « lecture » sur le hôte à l'origine de l'événement. Au moins un hôte d'une expression de déclencheur doit être accessible.

Vous pouvez configurer des scénarios personnalisés pour l'envoi de messages à l'aide des escalades.

Pour recevoir et lire correctement les e-mails de Zabbix, les serveurs/clients de messagerie doivent prendre en charge le format standard « e-mail SMTP/MIME », car Zabbix envoie des données UTF-8 (si le sujet contient uniquement des caractères ASCII, il n'est pas encodé en UTF-8). Le sujet et le corps du message sont encodés en base64 afin de respecter la norme du format « e-mail SMTP/MIME ».

La limite de taille des messages après l'expansion de toutes les macros est identique à celle des commandes à distance.

Suivi des messages

Vous pouvez consulter l'état des messages envoyés dans *Monitoring* > *Problems*.

Dans la colonne *Actions*, vous pouvez voir des informations récapitulatives sur les actions effectuées. Les nombres en vert représentent les messages envoyés, les nombres en rouge - les messages ayant échoué. *In progress* indique qu'une action est en cours d'initialisation. *Failed* indique qu'aucune action n'a été exécutée avec succès.

Si vous cliquez sur l'heure de l'événement pour afficher les détails de l'événement, vous pourrez voir les détails des messages envoyés (ou non envoyés) en raison de l'événement dans le bloc *Actions*.

Dans *Reports* > *Action log*, vous verrez les détails de toutes les actions effectuées pour les événements pour lesquels une action est configurée.

Commandes à distance

Vue d'ensemble

Avec les commandes distantes, vous pouvez définir qu'une commande prédéfinie donnée soit exécutée automatiquement sur l'hôte surveillé lorsqu'une condition est remplie.

Les commandes distantes constituent donc un mécanisme puissant pour une surveillance proactive et intelligente.

Voici quelques utilisations évidentes de cette fonctionnalité :

- Redémarrer automatiquement une application (serveur web, middleware, CRM) si elle ne répond pas
- Utiliser la commande IPMI « reboot » pour redémarrer un serveur distant s'il ne répond pas aux requêtes
- Libérer automatiquement de l'espace disque (en supprimant les fichiers anciens et en nettoyant /tmp) lorsque l'espace disque vient à manquer
- Migrer une VM d'une machine physique vers une autre en fonction de la charge CPU
- Ajouter de nouveaux nœuds à un environnement cloud lorsque les ressources CPU (disque, mémoire ou autres) sont insuffisantes

La configuration d'une action pour les commandes distantes est similaire à celle d'une action d'envoi de message ; la seule différence est que Zabbix exécute une commande au lieu d'envoyer un message.

Les commandes distantes peuvent être exécutées par le serveur Zabbix, le proxy Zabbix ou l'agent Zabbix. Notez les points suivants :

- Les commandes distantes sur l'agent Zabbix peuvent être exécutées directement par le serveur Zabbix ou via un proxy Zabbix.
- Les commandes distantes exécutées par le serveur Zabbix sont lancées comme décrit dans Exécution de commandes, y compris la vérification du code de sortie.
- Les commandes distantes sont exécutées même si l'hôte cible est en période de maintenance.

Les commandes distantes sont désactivées par défaut sur l'agent Zabbix et le proxy Zabbix. Elles peuvent être activées comme suit :

- Pour les commandes distantes exécutées par l'agent Zabbix, ajoutez un paramètre `AllowKey=system.run[<command>,*]` ou `AllowKeyRegexp=^system\.run\[<command>,.*\]$` pour chaque commande autorisée dans la configuration de l'agent, où `*` correspond à l'un ou l'autre des modes d'exécution (`wait` ou `nowait`).
- Pour les commandes distantes exécutées par le proxy Zabbix, définissez le paramètre `EnableRemoteCommands` sur 1 dans la configuration du proxy. Notez qu'il n'est pas nécessaire d'activer les commandes distantes sur le proxy Zabbix lorsqu'elles sont exécutées par l'agent Zabbix sur un hôte surveillé par le proxy.

Attention:
N'utilisez pas `AllowKey=system.run[*]` ou `AllowKeyRegexp=^system\.run\[.*\]$` sauf si cela est nécessaire, car ces paramètres autorisent l'exécution de n'importe quelle commande sur l'hôte. Pour des raisons de sécurité, il est recommandé d'autoriser uniquement les commandes exactes, tandis que `system.run[*]` est refusé par défaut (`DenyKey=system.run[*]`).

Voir également Restriction des vérifications de l'agent.

Limite des commandes distantes

La limite des commandes distantes après résolution de toutes les macros dépend du type de base de données et du jeu de caractères (les caractères non ASCII nécessitent plus d'un octet pour être stockés) :

Base de données	Limite en caractères	Limite en octets
MySQL	65535	65535
PostgreSQL	65535	non limitée
SQLite (Zabbix proxy uniquement)	65535	non limitée

La sortie de l'exécution d'une commande distante (valeur de retour) est limitée à 16 Mo (y compris les espaces de fin qui sont tronqués). La limite des commandes distantes IPMI dépend de la bibliothèque IPMI installée. Notez que les limites de la base de données s'appliquent à toutes les commandes distantes.

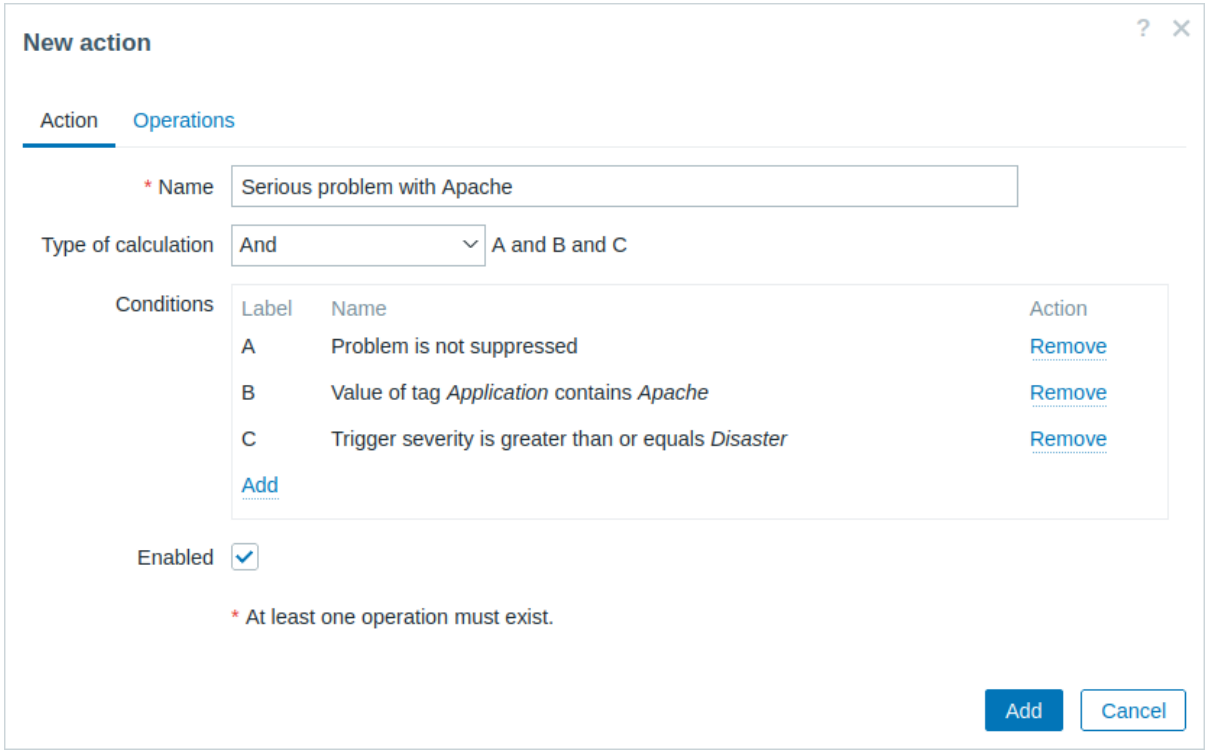
Configuration

Les commandes distantes exécutées par l'agent Zabbix doivent d'abord être autorisées dans la configuration de l'agent à l'aide du paramètre `AllowKey` ou `AllowKeyRegexp`.

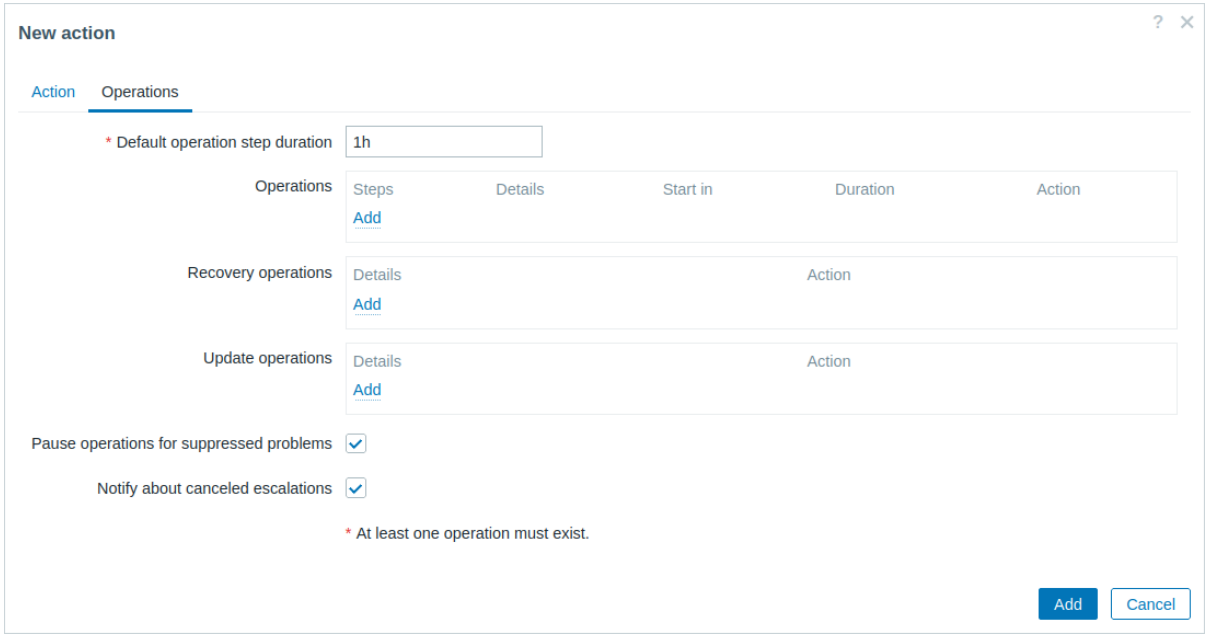
Assurez-vous que le paramètre `AllowKey=system.run[<command>,*]` ou `AllowKeyRegexp=^system\.run\[<command>,.*\]$` est ajouté à la configuration de l'agent pour chaque commande autorisée afin de permettre son exécution en mode `nowait`. Redémarrez le daemon de l'agent après avoir modifié le paramètre.

Ensuite, lors de la configuration d'une nouvelle action dans *Alertes* > *Actions* > *Actions de déclencheur* :

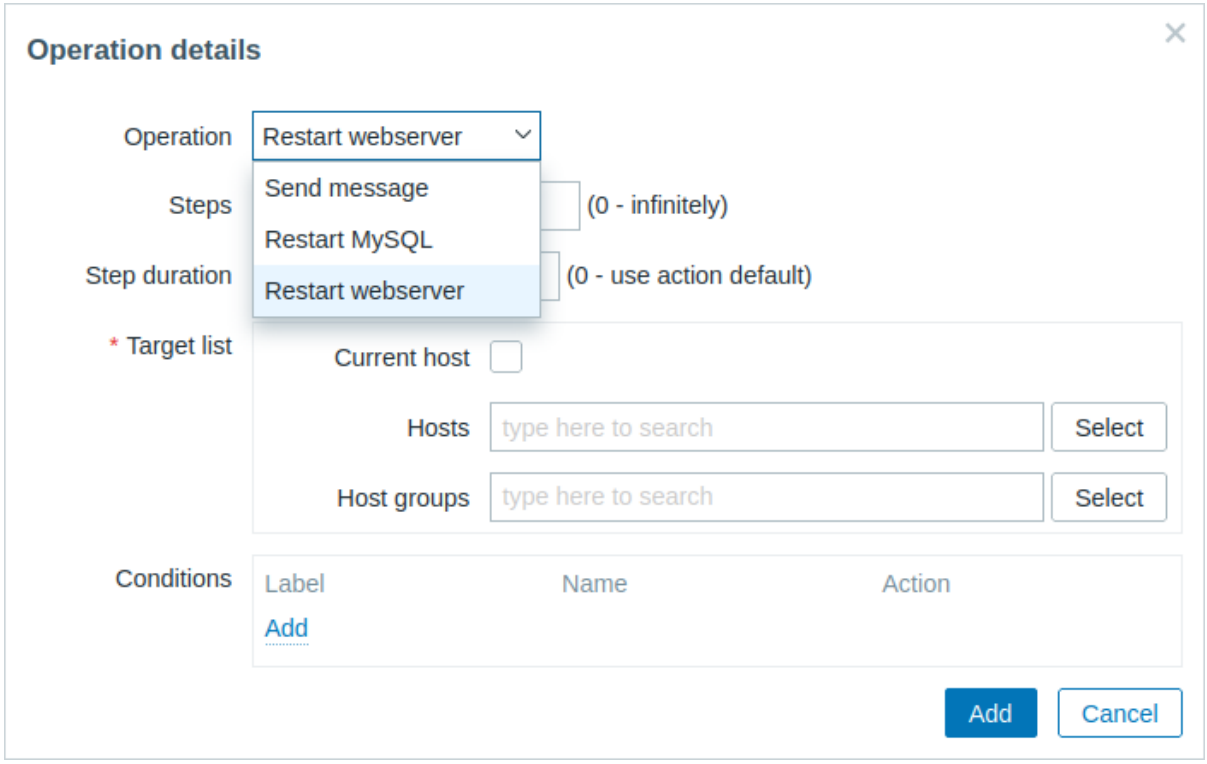
1. Définissez les conditions appropriées, par exemple en configurant l'activation de l'action lors de tout problème de gravité catastrophe concernant l'une des applications Apache.



2. Dans l'onglet *Opérations*, cliquez sur *Ajouter* dans la section *Opérations*, *Opérations de récupération* ou *Opérations de mise à jour*.



3. Sélectionnez l'un des scripts prédéfinis dans la liste déroulante *Opération* et définissez la *Liste des cibles* du script.



Scripts prédéfinis

Les scripts disponibles pour les opérations d'action (webhook, script, SSH, Telnet, IPMI) sont définis dans les scripts globaux avec *Action operation* sélectionné comme portée.

Par exemple :

```
sudo /etc/init.d/apache restart
```

Dans ce cas, Zabbix tentera de redémarrer un processus Apache. Avec cette commande, assurez-vous que la commande est exécutée sur l'agent Zabbix (cliquez sur le bouton *Zabbix agent* en regard de *Execute on*).

Attention:
Notez l'utilisation de sudo - l'utilisateur Zabbix ne dispose pas par défaut des permissions nécessaires pour redémarrer les services système. Voir ci-dessous pour des indications sur la configuration de sudo.

Note:
À partir de Zabbix agent 7.0, les commandes distantes peuvent également être exécutées sur un agent fonctionnant en mode actif. L'agent Zabbix - qu'il soit actif ou passif - doit s'exécuter sur l'hôte distant et exécute les commandes en arrière-plan.

Les commandes distantes sur Zabbix agent sont exécutées sans délai d'attente par la clé `system.run[,nowait]` et ne sont pas vérifiées pour les résultats d'exécution. Sur Zabbix serveur et Zabbix proxy, les commandes distantes sont exécutées avec le délai d'attente défini dans le paramètre `TrapperTimeout` du fichier `zabbix_server.conf` ou `zabbix_proxy.conf` et sont vérifiées pour les résultats d'exécution. Pour plus d'informations, voir *Script timeout*.

Permissions d'accès

Assurez-vous que l'utilisateur `zabbix` dispose des permissions d'exécution pour les commandes configurées. Il peut être utile d'utiliser sudo pour donner accès à des commandes privilégiées. Pour configurer l'accès, exécutez en tant que root :

```
visudo
```

Exemples de lignes pouvant être utilisées dans le fichier `sudoers` :

```
# permet à l'utilisateur 'zabbix' d'exécuter toutes les commandes sans mot de passe.
zabbix ALL=NOPASSWD: ALL

# permet à l'utilisateur 'zabbix' de redémarrer apache sans mot de passe.
zabbix ALL=NOPASSWD: /etc/init.d/apache restart
```

Note:
Sur certains systèmes, le fichier `sudoers` empêchera les utilisateurs non locaux d'exécuter des commandes. Pour modifier ce comportement, commentez l'option `requiretty` dans `/etc/sudoers`.

Commandes distantes avec plusieurs interfaces

Si le système cible possède plusieurs interfaces du type sélectionné (agent Zabbix ou IPMI), les commandes distantes seront exécutées sur l'interface par défaut.

Il est possible d'exécuter des commandes distantes via SSH et Telnet en utilisant une autre interface que celle de l'agent Zabbix. L'interface disponible à utiliser est sélectionnée dans l'ordre suivant :

- Interface par défaut de l'agent Zabbix
- Interface par défaut SNMP
- Interface par défaut JMX
- Interface par défaut IPMI

Commandes à distance IPMI

Pour les commandes distantes IPMI, la syntaxe suivante doit être utilisée :

```
<command> [<value>]
```

où

- <command> - une des commandes IPMI sans espaces
- <value> - 'on', 'off' ou tout entier non signé. <valeur> est un paramètre facultatif.

Exemples

Exemples de scripts globaux pouvant être utilisés comme commandes distantes dans les opérations d'action.

Exemple 1

Redémarrage de Windows dans certaines conditions.

Afin de redémarrer automatiquement Windows lorsqu'un problème est détecté par Zabbix, définissez le script suivant :

Script parameter	Value
Scope	'Action operation'
Type	'Script'
Command	`c:\windows\system32\shutdown.exe -r -f`

Exemple 2

Redémarrez l'hôte en utilisant le contrôle IPMI.

Paramètre du script	Valeur
Portée	'Action operation'
Type	'IPMI'
Commande	reset

Exemple 3

Éteignez l'hôte en utilisant le contrôle IPMI.

Paramètre du script	Valeur
Portée	'Action operation'
Type	'IPMI'
Commande	power off

Opérations supplémentaires

Aperçu

Dans cette section, vous trouverez quelques détails sur des opérations supplémentaires pour les événements de découverte/enregistrement automatique.

Ajout d'un hôte

Les hôtes sont ajoutés pendant le processus de découverte, dès qu'un hôte est découvert, plutôt qu'à la fin du processus de découverte.

> **Note:**
> Comme la découverte réseau peut prendre un certain temps, en raison de nombreux hôtes/services indisponibles, il est recommandé de faire preuve de patience et d'utiliser des plages d'IP raisonnables.

Lors de l'ajout d'un hôte, son nom est déterminé par la fonction standard **gethostbyname**. Si l'hôte peut être résolu, le nom résolu est utilisé. Sinon, l'adresse IP est utilisée. En outre, si une adresse IPv6 doit être utilisée comme nom d'hôte, alors tous les ":" (deux-points) sont remplacés par "_" (tirets bas), car les deux-points ne sont pas autorisés dans les noms d'hôte.

> **Attention:**
> Si la découverte est effectuée par un proxy, actuellement la résolution du nom d'hôte a toujours lieu sur le serveur Zabbix.

> **Attention:**
> Si un hôte existe déjà dans la configuration Zabbix avec le même nom qu'un hôte nouvellement découvert, Zabbix ajoutera **_N** au nom d'hôte, où **N** est un nombre croissant, à partir de 2.

Utilisation des macros dans les messages

Aperçu

Dans les objets et le texte des messages, vous pouvez utiliser des macros pour un signalement des problèmes plus efficace.

En plus d'un certain nombre de macros intégrées, les macros utilisateur et les macros d'expression sont également prises en charge. Une liste complète des macros prises en charge par Zabbix est disponible.

Exemples

Les exemples illustrent ici comment vous pouvez utiliser des macros dans les messages.

Exemple 1

Objet du message :

```
Problem: {TRIGGER.NAME}
```

Lorsque vous recevez le message, l'objet du message sera remplacé par quelque chose comme :

```
Problem: Processor load is too high on Zabbix server
```

Exemple 2

Message :

```
La charge du processeur est : {?last(/zabbix.zabbix.com/system.cpu.load[,avg1])}
```

Lorsque vous recevez le message, celui-ci sera remplacé par quelque chose comme :

```
La charge du processeur est : 1.45
```

Exemple 3

Message :

```
Dernière valeur : {?last(/{HOST.HOST}/{ITEM.KEY})}
MAX sur 15 minutes : {?max(/{HOST.HOST}/{ITEM.KEY},15m)}
MIN sur 15 minutes : {?min(/{HOST.HOST}/{ITEM.KEY},15m)}
```

Lorsque vous recevez le message, celui-ci sera remplacé par quelque chose comme :

```
Dernière valeur : 1.45
MAX sur 15 minutes : 2.33
MIN sur 15 minutes : 1.01
```

Exemple 4

Message :

```
http://<server_ip_or_name>/zabbix/tr_events.php?triggerid={TRIGGER.ID}&eventid={EVENT.ID}
```

Lorsque vous recevez le message, il contient un lien vers la page *Détails de l'événement*, qui fournit des informations sur l'événement, son déclencheur et une liste des derniers événements générés par le même déclencheur.

Exemple 5

Informer sur les valeurs de plusieurs hôtes dans une expression de déclencheur.

Message :

```
Nom du problème : {TRIGGER.NAME}
Expression du déclencheur : {TRIGGER.EXPRESSION}

1. Valeur de l'élément sur {HOST.NAME1} : {ITEM.VALUE1} ({ITEM.NAME1})
2. Valeur de l'élément sur {HOST.NAME2} : {ITEM.VALUE2} ({ITEM.NAME2})
```

Lorsque vous recevez le message, celui-ci sera remplacé par quelque chose comme :

```
Nom du problème : La charge du processeur est trop élevée sur un hôte local
Expression du déclencheur : last(/Myhost/system.cpu.load[percpu,avg1])>5 or last(/Myotherhost/system.cpu.l

1. Valeur de l'élément sur Myhost : 0.83 (Charge du processeur (moyenne sur 1 min par cœur))
2. Valeur de l'élément sur Myotherhost : 5.125 (Charge du processeur (moyenne sur 1 min par cœur))
```

Exemple 6

Réception des détails à la fois de l'événement de problème et de l'événement de rétablissement dans un message de rétablissement :

Message :

```
Problème :

ID de l'événement : {EVENT.ID}
Valeur de l'événement : {EVENT.VALUE}
Statut de l'événement : {EVENT.STATUS}
Heure de l'événement : {EVENT.TIME}
Date de l'événement : {EVENT.DATE}
Âge de l'événement : {EVENT.AGE}
Accusé de réception de l'événement : {EVENT.ACK.STATUS}
Historique des mises à jour de l'événement : {EVENT.UPDATE.HISTORY}

Rétablissement :

ID de l'événement : {EVENT.RECOVERY.ID}
Valeur de l'événement : {EVENT.RECOVERY.VALUE}
Statut de l'événement : {EVENT.RECOVERY.STATUS}
Heure de l'événement : {EVENT.RECOVERY.TIME}
Date de l'événement : {EVENT.RECOVERY.DATE}
Données opérationnelles : {EVENT.OPDATA}
```

Lorsque vous recevez le message, les macros seront remplacées par quelque chose comme :

```
Problème :

ID de l'événement : 21874
Valeur de l'événement : 1
Statut de l'événement : PROBLEM
Heure de l'événement : 13:04:30
Date de l'événement : 2018.01.02
Âge de l'événement : 5m 0s
Accusé de réception de l'événement : Oui
Historique des mises à jour de l'événement : 2018.01.02 13:05:51 "John Smith (Admin)"
Actions : acknowledged.
```

```
Rétablissement :

ID de l'événement : 21896
Valeur de l'événement : 0
Statut de l'événement : OK
Heure de l'événement : 13:10:07
Date de l'événement : 2018.01.02
Données opérationnelles : La valeur actuelle est 0.83
```

Opérations de récupération

Aperçu

Les opérations de récupération vous permettent d'être averti lorsque les problèmes sont résolus.

Les messages et les commandes à distance sont pris en charge dans les opérations de récupération. Bien que plusieurs opérations puissent être ajoutées, l'escalade n'est pas prise en charge - toutes les opérations sont affectées à une seule étape et seront donc exécutées simultanément.

Cas d'usage

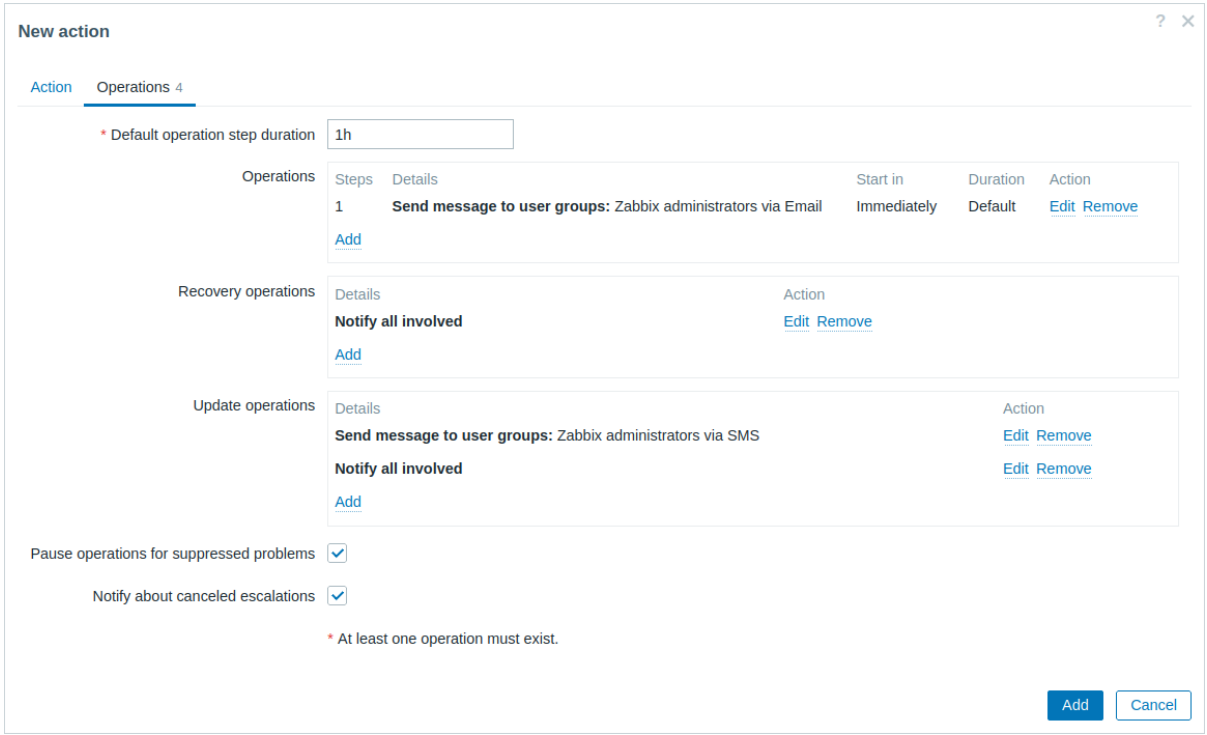
Certains cas d'usage pour les opérations de récupération sont les suivants :

1. Avertir lors d'une récupération tous les utilisateurs qui ont été informés du problème :
   - Sélectionnez 'Envoyer un message de récupération' comme type d'opération.
2. Avoir plusieurs opérations lors de la récupération : envoyer une notification et exécuter une commande à distance :
   - Ajouter des types d'opérations pour envoyer un message et exécuter une commande.
3. Ouvrir un ticket dans le système d'assistance/ticketing externe et le fermer lorsque le problème est résolu :
   - Créer un script externe qui communique avec le système d'assistance.
   - Créer une action ayant une opération qui exécute ce script et ouvre ainsi un ticket.
   - Avoir une opération de récupération qui exécute ce script avec d'autres paramètres et ferme le ticket.
   - Utiliser la macro {EVENT.ID} pour référencer le problème d'origine.

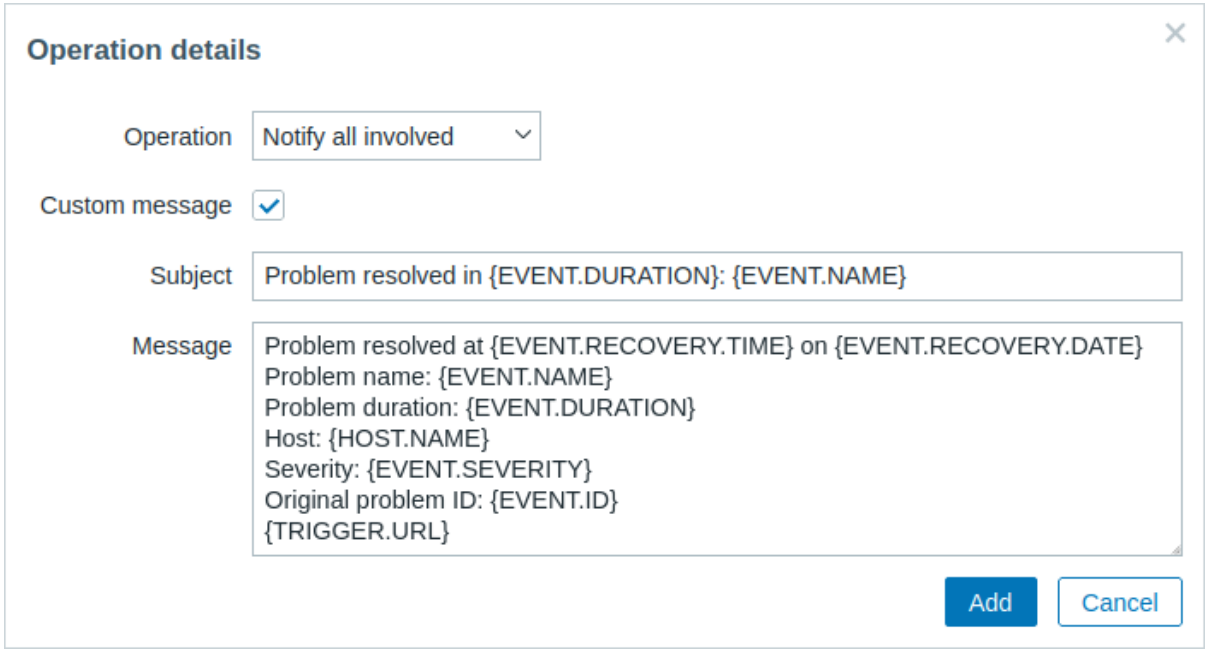
Configuration d'une opération de récupération

Pour configurer une opération de récupération, suivez ces étapes :

1. Accédez à l'onglet *Operations* dans le formulaire de configuration d'une action.
2. Pour configurer une nouvelle opération de récupération, cliquez sur *Add* dans la section *Recovery operations*. Pour modifier une opération existante, cliquez sur *Edit* à côté de l'opération.
3. Configurez les détails de l'opération.



Détails de l'opération de récupération



Trois types d'opérations sont disponibles pour les événements de récupération :

- **Envoyer un message** - envoyer un message de récupération à l'utilisateur spécifié ;
- **Notifier toutes les personnes impliquées** - envoyer un message de récupération à tous les utilisateurs qui ont été notifiés lors de l'événement de problème ;
- **<nom de la commande distante>** - exécuter une commande distante. Les commandes peuvent être exécutées si elles ont été définies au préalable dans les scripts globaux avec *Opération d'action* sélectionné comme portée.

Les paramètres de chaque type d'opération sont décrits ci-dessous. Tous les champs de saisie obligatoires sont marqués d'un astérisque rouge. Une fois terminé, cliquez sur *Ajouter* pour ajouter l'opération à la liste des *Opérations de récupération*.

Note:
Notez que si le même destinataire est défini dans plusieurs types d'opérations sans *Message personnalisé* spécifié, les notifications en double ne sont pas envoyées.

Type d'opération : envoyer un message

Paramètre	Description
Envoyer aux groupes d'utilisateurs	Sélectionnez les groupes d'utilisateurs auxquels envoyer le message de récupération. Le groupe d'utilisateurs doit disposer au minimum des permissions « lecture » sur l'hôte afin d'être notifié.
Envoyer aux utilisateurs	Sélectionnez les utilisateurs auxquels envoyer le message de récupération. L'utilisateur doit disposer au minimum des permissions « lecture » sur l'hôte afin d'être notifié.
Envoyer au type de média	Envoyez le message de récupération par défaut à tous les types de média disponibles (configurés et activés) ou à un seul type spécifique.
Message personnalisé	Si cette option est sélectionnée, un message personnalisé peut être défini.
Sujet	Sujet du message personnalisé. Le sujet peut contenir des macros.
Message	Le message personnalisé. Le message peut contenir des macros.

Type d'opération : commande distante

Paramètre	Description
Liste des cibles	Sélectionnez les cibles sur lesquelles exécuter la commande : **Hôte actuel** - la commande est exécutée sur l'hôte du déclencheur à l'origine de l'événement de problème. Cette option ne fonctionnera pas si le déclencheur concerne plusieurs hôtes. **Hôte** - sélectionnez le ou les hôtes sur lesquels exécuter la commande. **Groupe d'hôtes** - sélectionnez le ou les groupes d'hôtes sur lesquels exécuter la commande. La spécification d'un groupe d'hôtes parent sélectionne implicitement tous les groupes d'hôtes imbriqués. Ainsi, la commande distante sera également exécutée sur les hôtes des groupes imbriqués. Une commande sur un hôte n'est exécutée qu'une seule fois, même si l'hôte correspond plusieurs fois (par exemple, depuis plusieurs groupes d'hôtes, individuellement et depuis un groupe d'hôtes). La liste des cibles n'a aucun effet si la commande est exécutée sur le serveur Zabbix. Dans ce cas, sélectionner davantage de cibles entraîne uniquement l'exécution répétée de la commande sur le serveur. Notez que pour les scripts globaux, la sélection des cibles dépend également du paramètre *Groupe d'hôtes* dans la configuration du script global.

Type d'opération : notifier toutes les parties impliquées

Paramètre		Description
Message personnalisé		Si cette option est sélectionnée, un message personnalisé peut être défini.
	Sujet	Sujet du message personnalisé. Le sujet peut contenir des macros.
	Message	Le message personnalisé. Le message peut contenir des macros.

Opérations de mise à jour

Aperçu

Les opérations de mise à jour sont disponibles dans les actions avec les sources d'événements suivantes :

- *Déclencheurs* - lorsque des problèmes sont mis à jour par d'autres utilisateurs, c'est-à-dire commentés, reconnus, dont la gravité a été modifiée, fermés (manuellement) ;
- *Services* - lorsque la gravité d'un service a changé, mais que le service n'est toujours pas rétabli.

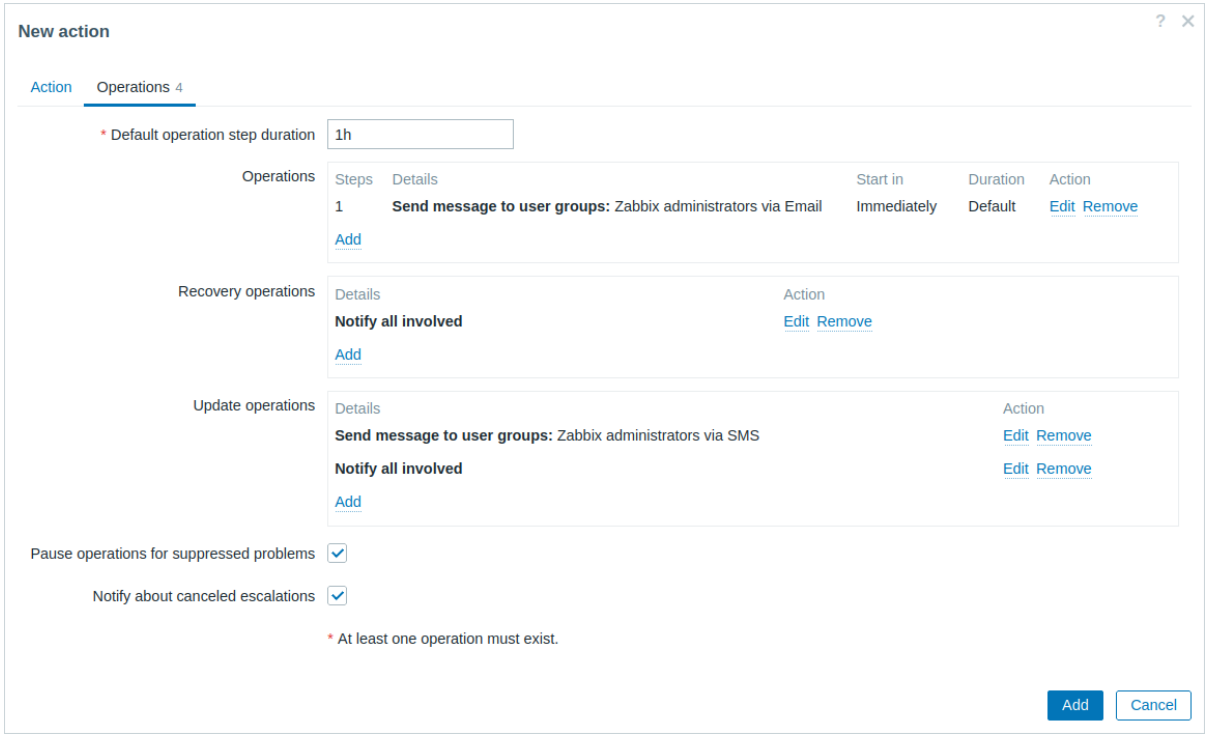
Veuillez noter que les utilisateurs ne reçoivent pas de notifications concernant leurs propres mises à jour.

Les messages et les commandes à distance sont tous deux pris en charge dans les opérations de mise à jour. Bien que plusieurs opérations puissent être ajoutées, l'escalade n'est pas prise en charge - toutes les opérations sont affectées à une seule étape et seront donc exécutées simultanément.

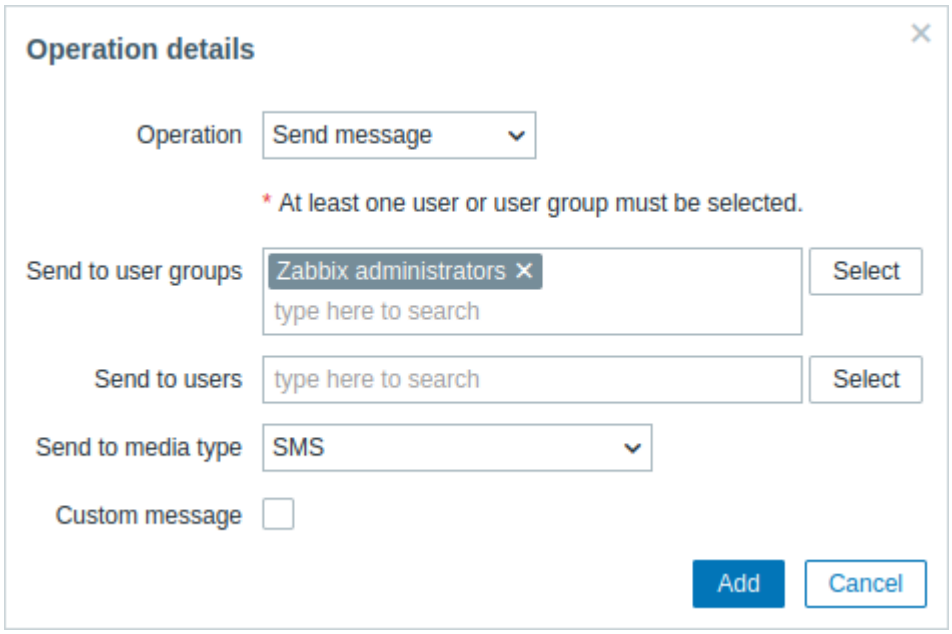
Configuration d'une opération de mise à jour

Pour configurer une opération de mise à jour, suivez ces étapes :

1. Accédez à l'onglet *Operations* dans le formulaire de configuration d'une action.
2. Pour configurer une nouvelle opération de mise à jour, cliquez sur *Add* dans la section *Update operations*. Pour modifier une opération existante, cliquez sur *Edit* à côté de l'opération.
3. Configurez les détails de l'opération.



Détails de l'opération de mise à jour



Les opérations de mise à jour offrent le même ensemble de paramètres que les opérations de récupération.

Escalades

Aperçu

Avec les escalades, vous pouvez créer des scénarios personnalisés pour l'envoi de notifications ou l'exécution de commandes à distance.

En pratique, cela signifie que :

- Les utilisateurs peuvent être informés immédiatement des nouveaux problèmes.
- Les notifications peuvent être répétées jusqu'à la résolution du problème.
- L'envoi d'une notification peut être différé.
- Les notifications peuvent être escaladées vers un autre groupe d'utilisateurs de niveau « supérieur ».
- Des commandes à distance peuvent être exécutées immédiatement ou lorsqu'un problème n'est pas résolu pendant une longue période.

Les actions sont escaladées en fonction de l'**étape d'escalade**. Chaque étape a une durée.

Vous pouvez définir à la fois la durée par défaut et une durée personnalisée pour une étape individuelle. La durée minimale d'une étape d'escalade est de 60 secondes.

Vous pouvez démarrer des actions, telles que l'envoi de notifications ou l'exécution de commandes, à partir de n'importe quelle étape. L'étape 1 est destinée aux actions immédiates. Si vous souhaitez différer une action, vous pouvez l'affecter à une étape ultérieure. Pour chaque étape, plusieurs actions peuvent être définies.

Le nombre d'étapes d'escalade n'est pas limité.

Les escalades sont définies lors de la configuration d'une opération. Les escalades sont prises en charge uniquement pour les opérations de problème, pas pour la récupération.

Aspects divers du comportement des escalades

Examinons ce qui se passe dans différentes circonstances lorsqu'une action contient plusieurs étapes d'escalade.

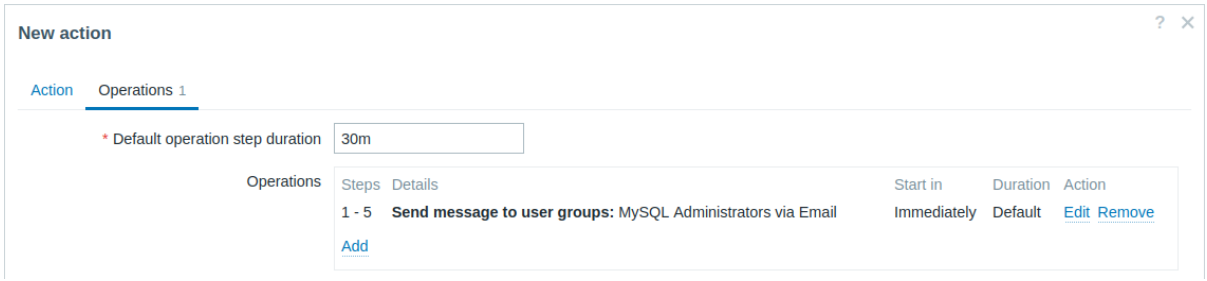
Situation	Comportement
Le hôte concerné passe en maintenance après l'envoi de la notification initiale du problème	Selon le paramètre *Mettre en pause les opérations pour les problèmes supprimés* dans la configuration de l'action, toutes les étapes d'escalade restantes sont exécutées soit avec un délai dû à la période de maintenance, soit sans délai. Une période de maintenance n'annule pas les opérations.
*La période définie dans la condition d'action **Période** se termine après l'envoi de la notification initiale*	Toutes les étapes d'escalade restantes sont exécutées. La condition *Période* ne peut pas arrêter les opérations ; elle agit sur le démarrage ou non des actions, et non sur les opérations.
Un problème commence pendant une maintenance et persiste (n'est pas résolu) après la fin de la maintenance	Selon le paramètre *Mettre en pause les opérations pour les problèmes supprimés* dans la configuration de l'action, toutes les étapes d'escalade sont exécutées soit à partir du moment où la maintenance se termine, soit immédiatement.
Un problème commence pendant une maintenance sans données et persiste (n'est pas résolu) après la fin de la maintenance	Il doit attendre que le déclencheur se déclenche avant que toutes les étapes d'escalade soient exécutées.
Différentes escalades se succèdent rapidement et se chevauchent	L'exécution de chaque nouvelle escalade remplace l'escalade précédente, mais au moins une étape d'escalade est toujours exécutée pour l'escalade précédente. Ce comportement est pertinent pour les actions liées aux événements créés à CHAQUE évaluation du problème par le déclencheur.
Pendant une escalade en cours (par exemple, lors de l'envoi d'un message), selon tout type d'événement : - l'action est désactivée Selon un événement de déclencheur : - le déclencheur est désactivé - le hôte ou l'élément est désactivé Selon un événement interne concernant les déclencheurs : - le déclencheur est désactivé Selon un événement interne concernant les éléments/règles de découverte de bas niveau : - l'élément est désactivé - le hôte est désactivé	Le message en cours est envoyé, puis un message supplémentaire est envoyé pour l'escalade. Le message de suivi contient au début du corps du message le texte d'annulation (*REMARQUE : Escalade annulée*) indiquant la raison (par exemple, *REMARQUE : Escalade annulée : action « <Action name> » désactivée*). Le destinataire est ainsi informé que l'escalade est annulée et qu'aucune autre étape ne sera exécutée. Ce message est envoyé à toutes les personnes ayant précédemment reçu les notifications. La raison de l'annulation est également enregistrée dans le fichier journal du serveur (à partir du niveau de débogage 3=Avertissement). Notez que le message *Escalade annulée* est également envoyé si les opérations sont terminées, mais que des opérations de récupération sont configurées et ne sont pas encore exécutées.
Pendant une escalade en cours (par exemple, lors de l'envoi d'un message), l'action est supprimée	Aucun autre message n'est envoyé. L'information est enregistrée dans le fichier journal du serveur (à partir du niveau de débogage 3=Avertissement), par exemple : `escalation canceled: action id:334 deleted`

Exemples d'escalade

Exemple 1

Envoi d'une notification répétée toutes les 30 minutes (5 fois au total) à un groupe « MySQL Administrators ». Pour configurer cela :

- Dans l'onglet *Operations*, définissez *Default operation step duration* sur « 30m » (30 minutes).
- Définissez les *Steps* de l'escalade de « 1 » à « 5 ».
- Sélectionnez le groupe « MySQL Administrators » comme destinataires du message.



Les notifications seront envoyées à 0:00, 0:30, 1:00, 1:30 et 2:00 après le début du problème (sauf si, bien sûr, le problème est résolu plus tôt).

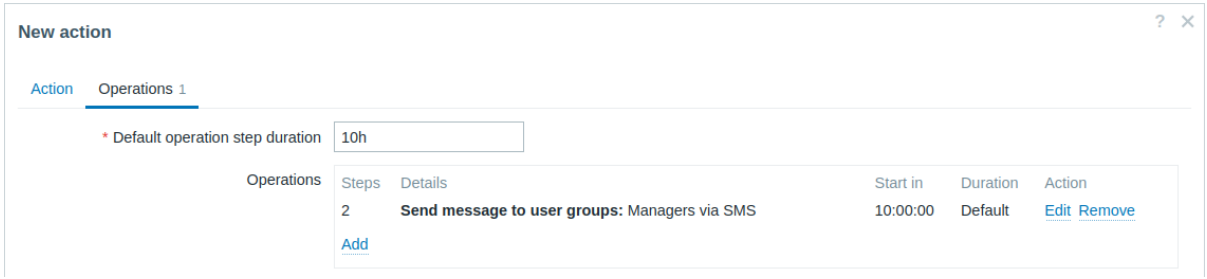
Si le problème est résolu et qu'un message de rétablissement est configuré, il sera envoyé à ceux qui ont reçu au moins un message de problème dans ce scénario d'escalade.

Note:
Si le déclencheur qui a généré une escalade active est désactivé, Zabbix envoie un message d'information à toutes les personnes qui ont déjà reçu des notifications.

Exemple 2

Envoi d'une notification différée concernant un problème de longue durée. Pour configurer :

- Dans l'onglet *Operations*, définissez *Default operation step duration* sur « 10h » (10 heures).
- Définissez les *Steps* de l'escalade de « 2 » à « 2 ».



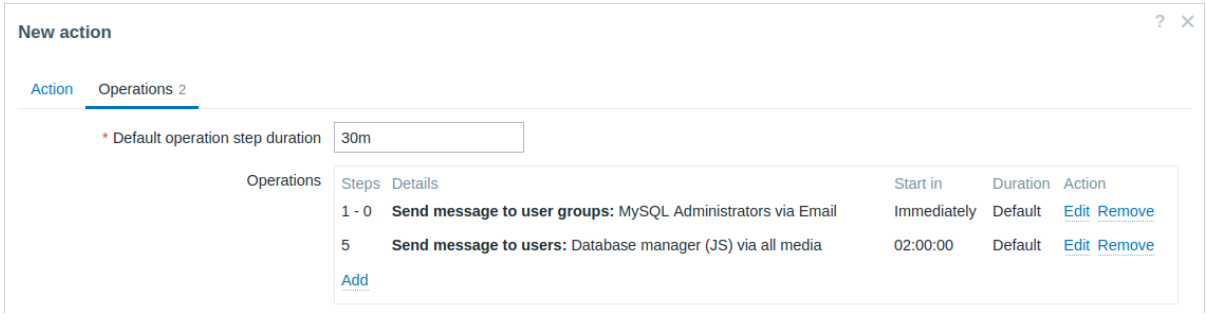
Une notification ne sera envoyée qu'à l'étape 2 du scénario d'escalade, soit 10 heures après le début du problème.

Vous pouvez personnaliser le texte du message avec quelque chose comme « Le problème dure depuis plus de 10 heures ».

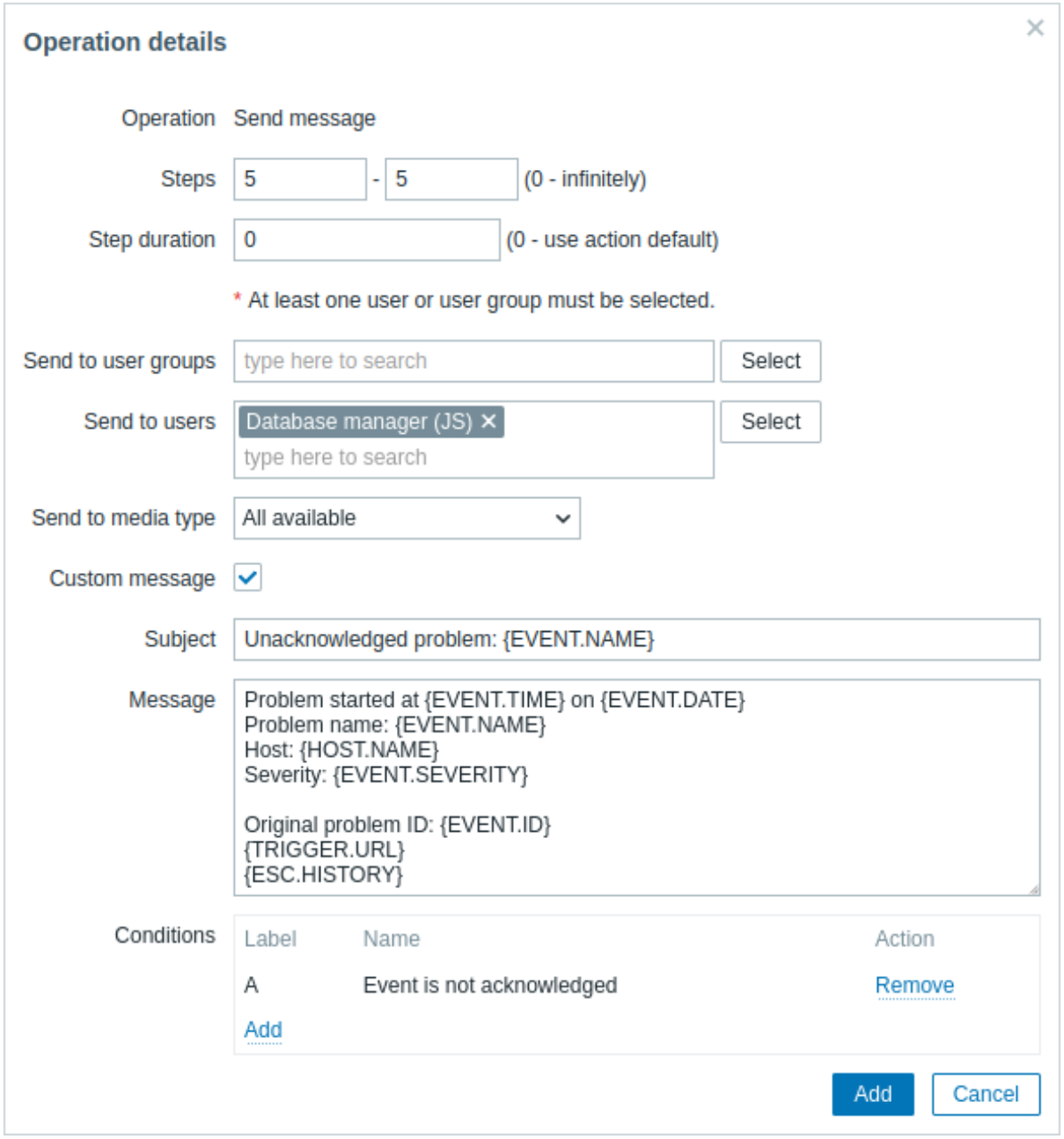
Exemple 3

Escalade du problème vers le responsable.

Dans le premier exemple ci-dessus, nous avons configuré l'envoi périodique de messages aux administrateurs MySQL. Dans ce cas, les administrateurs recevront quatre messages avant que le problème ne soit escaladé au responsable de la base de données. Notez que le responsable ne recevra un message que si le problème n'a pas encore été acquitté, ce qui signifie vraisemblablement que personne ne s'en occupe.



Détails de l'opération 2 :



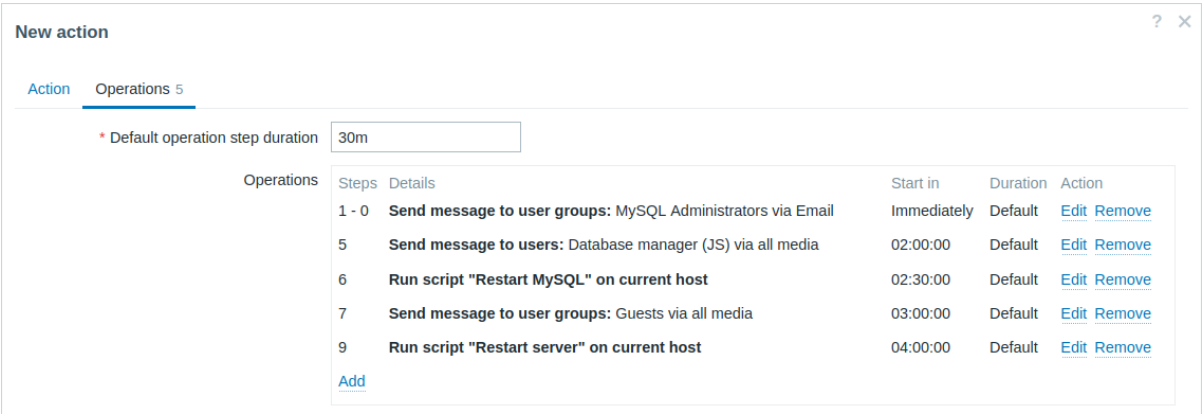
Notez l'utilisation de la macro {ESC.HISTORY} dans le message personnalisé. La macro contiendra des informations sur toutes les étapes précédemment exécutées de cette escalade, telles que les notifications envoyées et les commandes exécutées.

Exemple 4

Un scénario plus complexe. Après plusieurs messages aux administrateurs MySQL et une escalade vers le responsable, Zabbix essaiera de redémarrer la base de données MySQL. Cela se produira si le problème existe depuis 2h30 et qu'il n'a pas été acquitté.

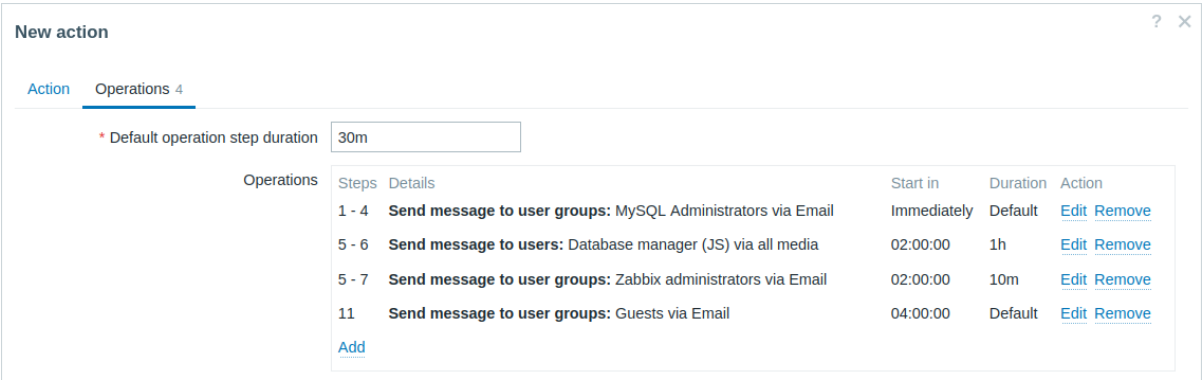
Si le problème persiste, après 30 minutes supplémentaires, Zabbix enverra un message à tous les utilisateurs invités.

Si cela ne résout pas le problème, après une heure supplémentaire, Zabbix redémarrera le serveur avec la base de données MySQL (deuxième commande à distance) à l'aide des commandes IPMI.



Exemple 5

Une escalade avec plusieurs opérations ayant des plages d'étapes qui se chevauchent et des intervalles personnalisés. La durée d'étape par défaut de l'opération est de 30 minutes.



Les notifications seront envoyées comme suit :

- Aux administrateurs MySQL à 0:00, 0:30, 1:00 et 1:30 après le début du problème.
- Au responsable de la base de données à 2:00 et 2:10 (la durée d'étape personnalisée plus courte de 10 minutes définie dans l'opération suivante remplace la durée d'étape plus longue d'1 heure configurée ici, comme indiqué dans Détails de l'opération pour *Durée d'étape* lorsque les étapes se chevauchent).
- Aux administrateurs Zabbix à 2:00, 2:10 et 2:20 après le début du problème (la durée d'étape personnalisée de 10 minutes est appliquée).
- Aux utilisateurs invités à 4:00 après le début du problème (la durée d'étape par défaut de 30 minutes s'applique entre les étapes 8 et 11).

Réception de notifications sur les éléments non pris en charge

Aperçu

Il est possible de recevoir des notifications sur les éléments non pris en charge dans Zabbix.

Cela fait partie du concept d'événements internes dans Zabbix, qui permet aux utilisateurs d'être avertis dans ces cas. Les événements internes reflètent un changement d'état :

- lorsque les éléments passent de « normal » à « non pris en charge » (et inversement) ;
- lorsque les déclencheurs passent de « normal » à « inconnu » (et inversement) ;
- lorsque les règles de découverte de bas niveau passent de « normal » à « non pris en charge » (et inversement).

Cette section présente une procédure pour **recevoir une notification** lorsqu'un élément devient non pris en charge.

Configuration

Globalement, le processus de configuration des notifications devrait sembler familier à ceux qui ont déjà configuré des alertes dans Zabbix.

Étape 1

Configurez un support de notification, tel qu'un e-mail, un SMS ou un script, à utiliser pour les notifications. Reportez-vous aux sections correspondantes du manuel pour effectuer cette tâche.

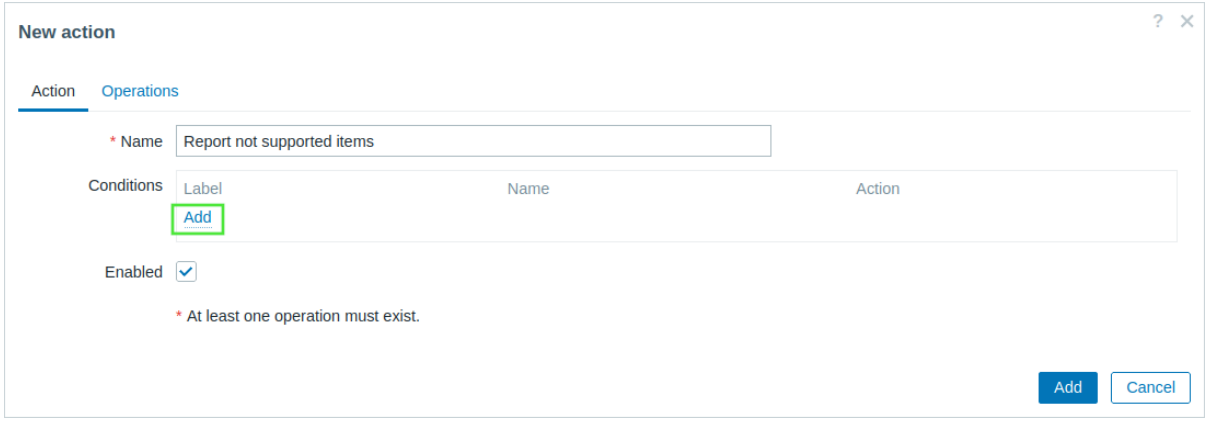
Attention:
Pour les notifications sur les événements internes, la sévérité par défaut (’Non classé’) est utilisée ; laissez donc cette option cochée lors de la configuration des supports de notification de l’utilisateur si vous souhaitez recevoir des notifications pour les événements internes.

Étape 2

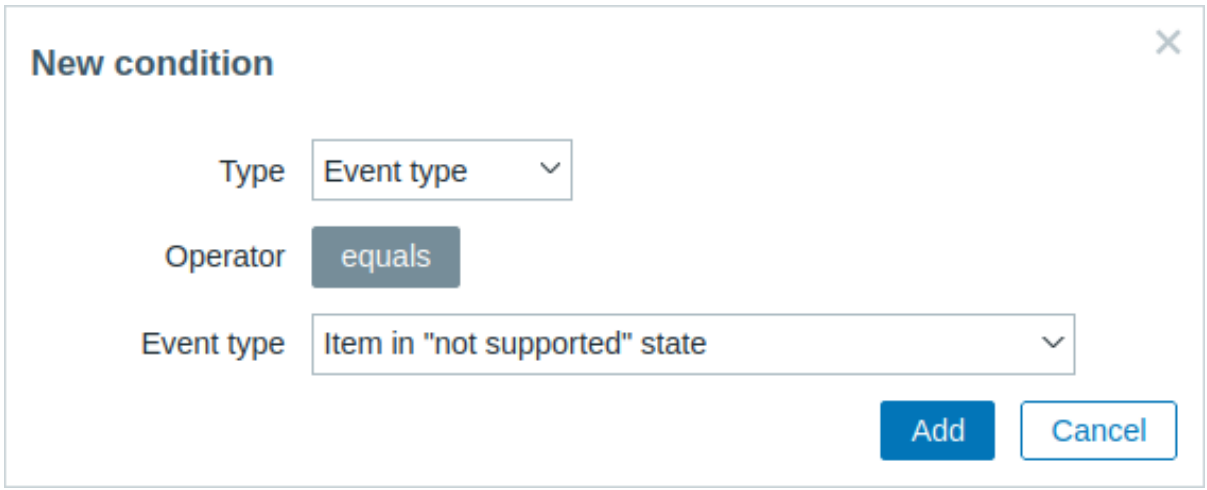
1. Accédez à *Alerts* > *Actions* > *Internal actions*.
2. Cliquez sur *Create action* dans le coin supérieur droit de la page pour ouvrir un formulaire de configuration d’action.

Étape 3

1. Dans l’onglet *Action*, saisissez un nom pour l’action.
2. Cliquez sur *Ajouter* dans le bloc *Conditions* pour ajouter une nouvelle condition.



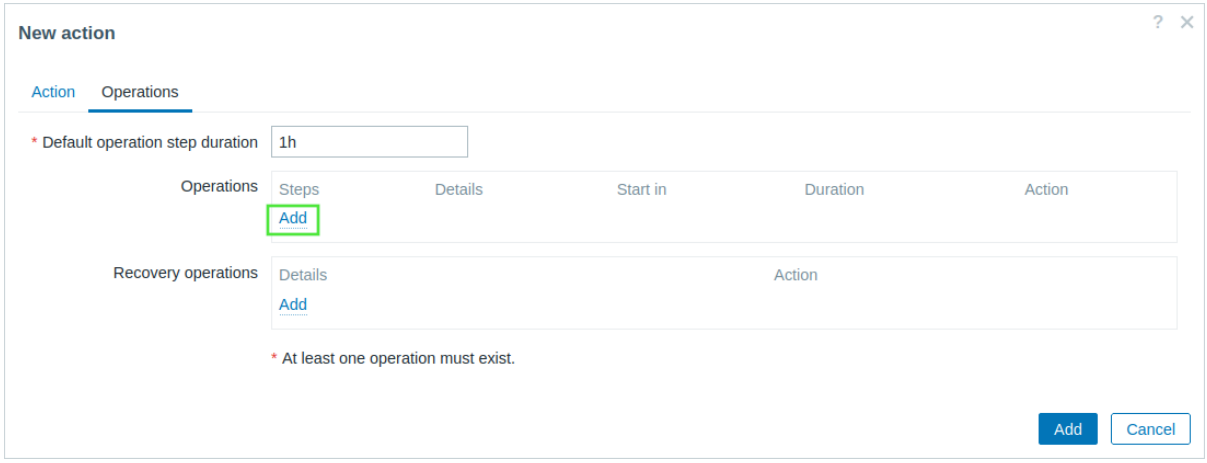
3. Dans la fenêtre contextuelle *Nouvelle condition*, sélectionnez « Type d’événement » comme type de condition, puis sélectionnez « Élément dans l’état « non pris en charge » » comme type d’événement.



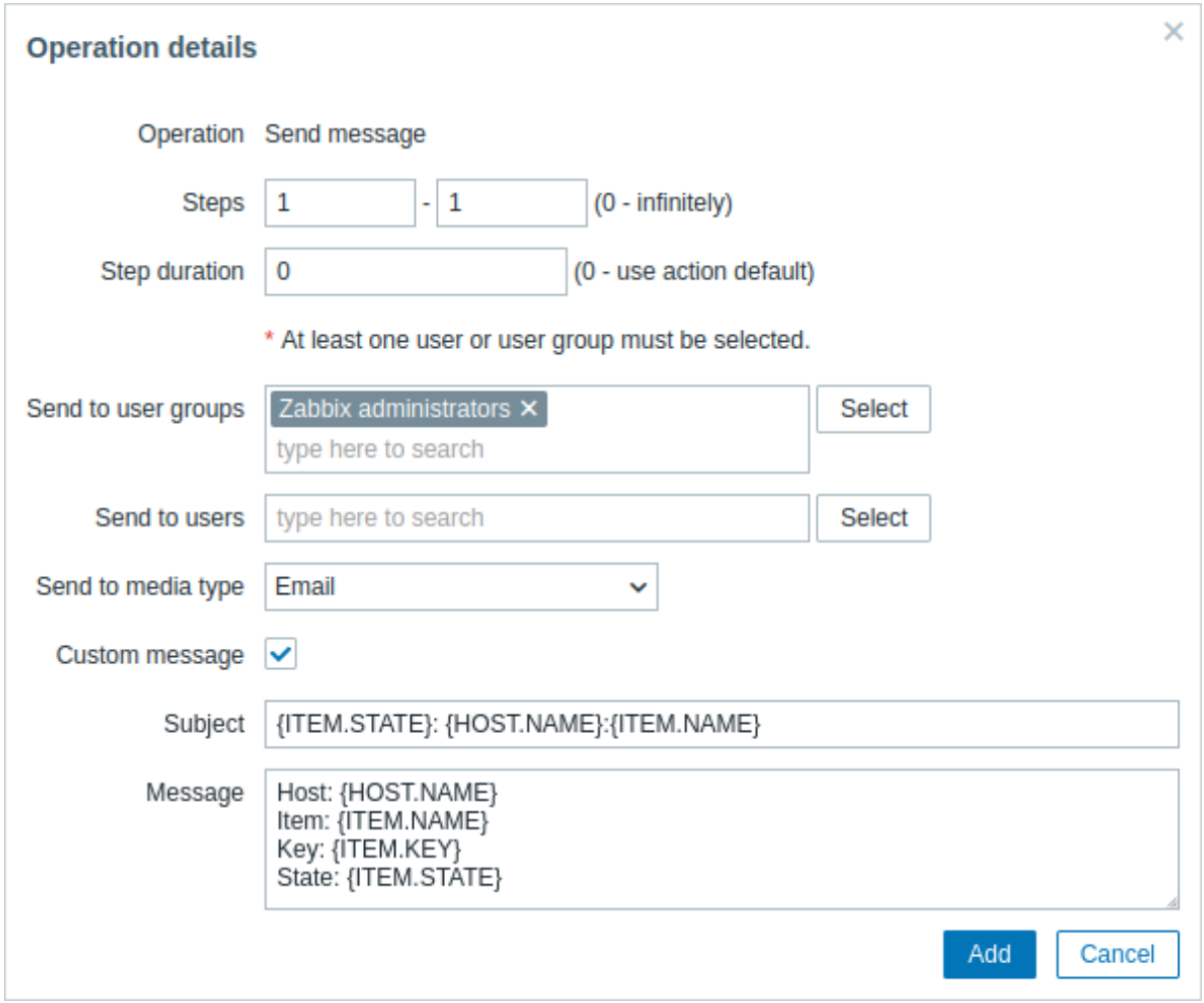
4. Cliquez sur *Ajouter* pour répertorier effectivement la condition dans le bloc *Conditions*.

Étape 4

1. Dans l’onglet *Opérations*, cliquez sur *Ajouter* dans le bloc *Opérations* pour ajouter une nouvelle opération.



2. Sélectionnez les destinataires du message (groupes d'utilisateurs/utilisateurs) ainsi que le type de média (ou « Tous ») à utiliser pour l'envoi.
3. Cochez la case *Message personnalisé* si vous souhaitez saisir l'objet et le contenu personnalisés du message concernant le problème.



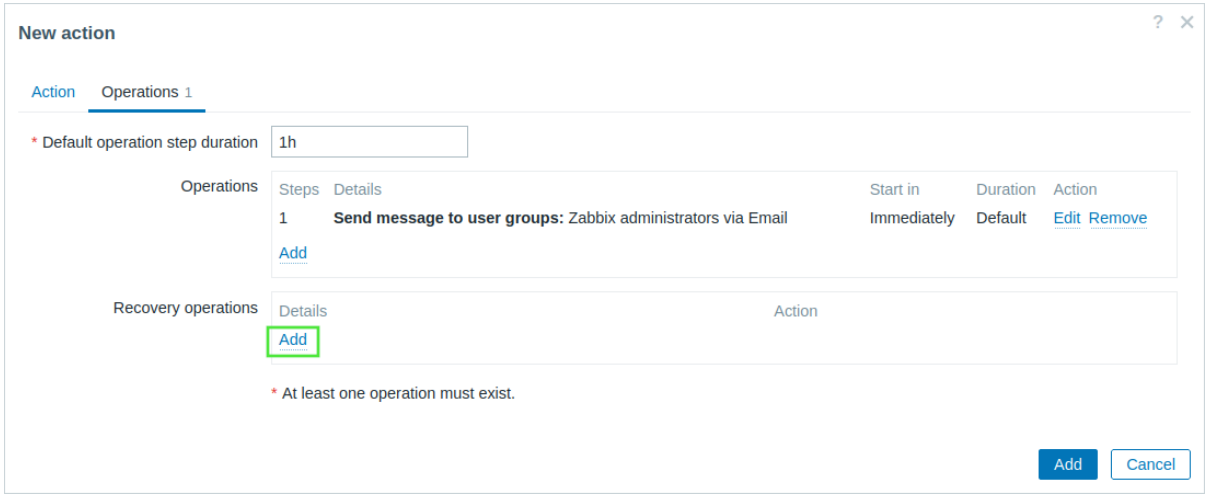
4. Cliquez sur *Ajouter* pour ajouter effectivement l'opération au bloc *Opérations*.

Si vous souhaitez recevoir plusieurs notifications, définissez la durée de l'étape de l'opération (intervalle entre les messages envoyés) et ajoutez une autre étape.

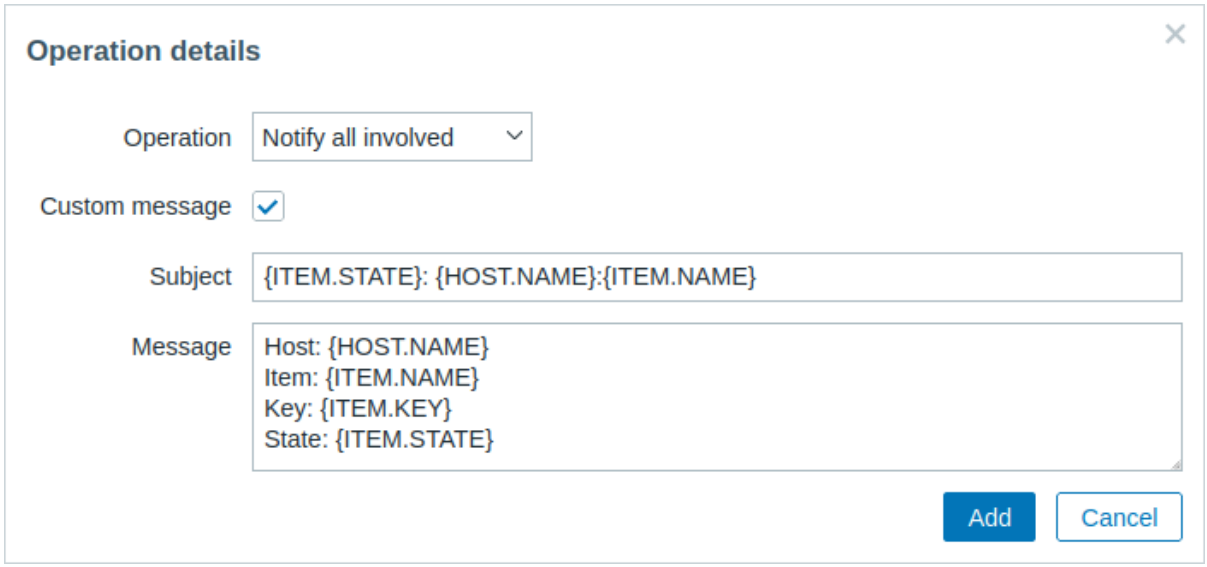
Étape 5

Le bloc *Opérations de récupération* vous permet de configurer une notification de récupération lorsqu'un élément revient à l'état normal.

1. Cliquez sur *Ajouter* dans le bloc *Opérations de récupération* pour ajouter une nouvelle opération de récupération.



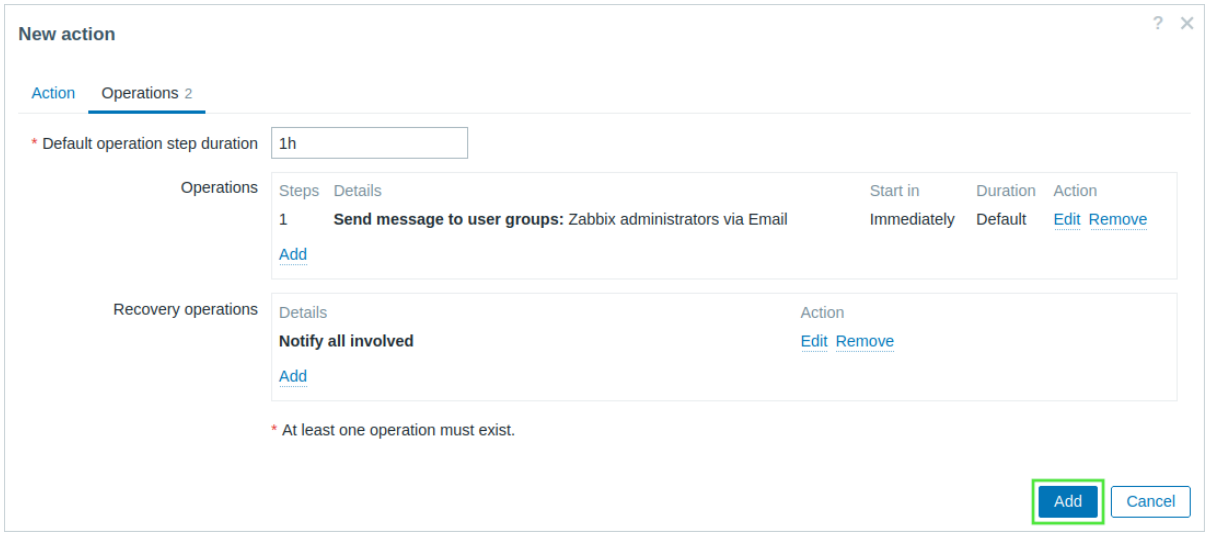
2. Sélectionnez le type d'opération « Notifier toutes les personnes concernées ».
3. Cochez la case *Message personnalisé* si vous souhaitez saisir l'objet et le contenu personnalisés du message de récupération.



4. Cliquez sur *Ajouter* dans la fenêtre contextuelle *Détails de l'opération* pour ajouter effectivement l'opération au bloc *Opérations de récupération*.

Étape 6

Une fois terminé, cliquez sur le bouton *Add* en bas du formulaire.



Et voilà, c'est terminé ! Vous pouvez maintenant vous attendre à recevoir votre première notification de Zabbix si un élément devient non pris en charge.

Macros

Vue d'ensemble

Zabbix prend en charge un certain nombre de macros intégrées qui peuvent être utilisées dans diverses situations. Ces macros sont des variables identifiées par une syntaxe spécifique :

`{MACRO}`

Les macros sont remplacées par une valeur spécifique en fonction du contexte.

L'utilisation efficace des macros vous permet de gagner du temps et de rendre la configuration de Zabbix plus transparente.

Dans l'un des cas d'utilisation courants, une macro peut être utilisée dans un modèle. Ainsi, un déclencheur sur un modèle peut être nommé « Processor load is too high on {HOST.NAME} ». Lorsque le modèle est appliqué au hôte, par exemple au serveur Zabbix, le nom devient « Processor load is too high on Zabbix server » lorsque le déclencheur est affiché dans la section Monitoring.

Les macros peuvent être utilisées dans les paramètres de clé d'élément. Une macro peut être utilisée pour une partie seulement du paramètre, par exemple `item.key[server_{HOST.HOST}_local]`. Il n'est pas nécessaire de mettre le paramètre entre guillemets doubles, car Zabbix gère les éventuels symboles spéciaux ambigus présents dans la macro résolue.

Il existe d'autres types de macros dans Zabbix.

Zabbix prend en charge les macros suivantes :

- `{MACRO}` - macro intégrée (voir la liste complète)
- `{<macro>.<func>(<params>)}` - fonctions de macro
- `{$MACRO}` - macro définie par l'utilisateur, éventuellement avec un contexte
- `{#MACRO}` - macro pour la découverte de bas niveau
- `{?EXPRESSION}` - macro d'expression

Fonctions de macro

Vue d'ensemble

Les fonctions de macro offrent la possibilité de personnaliser les valeurs des macro (par exemple, raccourcir ou extraire des sous-chaînes spécifiques), ce qui les rend plus faciles à utiliser.

La syntaxe d'une fonction de macro est la suivante :

```
{macro.func(params)}
```

où

- **macro** - la macro à personnaliser ;
- **func** - la fonction à appliquer (voir fonctions prises en charge) ;
- **params** - une liste de paramètres de fonction séparés par des virgules, qui doivent être **entre guillemets doubles** si :
  - ils commencent par un espace ou par des guillemets doubles ;
  - ils contiennent des parenthèses fermantes ou une virgule.

Par exemple :

```
{{TIME}.fmttime(format,time_shift)}
{{ITEM.VALUE}.regsub(pattern, output)}
{{$USERMACRO}.regsub(pattern, output)}
{{#LLDMACRO}.regsub(pattern, output)}
```

Les fonctions de macro sont prises en charge pour

- les macros intégrées
- les macros utilisateur
- les macros de découverte bas niveau
- les macros d'expression

Les fonctions de macro peuvent être utilisées dans tous les emplacements prenant en charge les macros répertoriées. Cela s'applique sauf indication explicite selon laquelle seule une macro est attendue (par exemple, lors de la configuration des macros d'hôte ou des filtres d'une règle de découverte bas niveau).

Une seule fonction par macro est prise en charge ; les chaînes de plusieurs fonctions de macro ne sont pas prises en charge.

Note:
Veuillez consulter les exemples d'échappement pour les cas où les fonctions de macro sont utilisées dans d'autres contextes (fonction, clé d'élément, autre macro, etc).

Fonctions prises en charge

Les fonctions sont répertoriées sans informations supplémentaires. Cliquez sur une fonction pour afficher tous les détails.

Function	Description
btoa	Encodage de la valeur de macro en Base64.
fmtnum	Formatage des nombres pour contrôler le nombre de chiffres affichés après la virgule décimale.
fmttime	Formatage de l'heure.
htmldecode	Décodage de la valeur de macro depuis l'encodage HTML.
htmlencode	Encodage de la valeur de macro en HTML.
iregsub	Extraction d'une sous-chaîne par correspondance avec une expression régulière (insensible à la casse).
jsonpath	Renvoie le résultat JSONPath.
lowercase	Conversion des caractères de la valeur de macro en minuscules.
regrepl	Remplacement d'un caractère ou d'une sous-chaîne dans la valeur de macro.
regsub	Extraction d'une sous-chaîne par correspondance avec une expression régulière (sensible à la casse).
tr	Translittération des caractères de la valeur de macro.
uppercase	Conversion des caractères de la valeur de macro en majuscules.
urldecode	Décodage de la valeur de macro depuis l'encodage URL.
urlencode	Encodage de la valeur de macro en URL.
xmlxpath	Renvoie le résultat XML XPath.

Détails de la fonction

Les paramètres de fonction facultatifs sont indiqués par < >.

btoa

Encodage d'une valeur de macro en Base64. L'encodage Base64 est une méthode permettant de représenter des données binaires sous forme de texte, utile pour le stockage et la transmission sécurisée de contenu binaire via des protocoles basés sur du texte.

Exemple :

```
{{ITEM.VALUE}.btoa()} - encodera en Base64 une valeur comme "zabbix" en "emFiYml4"
```

fmtnum(digits)

Formatage des nombres pour contrôler le nombre de chiffres affichés après la virgule.

Paramètres :

- **digits** - le nombre de chiffres après la virgule. Plage valide : 0-20. Aucun zéro final ne sera produit.

Exemples :

```
{{ITEM.VALUE}.fmtnum(2)} - retournera "24.35" à partir d'une valeur reçue de "24.3483523"
{{ITEM.VALUE}.fmtnum(0)} - retournera "24" à partir d'une valeur reçue de "24.3483523"
```

fmttime(format,<time_shift>)

Formatage de l'heure.
 Notez que cette fonction peut être utilisée avec des macros qui se résolvent en une valeur dans l'un des formats d'heure suivants :

- `hh:mm:ss`
- `yyyy-mm-ddThh:mm:ss[tz]` (norme ISO8601)
- horodatage UNIX

Paramètres :

- **format** - chaîne de format obligatoire, compatible avec le formatage de la fonction `strftime` ;
- **time_shift** (facultatif) - décalage temporel appliqué à l'heure avant le formatage ; doit commencer par `-<N><time_unit>` ou `+<N><time_unit>`, où :
  - `N` - le nombre d'unités de temps à ajouter ou à soustraire ;
  - `time_unit` - h (heure), d (jour), w (semaine), M (mois) ou y (année).

Commentaires :

- Le paramètre `time_shift` prend en charge les opérations temporelles en plusieurs étapes et peut inclure `/<time_unit>` pour ramener au début de l'unité de temps (`/d` - minuit, `/w` - 1er jour de la semaine (lundi), `/M` - 1er jour du mois, etc.). Exemples : `-1w` - exactement 7 jours en arrière ; `-1w/w` - lundi de la semaine précédente ; `-1w/w+1d` - mardi de la semaine précédente.
- Les opérations temporelles sont calculées de gauche à droite, sans priorités. Par exemple, `-1M/d+1h/w` sera interprété comme `((-1M/d)+1h)/w`.

Exemples :

```
{{TIMESTAMP}.fmttime(%B)} - renverra "October" à partir d'une valeur reçue de "1633098961"
{{TIMESTAMP}.fmttime(%d %B,-1M/M)} - renverra "1 September" à partir d'une valeur reçue de "1633098961"
{{EVENT.TIMESTAMP}.fmttime("%x %X")} - renverra la représentation locale de la date et de l'heure ; par ex
{{EVENT.TIME}.fmttime("%x %X")} - renverra la représentation locale de la date et de l'heure ; par exemple
```

htmldecode

Décodage d'une valeur de macro à partir de l'encodage HTML.

Les caractères suivants sont pris en charge :

Valeur	Valeur décodée
&	&
<	<
>	>
"	"
'	'
'	'

Exemple :

```
{{ITEM.VALUE}.htmldecode()} - décodera en HTML une valeur telle que "&lt;" en "<"
```

htmlencode

Encodage d'une valeur de macro au format HTML.

Les caractères suivants sont pris en charge :

Valeur	Valeur encodée
&	&
<	<
>	>
"	"
'	'

Exemple :

```
{{ITEM.VALUE}.htmlencode()} - encodera en HTML un caractère comme "<" en "&lt;"
```

iregsub(pattern,output)

Extraction d'une sous-chaîne par correspondance avec une expression régulière (insensible à la casse).

Paramètres :

- **pattern** - l'expression régulière à faire correspondre ;
- **output** - les options de sortie. Les espaces réservés **\1 - \9** sont pris en charge pour les groupes de capture. **\0** renvoie le texte correspondant.

Commentaires :

- S'il n'y a aucune correspondance pour l'expression régulière, la fonction renvoie une chaîne vide.
- Si le modèle de la fonction est une expression régulière incorrecte, alors la macro est évaluée à 'UNKNOWN' (sauf pour les macros de découverte de bas niveau, auquel cas la fonction sera ignorée et la macro restera non résolue).
- Les références à des groupes de capture inexistants dans la chaîne de remplacement sont remplacées par une chaîne vide.

Exemple :

```
{{ITEM.VALUE}.iregsub("fail|error|fault|problem","ERROR")} - sera résolu en "ERROR" si les sous-chaînes "f
```

jsonpath(pattern,<default>)

Renvoie le résultat JSONPath.
 Types de valeurs pris en charge : *String*, *Text*, *Log*.

Paramètres :

- **pattern** - le modèle requis ;
- **default** - la valeur de secours facultative si la requête JSONPath ne renvoie aucune donnée. Notez qu'un échec est renvoyé en cas d'autres erreurs.

Commentaires :

- La fonction échoue avec une erreur si :
  - Le JSON n'est pas valide.
  - La requête JSONPath échoue.
  - Le résultat JSONPath ne contient aucune donnée et le paramètre `default` n'est pas fourni.

Exemple :

```
{{ITEM.VALUE}.jsonpath("$.columns.servicename", "NA")}
```

minuscules

Transformation de tous les caractères de la valeur de la macro en minuscules.
Fonctionne avec des jeux de caractères à un seul octet (tels que l'ASCII) et ne prend pas en charge l'UTF-8.

Exemple :

```
{{ITEM.VALUE}.lowercase()} - transformera une valeur telle que "Zabbix SERVER" en "zabbix server" (minuscu
```

regrepl(pattern,replacement,<pattern2>,<replacement2>,...)

Remplacement d'un caractère ou d'une sous-chaîne dans la valeur d'un macro.

Paramètres :

- **pattern** - l'expression régulière à faire correspondre ;
- **replacement** - la chaîne de remplacement. Les espaces réservés **\1 - \9** sont pris en charge dans les chaînes de remplacement pour capturer des groupes.

Commentaires :

- Toutes les occurrences correspondant au pattern sont remplacées.
- Les patterns et les remplacements sont traités séquentiellement, chaque paire suivante étant appliquée conformément au résultat du remplacement précédent.
- Les références à des groupes de capture inexistants dans la chaîne de remplacement sont remplacées par une chaîne vide.

Exemples :

```
{{ITEM.VALUE}.regrepl("oldParam", "newParam")} - remplacera "oldParam" par "newParam"
{{ITEM.VALUE}.regrepl("([^a-z])","\\\1")} - tous les caractères non alphabétiques seront échappés avec une
{$THRESHOLD:"{{#FSNAME}.regrepl(\"\\$\",\"\")}"} - supprimera une barre oblique inverse finale (par exempl
{{ITEM.VALUE}.regrepl("_v1\.0", "_v2.0", "\(final\)", "")} - remplacera plusieurs parties de la valeur de
```

regsub(pattern,output)

Extraction de sous-chaîne par correspondance avec une expression régulière (sensible à la casse).

Paramètres :

- **pattern** - l'expression régulière à faire correspondre ;
- **output** - les options de sortie. Les espaces réservés **\1 - \9** sont pris en charge pour les groupes de capture. **\0** renvoie le texte correspondant.

Commentaires :

- S'il n'y a aucune correspondance pour l'expression régulière, la fonction renvoie une chaîne vide.
- Si le modèle de la fonction est une expression régulière incorrecte, alors la macro est évaluée à 'UNKNOWN' (sauf pour les macros de découverte de bas niveau, auquel cas la fonction sera ignorée et la macro restera non résolue).
- Les références à des groupes de capture inexistants dans la chaîne de remplacement sont remplacées par une chaîne vide.

Exemples :

```
{{ITEM.VALUE}.regsub("^([0-9]+)", Problem ID: \1)} - sera résolu en "Problem ID: 123" si une valeur telle
{{ITEM.VALUE}.regsub("fail|error|fault|problem","ERROR")} - sera résolu en "ERROR" si les sous-chaînes "fa
```

Voir plus d'exemples.

tr(caractères,remplacement)

Translittération des caractères de la valeur de macro.

- **caractères** - l'ensemble des caractères à remplacer ;
- **remplacement** - l'ensemble des caractères de remplacement correspondants par position.

Exemples :

```
{{ITEM.VALUE}.tr(abc, xyz)} - remplacera toutes les occurrences de "a" par "x", de "b" par "y", de "c" par
{{ITEM.VALUE}.tr(abc, xyzq)} - remplacera toutes les occurrences de "a" par "x", de "b" par "y", de "c" pa
{{ITEM.VALUE}.tr(abcde, xyz)} - remplacera toutes les occurrences de "a" par "x", de "b" par "y", de "c" p
{{ITEM.VALUE}.tr("\\\'", "\/\"")} - remplacera toutes les occurrences de barre oblique inverse par barre o
{{ITEM.VALUE}.tr(A-Z,a-z)} - convertira toutes les lettres en minuscules
{{ITEM.VALUE}.tr(0-9a-z,*)} - remplacera tous les chiffres et toutes les lettres minuscules par "*"
{{ITEM.VALUE}.tr(0-9,ab)} - remplacera toutes les occurrences de 0 par "a", et toutes les occurrences de 1
{{ITEM.VALUE}.tr(0-9abcA-L,*)} - remplacera tous les chiffres, les caractères "abc" et la plage A-L par "*
{{ITEM.VALUE}.tr("\n","*")} - remplacera les occurrences de fin de ligne par *
{{ITEM.VALUE}.tr("e", "\n")} - remplacera tous les "e" par une fin de ligne
```

Pour inclure des caractères littéraux :

```
barre oblique inverse - doit être échappée comme \\
guillemet simple - doit être échappé comme \'
guillemet double - doit être échappé comme \"
```

Séquences d'échappement prises en charge avec la barre oblique inverse :

```
\\\\ => \\ - double barre oblique inverse vers barre oblique inverse simple
\\a  => \a - alerte
\\b  => \b - retour arrière
\\f  => \f - saut de page
\\n  => \n - nouvelle ligne
\\r  => \r - retour chariot
\\t  => \t - tabulation horizontale
\\v  => \v - tabulation verticale
```

uppercase

Transformation de tous les caractères de la valeur de la macro en majuscules.
Fonctionne avec des jeux de caractères à un seul octet (tels que l'ASCII) et ne prend pas en charge l'UTF-8.

Exemple :

```
{{ITEM.VALUE}.uppercase()} - transformera une valeur telle que "Zabbix Server" en "ZABBIX SERVER" (majuscu
```

urldecode

Décodage d'une valeur de macro à partir d'un encodage URL.

Exemple :

```
{{ITEM.VALUE}.urldecode()} - décodera une valeur URL-encodée telle que "%2F" en "/"
```

urlencode

Encodage d'une valeur de macro au format URL.

Exemple :

```
{{ITEM.VALUE}.urlencode()} - encodera en URL un caractère comme "/" en "%2F"
```

xmlxpath(pattern,<default>)

Renvoie le résultat XPath XML.
 Types de valeurs pris en charge : *String*, *Text*, *Log*.

Paramètres :

- **pattern** - le modèle requis ;
- **default** - la valeur de secours facultative si la requête XPath ne renvoie aucune donnée. Notez qu'un échec est renvoyé en cas d'autres erreurs.

Commentaires :

- La fonction échoue avec une erreur si :
  - Le XML n'est pas valide.
  - La requête XPath échoue.
  - Le résultat XPath ne contient aucune donnée et le paramètre `default` n'est pas fourni.
  - Zabbix a été compilé sans prise en charge de XML.

Exemple :

```
{{ITEM.VALUE}.xmlxpath("/response/error/status","unknown")}
```

Exemples supplémentaires

Le tableau ci-dessous présente d'autres exemples d'utilisation des fonctions de macro.

`{#IFALIAS}` est une macro LLD et n'est définie que dans les contextes de découverte de bas niveau (règles de découverte, prototypes et éléments/déclencheurs créés à partir de ceux-ci). Son utilisation en dehors de la LLD laissera le jeton non développé.

Fonction de macro	Valeur reçue	Sortie
`{{ITEM.VALUE}.regsub(^[0-9]+, Problem)}`	123Log line	Problem
`{{ITEM.VALUE}.regsub("^([0-9]+)", "Problem")}`	123 Log line	Problem
`{{ITEM.VALUE}.regsub(".*", "Problem ID: \1")}`	Log line	Problem ID:
`{{ITEM.VALUE}.regsub("^(\w+).*?([0-9]+)", " Problem ID: \1_\2 ")}`	MySQL crashed errno 123	Problem ID: MySQL_123
`{{ITEM.VALUE}.regsub("([1-9]+", "Problem ID: \1")}`	123 Log line	*UNKNOWN* (expression régulière invalide)
`{{#IFALIAS}.regsub("(.*)_([0-9]+)", \1)}`	customername_1	customername
`{{#IFALIAS}.regsub("(.*)_([0-9]+)", \2)}`	customername_11	
`{{#IFALIAS}.regsub("(.*)_([0-9]+", \1)}`	customername_1	{{#IFALIAS}.regsub("(.*)_([0-9]+", \1)} (expression régulière invalide)
`{$MACRO:"{{#IFALIAS}.regsub(\"(.*)_([0-9]+\", \1)}"}`	customername_1	{$MACRO:"customername"}
`{$MACRO:"{{#IFALIAS}.regsub(\"(.*)_([0-9]+\", \2)}"}`	customername_1	{$MACRO:"1"}
`{$MACRO:"{{#IFALIAS}.regsub(\"(.*)_([0-9]+\", \1)}"}`	customername_1	{$MACRO:"{{#IFALIAS}.regsub(\"(.*)_([0-9]+\", \1)}"} (expression régulière invalide)
`"{$MACRO:"\{{#IFALIAS}.regsub("(.*)_([0-9]+", \1)}\"}"`	customername_1	"{$MACRO:"\customername\"}"
`"{$MACRO:"\{{#IFALIAS}.regsub("(.*)_([0-9]+", \2)}\"}"`	customername_1	"{$MACRO:"\1\"}"
`"{$MACRO:\"{{#IFALIAS}.regsub(\\"(.*)_([0-9]+", \1)}\"}"`	customername_1	"{$MACRO:\"{{#IFALIAS}.regsub(\\"(.*)_([0-9]+\\", \1)}\"}" (expression régulière invalide)

Affichage des valeurs complètes des éléments

Les valeurs longues des macros résolues {ITEM.VALUE} et {ITEM.LASTVALUE} pour les éléments de type texte/log sont tronquées à 20 caractères dans certains emplacements du frontend. Pour voir les valeurs complètes de ces macros, vous pouvez utiliser des fonctions de macro, par exemple :

```
{{ITEM.VALUE}.regsub("(.*)", \1)}
{{ITEM.LASTVALUE}.regsub("(.*)", \1)}
```

Voir aussi : détails des macros {ITEM.VALUE} et {ITEM.LASTVALUE} macro details.

Macros utilisateur

Présentation générale

Les macros utilisateur sont prises en charge dans Zabbix pour offrir davantage de flexibilité, en plus des macros prises en charge par défaut.

Les macros utilisateur peuvent être définies au niveau global, du modèle et de l'hôte. Ces macros utilisent une syntaxe particulière :

```
{$MACRO}
```

Zabbix résout les macros selon l'ordre de priorité suivant :

1. Macros définies au niveau de l'hôte (vérifiées en premier).
2. Macros définies pour les modèles de premier niveau de l'hôte (c'est-à-dire les modèles liés directement à l'hôte), triées par ID de modèle.
3. Macros définies pour les modèles de deuxième niveau de l'hôte, triées par ID de modèle.
4. Macros définies pour les modèles de troisième niveau de l'hôte, triées par ID de modèle, etc.
5. Macros globales (vérifiées en dernier).

En d'autres termes, si une macro n'existe pas pour un hôte, Zabbix essaiera de la trouver dans les modèles de l'hôte, en augmentant progressivement la profondeur. Si elle n'est toujours pas trouvée, une macro globale sera utilisée, si elle existe.

Warning:
Si une macro portant le **même nom** existe dans plusieurs modèles liés du même niveau, la macro du modèle ayant l'ID le plus bas sera utilisée. La présence de macros portant le même nom dans plusieurs modèles constitue donc un risque de configuration.

Si Zabbix ne parvient pas à trouver une macro, celle-ci ne sera pas résolue.

Attention:
Les macros (y compris les macros utilisateur) restent non résolues dans la section Configuration (par exemple, dans la liste des déclencheurs), afin de rendre les configurations complexes plus transparentes.

Les macros utilisateur peuvent être utilisées dans :

- le nom de l'élément
- le paramètre de clé de l'élément
- les intervalles de mise à jour et les intervalles flexibles de l'élément
- le nom et la description du déclencheur
- les paramètres et constantes de l'expression du déclencheur (voir les exemples)
- de nombreux autres emplacements - consultez la liste complète

Cas d'utilisation courants des macros globales et hôtes

- utiliser une macro globale à plusieurs endroits ; puis modifiez la valeur de la macro et appliquez les modifications de configuration à tous les emplacements en un seul clic
- tirez parti de modèles avec des attributs spécifiques à l'hôte : mots de passe, numéros de port, noms de fichiers, expressions régulières, etc.

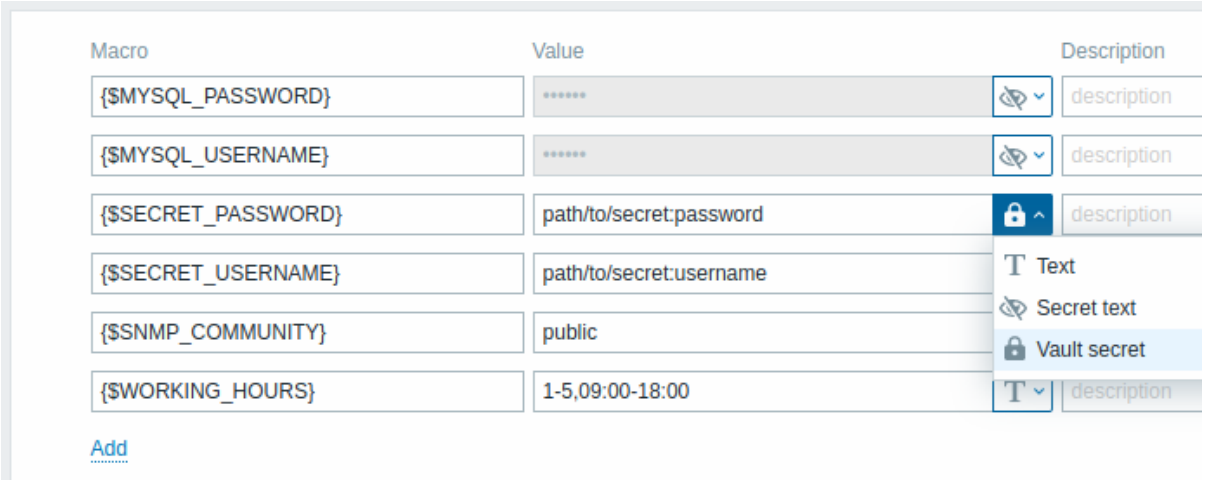
Note:
Il est recommandé d'utiliser des macros d'hôte plutôt que des macros globales, car l'ajout, la mise à jour ou la suppression de macros globales impose une mise à jour incrémentielle de la configuration pour tous les hôtes. Pour plus d'informations, consultez Passive and active agent checks.

Configuration

Pour définir des macros utilisateur, accédez à l'emplacement correspondant dans l'interface:

- Pour les macros globales, allez dans *Administration* > *Macros*.
- Pour les macros au niveau de l'hôte et du modèle, ouvrez les propriétés de l'hôte ou du modèle et recherchez l'onglet *Macros*.

Une macro utilisateur possède les attributs suivants:



Parameter	Description
Macro	Nom de la macro. Le nom doit être entouré d'accolades et commencer par un signe dollar. Exemple: `{$FRONTEND_URL}`. Les caractères suivants sont autorisés dans les noms de macro: **A-Z** (majuscules uniquement), **0-9**, _, **.**
Value	Valeur de la macro. Trois types de valeur sont pris en charge: **Text** (par défaut) - valeur en texte brut **Secret text** - la valeur est masquée par des astérisques **Vault secret** - la valeur contient un chemin/requête vers un vault secret. Pour modifier le type de valeur, cliquez sur le bouton à la fin du champ de saisie de la valeur. La longueur maximale d'une valeur de macro utilisateur est de 2048 caractères.
Description	Champ texte utilisé pour fournir plus d'informations sur cette macro.

Lors de la configuration des macros utilisateur, **notez** les comportements contextuels suivants:

- si des macros utilisateur sont utilisées dans des éléments ou des déclencheurs de modèle, envisagez d'ajouter également ces macros au modèle (même si elles sont définies globalement); de cette façon, les macros de type *Text* fonctionneront comme prévu après l'exportation du modèle au format XML et son importation dans un autre système (les valeurs des macros secrètes ne sont pas exportées)
- si des macros utilisateur sont utilisées dans des expressions de déclencheur, ces macros ne seront résolues que si elles font référence à un paramètre ou à une constante; elles ne seront PAS résolues si elles font référence à un hôte, à une clé d'élément, à une fonction, à un opérateur ou à une autre expression de déclencheur (les macros secrètes ne peuvent pas être utilisées dans les expressions de déclencheur)
- si des macros utilisateur sont utilisées sur un hôte qui possède une règle de découverte de bas niveau avec des prototypes d'hôte, les hôtes découverts hériteront de toutes les macros utilisateur définies sur cet hôte

Exemples

Exemple 1

Utilisation d'une macro au niveau de l'hôte dans la clé de l'élément « État du démon SSH » :

`net.tcp.service[ssh,,{$SSH_PORT}]`

Cet élément peut être attribué à plusieurs hôtes, à condition que la valeur de `{$SSH_PORT}` soit définie sur ces hôtes.

Exemple 2

Utilisation d'une macro au niveau du hôte dans le déclencheur « La charge du processeur est trop élevée » :

`last(/ca_001/system.cpu.load[,avg1])>{$MAX_CPULOAD}`

Un tel déclencheur serait créé sur le modèle, et non modifié sur chaque hôte.

> **Note:**
> Si vous souhaitez utiliser le nombre de valeurs comme paramètre de la fonction (par exemple, `max(/host/key,#3)`), incluez le symbole dièse dans la définition de la macro comme suit : `SOME_PERIOD => #3`.

Exemple 3

Utilisation de deux macros dans le déclencheur « La charge CPU est trop élevée » :

```
min(/ca_001/system.cpu.load[,avg1],{$CPULOAD_PERIOD})>{$MAX_CPULOAD}
```

Notez qu'une macro peut être utilisée comme paramètre d'une fonction de déclencheur, dans cet exemple la fonction `min()`.

Exemple 4

Synchroniser la condition d'indisponibilité de l'agent avec l'intervalle de mise à jour de l'élément :

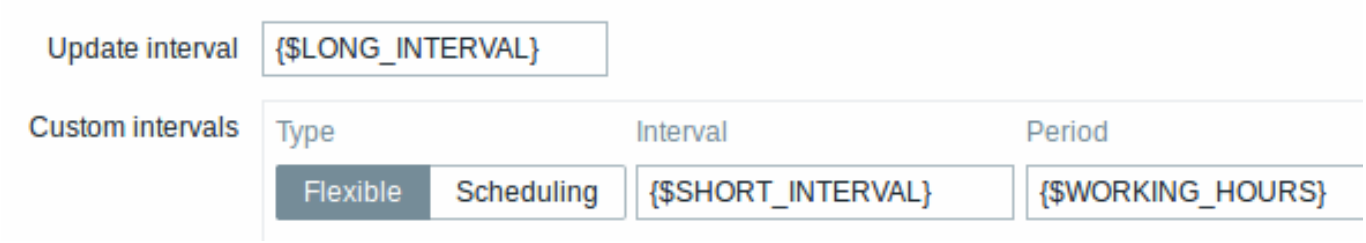
- définir la macro `{$INTERVAL}` et l'utiliser dans l'intervalle de mise à jour de l'élément ;
- utiliser `{$INTERVAL}` comme paramètre du déclencheur d'indisponibilité de l'agent :

```
nodata(/ca_001/agent.ping,{$INTERVAL})=1
```

Exemple 5

Centraliser la configuration des heures de travail :

- créer une macro globale `{$WORKING_HOURS}` égale à `1-5,09:00-18:00` ;
- l'utiliser dans le champ *Heures de travail* de *Administration* > *Général* > *Interface* ;
- l'utiliser dans le champ *Quand actif* de l'onglet *Média* d'un utilisateur, dans *Utilisateurs* > *Utilisateurs* ;
- l'utiliser pour configurer une collecte plus fréquente des éléments pendant les heures de travail :

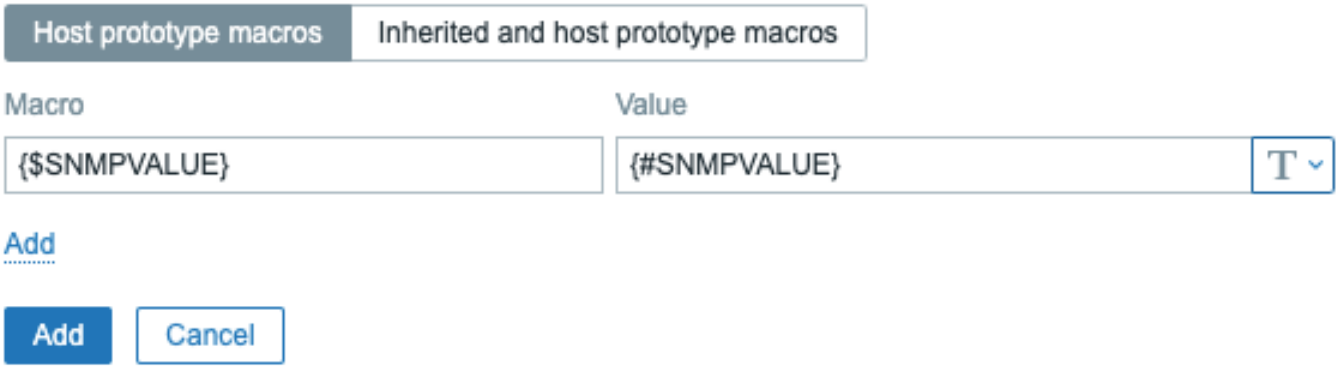


- l'utiliser dans la condition d'action *Période de temps* ;
- ajuster les heures de travail dans *Administration* > *Macros*, si nécessaire.

Exemple 6

Utilisez une macro de prototype de hôte pour configurer des éléments pour les hôtes découverts :

1. Dans un prototype de hôte, définissez la macro utilisateur `{$SNMPVALUE}` avec la macro de découverte de bas niveau `{#SNMPVALUE}` comme valeur :



2. Assignez le modèle *Generic SNMPv2* au prototype de hôte.
3. Utilisez `{$SNMPVALUE}` dans le champ *OID SNMP* des éléments du modèle *Generic SNMPv2*.

Contexte des macros utilisateur

Voir macros utilisateur avec contexte.

Macros utilisateur avec contexte

Vue d'ensemble

Un contexte facultatif peut être utilisé dans les macros utilisateur, ce qui permet de remplacer la valeur par défaut par une valeur spécifique au contexte.

Le contexte est ajouté au nom de la macro ; la syntaxe dépend du fait que le contexte soit une valeur textuelle statique :

```
{$MACRO:"texte statique"}
```

ou une expression régulière :

```
{$MACRO:regex:"expression régulière"}
```

Notez qu'une macro avec un contexte d'expression régulière peut uniquement être définie dans la configuration des macros utilisateur. Si le préfixe `regex:` est utilisé ailleurs comme contexte de macro utilisateur, par exemple dans une expression de déclencheur, il sera traité comme un contexte statique.

Les guillemets du contexte sont facultatifs (voir également les remarques importantes).

Exemples de contextes de macro :

Exemple	Description
`{$LOW_SPACE_LIMIT}`	Macro utilisateur sans contexte.
`{$LOW_SPACE_LIMIT:/tmp}`	Macro utilisateur avec contexte (chaîne statique).
`{$LOW_SPACE_LIMIT:regex:"^/tmp$"}`	Macro utilisateur avec contexte (expression régulière). Équivalent à `{$LOW_SPACE_LIMIT:/tmp}`.
`{$LOW_SPACE_LIMIT:regex:"^/var/log/.*$"}`	Macro utilisateur avec contexte (expression régulière). Correspond à toutes les chaînes commençant par /var/log/.

Cas d'utilisation

Les macros utilisateur avec contexte peuvent être définies pour obtenir des seuils plus flexibles dans les expressions de déclencheur (basées sur les valeurs récupérées par la découverte de bas niveau). Par exemple, vous pouvez définir les macros suivantes :

- {$LOW_SPACE_LIMIT} = 10
- {$LOW_SPACE_LIMIT:/home} = 20
- {$LOW_SPACE_LIMIT:regex:"^/[a-z]+$"} = 30

Ensuite, une macro de découverte de bas niveau peut être utilisée comme contexte de macro dans un prototype de déclencheur pour la découverte des systèmes de fichiers montés :

```
last(/host/vfs.fs.size[{#FSNAME},pfree])<{$LOW_SPACE_LIMIT:"{#FSNAME}"}
```

Après la découverte, différents seuils d'espace disque faible s'appliqueront dans les déclencheurs en fonction des points de montage ou des types de systèmes de fichiers découverts. Des événements de problème seront générés si :

- le dossier /home dispose de moins de 20 % d'espace disque libre
- les dossiers qui correspondent au motif regexp (comme /etc, /tmp ou /var) disposent de moins de 30 % d'espace disque libre
- les dossiers qui ne correspondent pas au motif regexp et qui ne sont pas /home disposent de moins de 10 % d'espace disque libre

Notes importantes

- S'il existe plusieurs macros utilisateur avec contexte, Zabbix essaiera d'abord de faire correspondre les macros de contexte simples, puis les macros de contexte avec des expressions régulières dans un ordre indéfini.

Warning:
Ne créez pas de macros de contexte différentes correspondant à la même chaîne pour éviter un comportement indéfini.

- Si une macro avec son contexte n'est pas trouvée sur l'hôte, sur les modèles liés ou globalement, la macro sans contexte est recherchée.
- Seules les macros de découverte de bas niveau sont prises en charge dans le contexte. Toutes les autres macros sont ignorées et traitées comme du texte brut.

Techniquement, le contexte de la macro est spécifié à l'aide de règles similaires aux paramètres de la clé d'élément, sauf que le contexte de la macro n'est pas analysé comme plusieurs paramètres s'il y a un caractère `,` :

- Le contexte de la macro doit être entre guillemets avec " si le contexte contient un caractère } ou commence par un caractère ". Les guillemets à l'intérieur du contexte entre guillemets doivent être échappés avec le caractère \.
- Le caractère \ lui-même n'est pas échappé, ce qui signifie qu'il est impossible d'avoir un contexte entre guillemets se terminant par le caractère \ - la macro {$MACRO:"a:\b\c\"} est invalide.
- Les espaces de début dans le contexte sont ignorés, les espaces de fin ne le sont pas :
  - Par exemple {$MACRO:A} est identique à {$MACRO: A}, mais pas à {$MACRO:A }.
- Tous les espaces avant les guillemets de début et après les guillemets de fin sont ignorés, mais tous les espaces à l'intérieur des guillemets ne le sont pas :
  - Les macros {$MACRO:"A"}, {$MACRO: "A"}, {$MACRO:"A" } et {$MACRO: "A" } sont identiques, mais les macros {$MACRO:"A"} et {$MACRO:" A "} ne le sont pas.

Les macros suivantes sont toutes équivalentes, car elles ont le même contexte : {$MACRO:A}, {$MACRO:A} et {$MACRO:"A"}. Cela contraste avec les clés d'élément, où 'key[a]', 'key[a]' et 'key["a"]' sont sémantiquement identiques, mais différents à des fins d'unicité.

Macros utilisateur secrètes

Vue d'ensemble

Zabbix propose deux options pour protéger les informations sensibles dans les valeurs des macros utilisateur :

- Texte secret
- Secret du coffre-fort

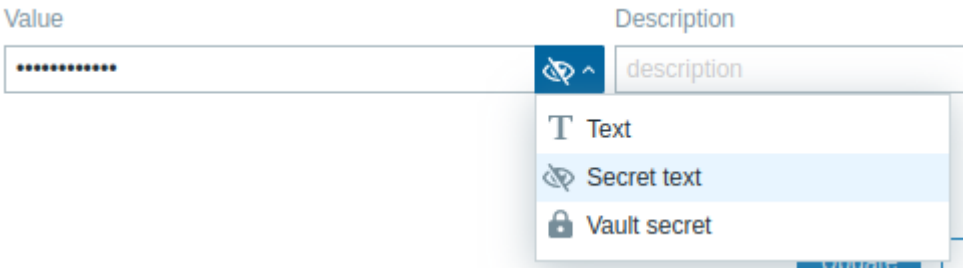
Note:
Bien que la valeur d'une macro secrète soit masquée, elle peut être révélée par son utilisation dans des éléments. Par exemple, dans un script externe, une instruction echo faisant référence à une macro secrète peut être utilisée pour révéler la valeur de la macro dans l'interface, car le serveur Zabbix a accès à la valeur réelle de la macro. Consultez les emplacements où les valeurs des macros secrètes sont démasquées.

Les macros secrètes ne peuvent pas être utilisées dans les expressions de déclencheur.

Texte secret

Avec les macros de texte secret, la valeur de la macro est masquée par des astérisques.

Pour rendre une valeur de macro secrète, cliquez sur le bouton à la fin du champ *Value* et sélectionnez l'option *Secret text* :



Une fois la configuration enregistrée, il ne sera plus possible d'afficher la valeur.

Pour modifier la valeur de la macro, survolez le champ *Value* et cliquez sur le bouton *Set new value* (qui apparaît au survol) :



Lorsque vous cliquez sur le bouton *Set new value* (ou que vous modifiez le type de valeur de la macro), la valeur actuelle sera effacée. Vous pouvez restaurer la valeur d'origine en cliquant sur la flèche à la fin du champ *Value* (disponible uniquement avant l'enregistrement de la nouvelle configuration). Notez que la restauration de la valeur d'origine ne l'exposera pas.

Note:
Les URL qui contiennent une macro secrète ne fonctionneront pas, car la macro qu'elles contiennent sera résolue en "******".

Secret Vault

Avec les macros de secret Vault, la valeur de la macro est stockée dans un logiciel externe de gestion des secrets (vault).

Pour configurer une macro de secret Vault, cliquez sur le bouton à la fin du champ *Value* et sélectionnez l'option *Vault secret* :

La valeur de la macro doit pointer vers un secret du vault. Le format d'entrée dépend du fournisseur de vault. Pour des exemples de configuration spécifiques à chaque fournisseur, voir :

- HashiCorp
- CyberArk

Les valeurs des macros secrètes Vault sont récupérées depuis le coffre-fort par le serveur Zabbix (et le proxy Zabbix, si *Resolve secret vault macros by* est défini sur *Zabbix server and proxy*) à chaque actualisation des données de configuration, puis stockées dans le cache de configuration. Le serveur Zabbix et le proxy Zabbix peuvent utiliser des coffres-forts différents.

Si *Resolve secret vault macros by* est défini sur *Zabbix server*, alors les secrets du coffre-fort sont récupérés uniquement par le serveur et le proxy Zabbix reçoit les valeurs des macros secrètes Vault depuis le serveur Zabbix à chaque synchronisation de configuration, puis les stocke dans son propre cache de configuration. Cela signifie qu'un proxy Zabbix ne peut pas démarrer la collecte de données après un redémarrage tant qu'il n'a pas reçu la mise à jour de configuration du serveur Zabbix.

Pour actualiser manuellement les valeurs secrètes depuis le coffre-fort, utilisez l'option `secrets_reload` de contrôle d'exécution (serveur uniquement).

Le chiffrement doit être activé entre le serveur Zabbix et le proxy ; sinon, un message d'avertissement du serveur est consigné dans le journal.

Warning:
Si une valeur de macro ne peut pas être récupérée correctement, l'élément correspondant qui utilise cette valeur deviendra non pris en charge.

Emplacements non masqués

Cette liste fournit les emplacements des paramètres où les valeurs des macros secrètes ne sont pas masquées.

Note:
Les valeurs des macros secrètes resteront masquées dans les emplacements ci-dessous si elles sont référencées indirectement. Par exemple, {ITEM.KEY}, {ITEM.KEY<1-9>}, {LLDRULE.KEY} macros intégrées utilisées dans les types de média (paramètres Script ou webhook) seront résolues en clés d'élément contenant des macros secrètes masquées, telles que `net.tcp.port[******,******]` au lieu de `net.tcp.port[192.0.2.1,80]`.

Context	Parameter
Éléments, prototypes d'élément, règles LLD	
Élément	*Paramètres de clé d'élément*
Prototype d'élément	*Paramètres de clé de prototype d'élément*
Règle de découverte de bas niveau	*Paramètres de clé d'élément de découverte*
agent SNMP	*Communauté SNMP*
	Nom de contexte (SNMPv3)
	Nom de sécurité (SNMPv3)
	Phrase secrète d'authentification (SNMPv3)
	Phrase secrète de confidentialité (SNMPv3)
agent HTTP	*URL*
	Champs de requête
	Corps de la requête
	En-têtes
	Nom d'utilisateur
	Mot de passe
	Mot de passe de la clé SSL
Script	*Paramètres*
	Script
Navigateur	*Paramètres*
	Script
Surveillance de base de données	*Requête SQL*
agent TELNET	*Script*
	Nom d'utilisateur
	Mot de passe
agent SSH	*Script*
	Nom d'utilisateur
	Mot de passe
Vérification simple	*Nom d'utilisateur*

Context	Parameter
	Mot de passe
agent JMX	*Nom d'utilisateur*
	Mot de passe
Prétraitement de la valeur d'élément	
Étape de prétraitement JavaScript	*Script*
Scénarios web	
Scénario web	*Valeur de variable*
	Valeur d'en-tête
	URL
	Valeur de champ de requête
	Valeur de champ POST
	POST brut
Authentification du scénario web	*Utilisateur*
	Mot de passe
	Mot de passe de la clé SSL
Connecteurs	
Connecteur	*URL*
	Nom d'utilisateur
	Mot de passe
	Jeton
	Proxy HTTP
	Fichier de certificat SSL
	Fichier de clé SSL
	Mot de passe de la clé SSL
Découverte réseau	
SNMP	*Communauté SNMP*
	Nom de contexte (SNMPv3)
	Nom de sécurité (SNMPv3)
	Phrase secrète d'authentification (SNMPv3)
	Phrase secrète de confidentialité (SNMPv3)
Scripts globaux	
Webhook	*Script JavaScript*
	Valeur du paramètre du script JavaScript
Telnet	*Nom d'utilisateur*
	Mot de passe
SSH	*Nom d'utilisateur*
	Mot de passe
Script	*Script*
Types de média	
Script	*Paramètres du script*
Webhook	*Paramètres*
Gestion IPMI	
Hôte	*Nom d'utilisateur*
	Mot de passe

Macros de découverte de bas niveau

Vue d'ensemble

Il existe un type de macro utilisé dans la fonction découverte de bas niveau (LLD) :

```
{#MACRO}
```

Il s'agit d'une macro utilisée dans une règle LLD qui renvoie les valeurs réelles du nom du système de fichiers, de l'interface réseau, de l'OID SNMP, etc.

Ces macros peuvent être utilisées pour créer des prototypes d'entités (prototypes d'élément, de déclencheur, de graphique, de règle LLD, d'hôte et de groupe d'hôtes). Lors de la découverte de systèmes de fichiers, d'interfaces réseau, de machines virtuelles, etc., ces macros sont remplacées par des valeurs réelles et constituent la base de la création d'entités réelles.

Certaines macros de découverte de bas niveau sont fournies « prêtes à l'emploi » avec la fonction LLD dans Zabbix - {#FSNAME}, {#FSTYPE}, {#IFNAME}, {#SNMPINDEX}, {#SNMPVALUE}. Cependant, il n'est pas obligatoire de respecter ces noms lors de la création d'une règle de découverte de bas niveau personnalisée. Vous pouvez alors utiliser n'importe quel autre nom de macro LLD et faire référence à ce nom.

Types de données pris en charge

Lors de la définition de règles de découverte personnalisées, les valeurs de propriété renvoyées dans les objets JSON pour les macros LLD doivent être de l'un des types primitifs suivants :

- string
- number
- boolean

Les tableaux, les objets et les valeurs null ne sont pas pris en charge. Toute macro LLD faisant référence à une telle valeur ne sera pas développée et apparaîtra littéralement (par ex. `'{#MY_MACRO}'`) lors du prétraitement et de la création des éléments.

Emplacements pris en charge

Les macros LLD peuvent être utilisées :

- dans le filtre de la règle de découverte de bas niveau
- dans les règles de découverte de bas niveau imbriquées, dans :
  - les paramètres de prétraitement JSONPath
  - le champ JSONPath pour les macros LLD personnalisées
- pour les prototypes d'éléments et les prototypes de découverte, dans :
  - le nom
  - les paramètres de clé
  - l'unité
  - l'intervalle de mise à jour[1]
  - le délai d'expiration[1]
  - la période de stockage de l'historique[1]
  - la période de stockage des tendances[1]
  - les étapes de prétraitement de la valeur de l'élément
  - l'OID SNMP
  - le champ du capteur IPMI
  - l'expression d'élément calculé/agrégé, dans :
    - les constantes d'expression et les paramètres de fonction
    - les paramètres de clé d'élément
  - les conditions de filtre d'élément agrégé (nom du groupe d'hôtes et nom du tag)
  - le script SSH et le script Telnet
  - la requête SQL de supervision de base de données
  - le champ de point de terminaison de l'élément JMX
  - la description
  - le champ URL de l'agent HTTP
  - le champ des champs de requête HTTP de l'agent HTTP
  - le champ du corps de requête de l'agent HTTP
  - le champ des codes d'état requis de l'agent HTTP
  - la clé et la valeur du champ des en-têtes de l'agent HTTP
  - le champ du nom d'utilisateur d'authentification HTTP de l'agent HTTP
  - le champ du mot de passe d'authentification HTTP de l'agent HTTP
  - le champ du proxy HTTP de l'agent HTTP
  - le champ du fichier de certificat SSL HTTP de l'agent HTTP
  - le champ du fichier de clé SSL HTTP de l'agent HTTP
  - le champ du mot de passe de clé SSL HTTP de l'agent HTTP
  - les tags

- pour les prototypes de déclencheur, dans :
  - le nom
  - les données opérationnelles
  - l'expression (uniquement dans les constantes et les paramètres de fonction)
  - l'URL
  - la description
  - les tags
- pour les prototypes de graphique, dans :
  - le nom
- pour les prototypes d'hôte, dans :
  - le nom
  - le nom visible
  - les champs d'interface personnalisés : IP, DNS, port, communauté SNMP v1/v2, nom de contexte SNMP v3, nom de sécurité SNMP v3, phrase secrète d'authentification SNMP v3, phrase secrète de confidentialité SNMP v3
  - le nom du prototype de groupe d'hôtes
  - la valeur du tag d'hôte
  - la valeur de la macro d'hôte
  - (voir la liste complète)

Dans tous ces emplacements, à l'exception du filtre de la règle de découverte de bas niveau, les macros LLD peuvent être utilisées à l'intérieur d'un contexte de macro utilisateur statique.

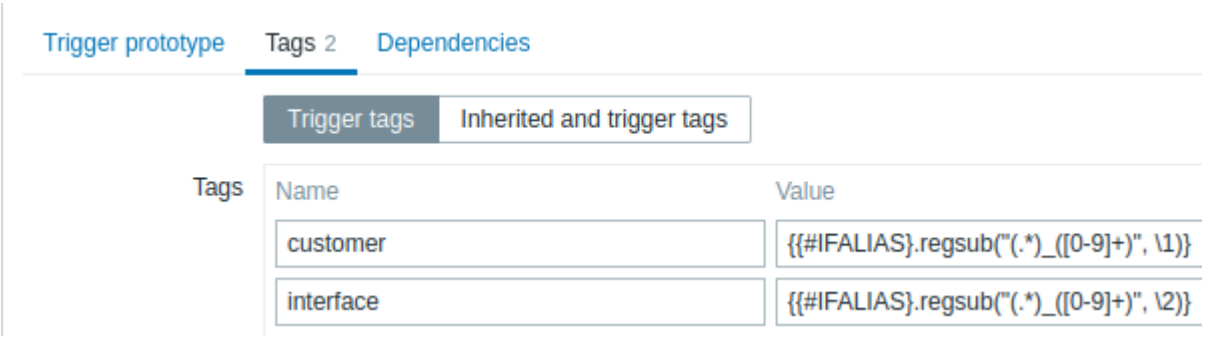
Utilisation des fonctions de macro

Les fonctions de macro sont prises en charge avec les macros de découverte de bas niveau (à l'exception du filtre de la règle de découverte de bas niveau), ce qui permet d'extraire une partie donnée de la valeur de la macro à l'aide d'une expression régulière.

Par exemple, vous pouvez vouloir extraire le nom du client et le numéro de l'interface à partir de la macro LLD suivante à des fins d'étiquetage des événements :

```
{#IFALIAS}=customername_1
```

Pour ce faire, la fonction de macro `regsub` peut être utilisée avec la macro dans le champ de valeur de l'étiquette d'événement d'un prototype de déclencheur :



Notez que les virgules ne sont pas autorisées dans les paramètres de clé d'un élément non placé entre guillemets ; le paramètre contenant une fonction de macro doit donc être placé entre guillemets. Le caractère barre oblique inverse (\) doit être utilisé pour échapper les guillemets doubles à l'intérieur du paramètre. Exemple :

```
net.if.in["{{#IFALIAS}.regsub(\"(.*)_([0-9]+)\", \1)}",bytes]
```

Pour plus d'informations sur la syntaxe des fonctions de macro, consultez : Fonctions de macro

Les fonctions de macro sont prises en charge dans les macros de découverte de bas niveau depuis Zabbix 4.0.

Prototypes d'élément sans paramètres de clé

Lorsque les paramètres de clé d'élément ne sont pas utilisés, placez la macro LLD à l'intérieur des crochets de paramètre de la clé d'élément `[...]` afin qu'elle soit traitée comme un paramètre et substituée lors de la découverte :

```
v_[{#MACRO}]
```

Notes de bas de page

[1] Dans les champs marqués avec [1], une seule macro doit remplir l'intégralité du champ. Plusieurs macros dans un champ ou des macros mélangées avec du texte ne sont pas prises en charge.

Macros d'expression

Aperçu

Les macros d'expression vous permettent d'effectuer des calculs dans des champs.

Leur valeur est calculée en résolvant d'abord toutes les macros internes, puis en évaluant l'expression résultante.

Syntaxe :

```
{?EXPRESSION}
```

`EXPRESSION` utilise la même syntaxe et prend en charge les mêmes fonctions que les expressions de déclencheur.

Exemple :

```
{?trendavg(/host/item1,1M:now/M)/trendavg(/host/item1,1M:now/M-1y)*100}
```

Pour les emplacements qui prennent en charge les macros d'expression, recherchez "{?EXPRESSION}" dans le tableau des macros prises en charge.

Notes d'utilisation :

- Utilisez les macros {FUNCTION.*} pour référencer les valeurs de fonction des expressions de déclencheur/expressions de récupération.
- Utilisez les macros {HOST.HOST<1-9>} et {ITEM.KEY<1-9>} pour référencer les hôtes et les éléments.
- Dans les modèles, utilisez les macros {HOST.HOST<1-9>} ou omettez complètement l'hôte pour le premier hôte — par exemple, `{?avg(//item1,1h)}` — au lieu des noms de modèle, car les noms de modèle ne sont pas remplacés par les noms d'hôte lors de la liaison de modèle.

```
{?{FUNCTION.VALUE2} - {FUNCTION.VALUE3}}
{?max(/{HOST.HOST}/{ITEM.KEY},3h)}
```

Voir aussi les exemples d'expressions de déclencheur pour un exemple d'utilisation d'une macro d'expression dans un nom d'événement.

Utilisateurs et groupes d'utilisateurs

Aperçu

Tous les utilisateurs de Zabbix accèdent à l'application Zabbix via l'interface web. Chaque utilisateur se voit attribuer un nom de connexion unique et un mot de passe.

Tous les mots de passe des utilisateurs sont chiffrés et stockés dans la base de données Zabbix. Les utilisateurs ne peuvent pas utiliser leur identifiant utilisateur et leur mot de passe pour se connecter directement au serveur UNIX, sauf s'ils ont également été configurés en conséquence dans UNIX. La communication entre le serveur web et le navigateur de l'utilisateur peut être protégée à l'aide de SSL.

Grâce à un schéma de permissions utilisateur flexible, vous pouvez restreindre et différencier les droits pour :

- Accéder aux fonctions administratives de l'interface Zabbix.
- Effectuer certaines actions dans l'interface.
- Accéder aux hôtes surveillés dans les groupes d'hôtes.
- Utiliser des méthodes API spécifiques.

Configuration d'un utilisateur

Vue d'ensemble

L'installation initiale de Zabbix comporte deux utilisateurs prédéfinis :

- *Admin* - un superutilisateur Zabbix disposant de tous les droits.
- *guest* - un utilisateur Zabbix spécial. L'utilisateur 'guest' est désactivé par défaut. Pour activer cet utilisateur, supprimez le groupe prédéfini *Disabled* des groupes d'utilisateurs. Vous pouvez vous connecter avec cet utilisateur et accéder aux pages de supervision dans Zabbix. Notez que, par défaut, 'guest' ne dispose d'aucun droit sur les objets Zabbix.

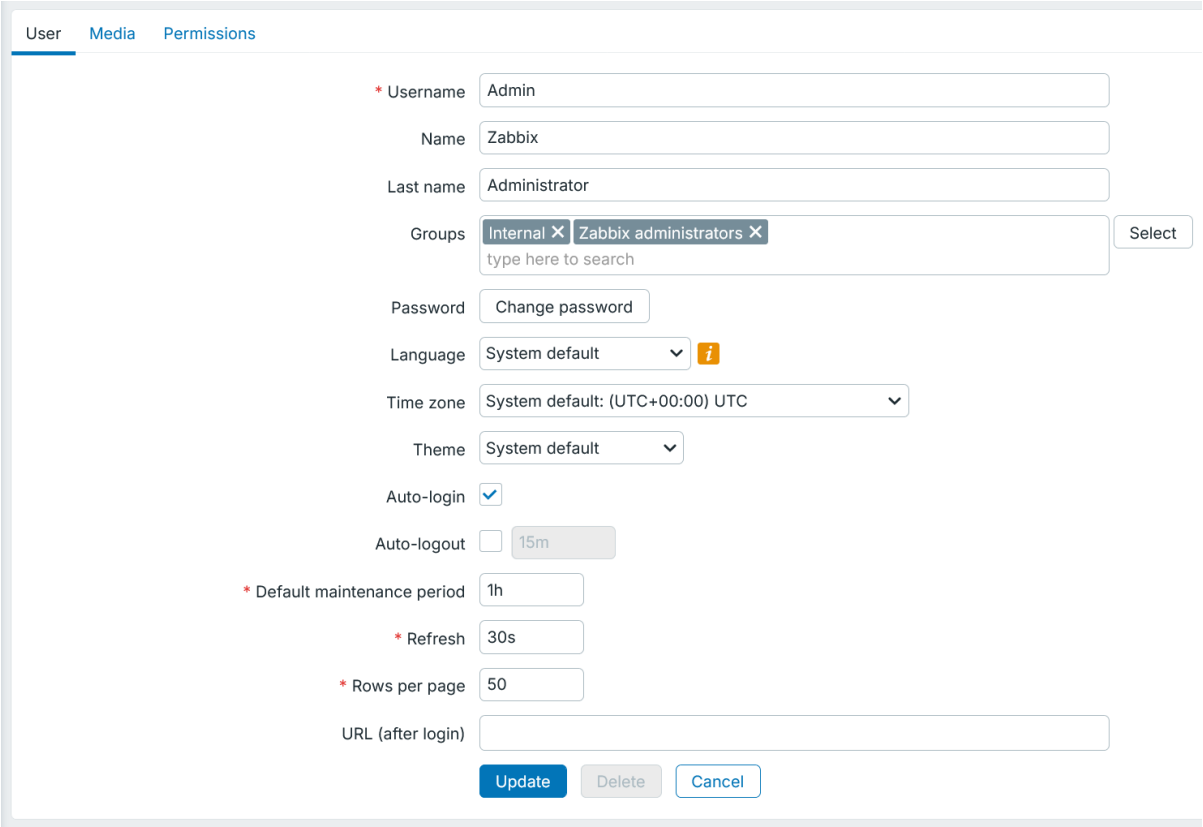
Pour configurer un utilisateur :

1. Accédez à *Utilisateurs* > *Utilisateurs*.

2. Cliquez sur *Créer un utilisateur* (ou sur le nom d'un utilisateur pour modifier un utilisateur existant).
3. Modifiez les attributs de l'utilisateur dans le formulaire.

Attributs généraux

L'onglet *Utilisateur* contient les attributs généraux de l'utilisateur :



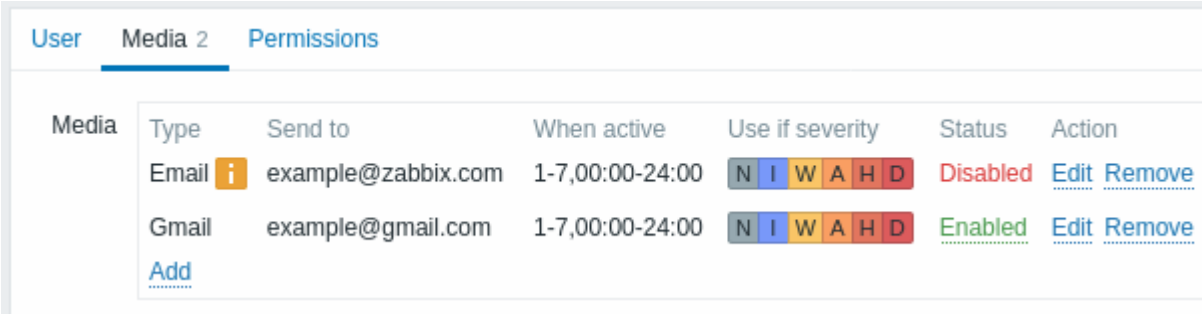
Tous les champs obligatoires sont marqués d'un astérisque rouge.

Parameter	Description
Nom d'utilisateur	Nom d'utilisateur unique, utilisé comme identifiant de connexion.
Prénom	Prénom de l'utilisateur (facultatif). S'il n'est pas vide, il est visible dans les informations d'acquittement et les informations sur le destinataire de la notification.
Nom	Nom de famille de l'utilisateur (facultatif). S'il n'est pas vide, il est visible dans les informations d'acquittement et les informations sur le destinataire de la notification.
Groupes	Sélectionnez les groupes d'utilisateurs auxquels l'utilisateur appartient. Ce champ propose la saisie semi-automatique : commencez à saisir le nom d'un groupe d'utilisateurs pour afficher une liste déroulante des groupes correspondants. Faites défiler la liste pour effectuer une sélection. Vous pouvez également cliquer sur *Sélectionner* pour ajouter des groupes. Cliquez sur « x » pour supprimer le groupe sélectionné. L'appartenance à des groupes d'utilisateurs détermine les groupes d'hôtes et les hôtes auxquels l'utilisateur a accès.
Mot de passe	Deux champs permettent de saisir le mot de passe de l'utilisateur, ou un bouton *Modifier le mot de passe* s'affiche si l'utilisateur existe déjà. Cliquer sur le bouton *Modifier le mot de passe* ouvre deux champs permettant de saisir un nouveau mot de passe. Pour l'utilisateur ayant le *rôle de super administrateur* qui modifie son propre mot de passe, cliquer sur le bouton *Modifier le mot de passe* ouvre un champ supplémentaire permettant de saisir le mot de passe actuel (ancien). Une fois le mot de passe modifié, l'utilisateur dont le mot de passe a été modifié est déconnecté de toutes ses sessions actives. Notez que le mot de passe ne peut être modifié que pour les utilisateurs utilisant l'authentification interne de Zabbix.
Langue	Langue de l'interface Zabbix. L'extension php gettext est requise pour que les traductions fonctionnent.

Parameter	Description
Fuseau horaire	Sélectionnez le fuseau horaire qui remplacera le fuseau horaire global au niveau de l'utilisateur, ou sélectionnez **Valeur par défaut du système** pour utiliser les paramètres globaux du fuseau horaire.
Thème	Définit l'apparence de l'interface : **Valeur par défaut du système** - utilise les paramètres système par défaut **Bleu** - thème bleu standard **Bleu (classique)** - thème bleu standard utilisant les anciennes polices **Sombre** - thème sombre alternatif **Bleu sombre** - thème bleu sombre **Sombre (classique)** - thème sombre alternatif utilisant les anciennes polices **Clair à contraste élevé** - thème clair à contraste élevé **Sombre à contraste élevé** - thème sombre à contraste élevé
Connexion automatique	Cochez cette case pour que Zabbix mémorise l'utilisateur et le connecte automatiquement pendant 30 jours. Lors d'une connexion avec *Se souvenir pendant 30 jours* : - La déconnexion automatique est réinitialisée (la session persiste pendant 30 jours). - La connexion automatique est activée pour permettre une réauthentification transparente. Lors d'une connexion sans *Se souvenir pendant 30 jours* : - La connexion automatique est activée pour permettre une réauthentification transparente. - La déconnexion automatique reste régie par le délai d'expiration standard. Des cookies de navigateur sont utilisés à cette fin.
Déconnexion automatique	Lorsque cette case est cochée, l'utilisateur est automatiquement déconnecté après le nombre de secondes défini (minimum 90 secondes, maximum 1 jour). Notez que ce paramètre est remplacé si *Se souvenir pendant 30 jours* est activé, car la session est prolongée pendant toute cette période. Les suffixes de temps sont pris en charge, par exemple 90s, 5m, 2h, 1d. Notez que cette option ne fonctionnera pas : * Si l'option de configuration globale « Afficher un avertissement si le serveur Zabbix est arrêté » est activée et que l'interface Zabbix reste ouverte. * Lorsque les pages du menu Surveillance actualisent les informations en arrière-plan. * Si la connexion est effectuée avec l'option *Se souvenir de moi pendant 30 jours* cochée.
Période de maintenance par défaut	Définissez la période de maintenance par défaut pour les maintenances ponctuelles.
Actualisation	Définissez la fréquence d'actualisation utilisée pour les graphiques, les données en texte brut, etc. Peut être définie sur 0 pour la désactiver. Les suffixes de temps sont pris en charge, par exemple 90s, 5m, 1h.
Lignes par page	Vous pouvez déterminer le nombre de lignes à afficher par page dans les listes.
URL (après la connexion)	Vous pouvez faire en sorte que Zabbix redirige l'utilisateur vers une URL spécifique après une connexion réussie, par exemple vers la page *Problèmes*.

Médias utilisateur

L'onglet *Médias* contient la liste de tous les médias définis pour l'utilisateur. Les médias sont utilisés pour envoyer des notifications.



Cliquez sur *Ajouter* pour attribuer un média à l'utilisateur.

Si le type de média a été désactivé :

- Une icône d'information jaune est affichée après le nom.
- *Désactivé* est affiché dans la colonne Statut.

Note:
Les autorisations permettant aux utilisateurs de modifier eux-mêmes les détails de leurs médias peuvent être accordées ou révoquées en fonction de leur rôle utilisateur (voir l'option *Créer et modifier ses propres médias*). Les autorisations des super administrateurs leur permettant de modifier les détails des médias d'autres utilisateurs peuvent également être accordées ou révoquées en fonction de leur rôle utilisateur (voir l'option *Créer et modifier les médias utilisateur*).

Consultez la section Types de médias pour plus d'informations sur la configuration des médias utilisateur.

Permissions

L'onglet *Permissions* contient des informations sur les éléments suivants :

- Rôle utilisateur (obligatoire pour tout nouvel utilisateur créé), qui ne peut être modifié que par un utilisateur de type *Super administrateur*.

Attention:
Les utilisateurs ne peuvent pas être créés sans rôle utilisateur (sauf avec l'API User de Zabbix). Les utilisateurs créés précédemment qui n'ont pas de rôle peuvent toujours être modifiés sans qu'un rôle leur soit attribué. Cependant, une fois qu'un rôle est attribué, il peut uniquement être modifié et non supprimé.

 Notez que les utilisateurs sans rôle peuvent se connecter à Zabbix uniquement à l'aide de l'authentification LDAP ou SAML, à condition que leurs informations LDAP/SAML correspondent aux mappages de groupes d'utilisateurs configurés dans Zabbix.

- Type d'utilisateur (*Utilisateur*, *Administrateur*, *Super administrateur*) défini dans la configuration du rôle utilisateur.
- Groupes d'hôtes et de modèles auxquels l'utilisateur a accès.
  - Par défaut, les utilisateurs de type *Utilisateur* et *Administrateur* n'ont accès à aucun groupe, modèle ni hôte. Pour leur accorder cet accès, les utilisateurs doivent être inclus dans des groupes d'utilisateurs configurés avec des permissions sur les entités concernées.
- Droits d'accès aux sections et aux éléments de l'interface Zabbix, aux modules et aux méthodes de l'API.
  - Les éléments auxquels l'accès est autorisé sont affichés en vert, tandis que ceux dont l'accès est refusé sont affichés en gris clair.
  - Si l'accès à l'API est activé, les méthodes de l'API du rôle utilisateur sont répertoriées sous *Méthodes autorisées* ou *Méthodes refusées*, selon que *Liste d'autorisation* ou *Liste de refus* est sélectionnée dans la configuration du rôle utilisateur. Si aucune méthode d'API n'est spécifiée, *Aucune* est affiché. *Aucune* sous *Méthodes autorisées* signifie qu'aucune méthode d'API n'est autorisée, tandis que *Aucune* sous *Méthodes refusées* signifie que toutes les méthodes d'API disponibles pour le type d'utilisateur sont autorisées.
- Droits d'effectuer des actions spécifiques.
  - Les actions que l'utilisateur est autorisé à effectuer sont affichées en vert, tandis que celles qui sont refusées sont affichées en gris clair.

Consultez la page Permissions pour plus de détails.

Permissions

Vue d'ensemble

Les autorisations dans Zabbix dépendent du type d'utilisateur, des rôles utilisateur personnalisés et de l'accès aux hôtes, qui est défini en fonction du groupe d'utilisateurs.

Types d'utilisateurs

Les autorisations dans Zabbix dépendent principalement du type d'utilisateur :

- *Utilisateur* - dispose de droits d'accès limités aux sections du menu (voir ci-dessous) et, par défaut, n'a accès à aucune ressource. Toute autorisation sur les groupes d'hôtes ou de modèles doit être attribuée explicitement.
- *Administrateur* - dispose de droits d'accès incomplets aux sections du menu (voir ci-dessous). Par défaut, l'utilisateur n'a accès à aucun groupe d'hôtes. Toute autorisation sur les groupes d'hôtes ou de modèles doit être attribuée explicitement.
- *Super administrateur* - a accès à toutes les sections du menu. L'utilisateur dispose d'un accès en lecture-écriture à tous les groupes d'hôtes et de modèles. Les autorisations ne peuvent pas être révoquées en refusant l'accès à des groupes spécifiques.

Accès au menu

Note:
L'accès restreint à certains éléments de l'interface empêche uniquement l'ouverture de cette page ; il ne supprime pas la possibilité d'accéder aux données sous-jacentes depuis d'autres parties de l'interface.

Le tableau suivant illustre l'accès aux sections du menu Zabbix selon le type d'utilisateur :

Section du menu		Utilisateur	Administrateur	Super administrateur
Tableaux de bord		+	+	+
Supervision		+	+	+
	Problèmes	+	+	+
	Hôtes	+	+	+
	Dernières données	+	+	+
	Cartes	+	+	+
	Découverte		+	+
APM		+	+	+
	Traces	+	+	+
	Métriques	+	+	+
	Journaux	+	+	+
Services		+	+	+
	Services	+	+	+
	SLA		+	+
	Rapport SLA	+	+	+
Inventaire		+	+	+
	Vue d'ensemble	+	+	+
	Hôtes	+	+	+
Rapports		+	+	+
	Informations système		+	+
	Rapports planifiés		+	+
	Rapport de disponibilité	+	+	+
	Top 100 des déclencheurs	+	+	+
	Journal d'audit			+
	Journal des actions			+
	Notifications		+	+
Collecte de données			+	+
	Groupes de modèles		+	+
	Groupes d'hôtes		+	+
	Modèles		+	+
	Hôtes		+	+
	Maintenance		+	+
	Corrélation des événements			+
	Découverte		+	+
Alertes			+	+
	Actions de déclencheur		+	+
	Actions de service		+	+
	Actions de découverte		+	+
	Actions d'enregistrement automatique		+	+
	Actions internes		+	+
	Types de média			+
	Scripts			+
Utilisateurs				+
	Groupes d'utilisateurs			+
	Rôles d'utilisateurs			+
	Utilisateurs			+
	Jetons API			+
	Authentification			+
	Appareils			+
Administration				+
	Source de données			+
	Général			+
	Journal d'audit			+
	Maintenance des données			+
	Groupes de proxy			+
	Proxys			+
	Macros			+
	File d'attente			+

Rôles utilisateur

Les rôles utilisateur permettent d'apporter des ajustements personnalisés aux autorisations définies par le type d'utilisateur. Aucune autorisation ne peut être ajoutée (cela dépasserait celles du type d'utilisateur), mais certaines autorisations peuvent être révoquées.

En outre, un rôle utilisateur détermine l'accès non seulement aux sections du menu, mais également aux services, aux modules, aux méthodes API et à diverses actions dans l'interface.

Les rôles utilisateur sont configurés dans la section *Utilisateurs* > *Rôles utilisateur* par les utilisateurs Super administrateur.

Les rôles utilisateur sont attribués aux utilisateurs dans le formulaire de configuration de l'utilisateur, onglet *Autorisations*, par les utilisateurs Super administrateur.

Users

User | Media | Permissions

* Role: Admin role ✕ | Select

User type: Admin

Permissions

User group permissions

Group	Type	Permissions
All groups	Hosts	None
All groups	Templates	None

Proxies: None

Proxy groups: None

Permissions can be assigned for user groups only.

Access to UI elements

Dashboards: Dashboards

Monitoring: Problems | Hosts | Latest data | Maps | Discovery

APM: Traces | Metrics | Logs

Services: Services | SLA | SLA report

Inventory: Overview | Hosts

Reports: System information | Scheduled reports | Availability report | Top 100 triggers | Notifications

Data collection: Template groups | Host groups | Templates | Hosts | Maintenance | Discovery

Alerts: Trigger actions | Service actions | Discovery actions | Autoregistration actions | Internal actions

User profile settings

Redirect URL after login

Access to services

Read-write access to services: None

Read-only access to services: All

Access to modules

Action log | Clock | Discovery status | Favorite graphs | Favorite maps | Gauge | Geomap | Graph | Graph (classic) | Graph prototype | Honeycomb | Host availability | Host card | Host navigator | Item card | Item history | Item navigator | Item value | Map | Map navigation tree | Pie chart | Problem hosts | Problems | Problems by severity | Scatter plot | SLA report | System information | Top hosts | Top items | Top triggers | Trigger overview | URL | Web monitoring

Access to API

Disabled

Devices

Enabled | Manage own devices

Access to actions

Create and edit dashboards | Create and edit maps | Create and edit maintenance | Add problem comments | Change severity | Acknowledge problems | Suppress problems | Close problems | Execute scripts | Manage API tokens | Manage scheduled reports | Manage SLA | Invoke "Execute now" on read-only hosts | Change problem ranking | Create and edit own media | Select "Server" for monitoring and discovery

Add | Cancel

Accès aux hôtes

L'accès à toutes les données des hôtes et des modèles dans Zabbix est accordé aux groupes d'utilisateurs uniquement au niveau des groupes d'hôtes/de modèles.

Cela signifie qu'un utilisateur individuel ne peut pas se voir accorder directement l'accès à un hôte (ou à un groupe d'hôtes). L'accès à un hôte ne peut lui être accordé qu'en faisant partie d'un groupe d'utilisateurs auquel l'accès au groupe d'hôtes contenant l'hôte a été accordé.

De même, l'accès à un modèle ne peut être accordé à un utilisateur qu'en faisant partie d'un groupe d'utilisateurs auquel l'accès au groupe de modèles contenant le modèle a été accordé.

Groupes d'utilisateurs

Vue d'ensemble

Les groupes d'utilisateurs vous permettent de regrouper les utilisateurs à des fins organisationnelles et d'attribuer des permissions d'accès aux données. Les permissions permettant de consulter et de configurer les données des groupes d'hôtes et des groupes de modèles sont attribuées aux groupes d'utilisateurs, et non aux utilisateurs individuels.

Il peut souvent être pertinent de séparer les informations accessibles à un groupe d'utilisateurs de celles accessibles à un autre. Pour ce faire, regroupez les utilisateurs, puis attribuez des permissions différentes aux groupes d'hôtes et aux groupes de modèles.

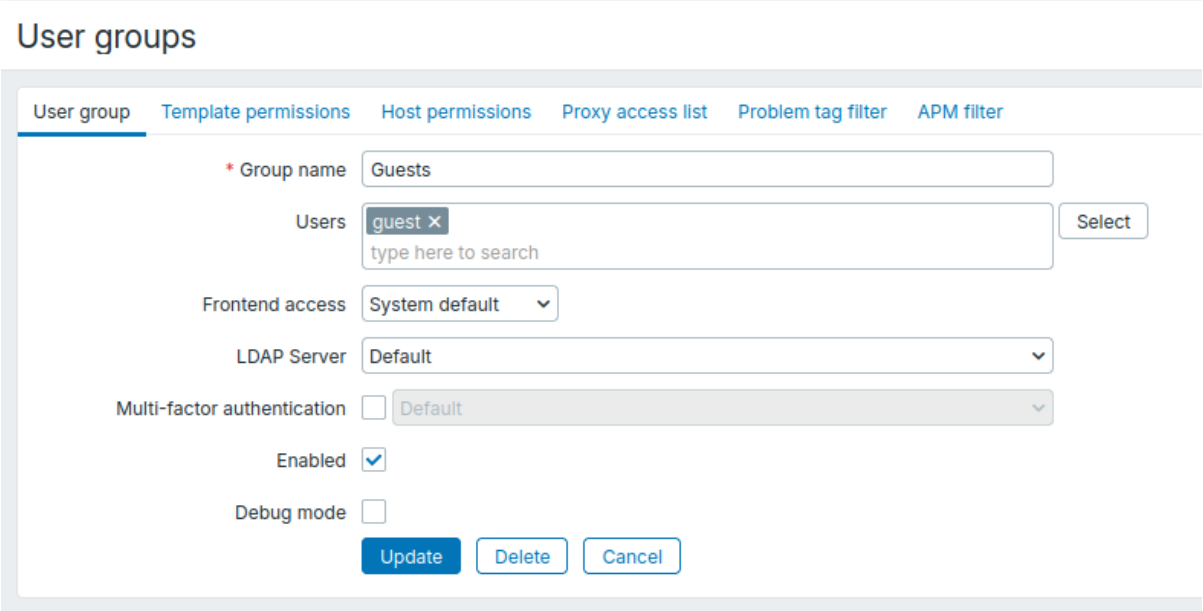
Un utilisateur peut appartenir à un nombre quelconque de groupes.

Configuration

Pour configurer un groupe d'utilisateurs :

1. Accédez à *Utilisateurs* > *Groupes d'utilisateurs*.
2. Cliquez sur *Créer un groupe d'utilisateurs* (ou sur le nom du groupe pour modifier un groupe existant).
3. Modifiez les attributs du groupe dans le formulaire.

L'onglet **Groupe d'utilisateurs** contient les attributs généraux du groupe :

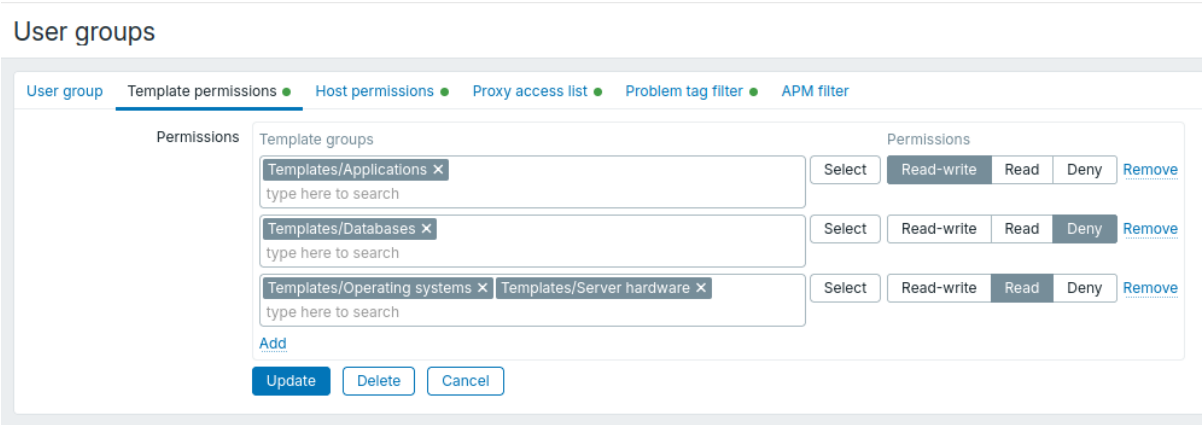


Tous les champs obligatoires sont marqués d'un astérisque rouge.

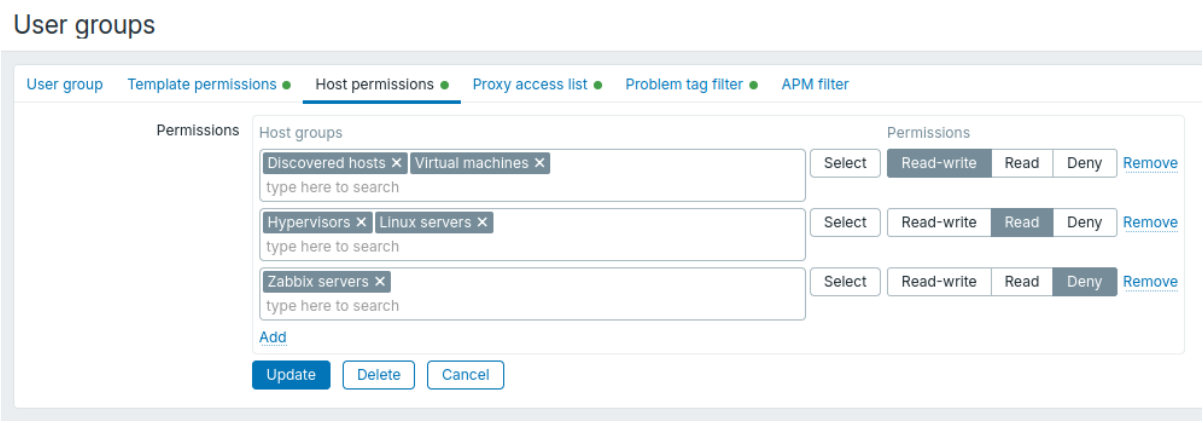
Paramètre	Description
Nom du groupe	Nom unique du groupe.
Utilisateurs	Pour ajouter des utilisateurs au groupe, commencez à saisir le nom d'un utilisateur existant. Lorsque la liste déroulante contenant les noms d'utilisateurs correspondants apparaît, faites défiler la liste pour effectuer votre sélection. Vous pouvez également cliquer sur le bouton *Sélectionner* pour sélectionner des utilisateurs dans une fenêtre contextuelle.

Paramètre	Description
Accès à l'interface	Méthode d'authentification des utilisateurs du groupe. **Par défaut du système** - utilise la méthode d'authentification par défaut (définie globalement) **Interne** - utilise l'authentification interne de Zabbix (même si l'authentification LDAP est utilisée globalement). Ignoré si l'authentification HTTP est la méthode globale par défaut. **LDAP** - utilise l'authentification LDAP (même si l'authentification interne est utilisée globalement). Ignoré si l'authentification HTTP est la méthode globale par défaut. **Désactivé** - l'accès à l'interface Zabbix est interdit pour ce groupe
Serveur LDAP	Sélectionnez le serveur LDAP à utiliser pour authentifier l'utilisateur. Ce champ est activé uniquement si *Accès à l'interface* est défini sur LDAP ou Par défaut du système.
Authentification multifacteur	Sélectionnez la méthode d'authentification multifacteur à utiliser pour authentifier l'utilisateur : **Par défaut** - utilise la méthode définie par défaut dans la configuration MFA ; cette option est sélectionnée par défaut pour les nouveaux groupes d'utilisateurs si la MFA est activée ; **<Nom de la méthode>** - utilise la méthode sélectionnée (par exemple, « Zabbix TOTP ») ; **Désactivée** - la MFA est désactivée pour ce groupe ; cette option est sélectionnée par défaut pour les nouveaux groupes d'utilisateurs si la MFA est désactivée. Notez que si un utilisateur appartient à plusieurs groupes d'utilisateurs avec la MFA activée (ou si au moins un groupe a la MFA activée), les règles d'authentification suivantes s'appliquent : si un groupe utilise la méthode MFA « Par défaut », celle-ci sera utilisée pour authentifier l'utilisateur ; sinon, la première méthode (par ordre alphabétique) sera utilisée pour l'authentification.
Activé	État du groupe d'utilisateurs et de ses membres. *Coché* - le groupe d'utilisateurs et ses utilisateurs sont activés *Non coché* - le groupe d'utilisateurs et ses utilisateurs sont désactivés
Mode débogage	Cochez cette case pour activer le mode débogage pour les utilisateurs.

L'onglet **Autorisations des modèles** permet de spécifier l'accès du groupe d'utilisateurs aux groupes de modèles et, par leur intermédiaire, aux données des modèles :



L'onglet **Autorisations des hôtes** permet de spécifier l'accès du groupe d'utilisateurs aux groupes d'hôtes et, par leur intermédiaire, aux données des hôtes :



Cliquez sur Add pour choisir les groupes de modèles/hôtes (qu'il s'agisse d'un groupe parent ou imbriqué) et leur attribuer des

autorisations. Commencez à saisir le nom du groupe (une liste déroulante des groupes correspondants apparaîtra) ou cliquez sur *Sélectionner* pour ouvrir une fenêtre contextuelle répertoriant tous les groupes.

Utilisez ensuite les boutons d'option pour attribuer des autorisations aux groupes sélectionnés. Les autorisations possibles sont les suivantes :

- **Lecture-écriture** - accès en lecture-écriture à un groupe
- **Lecture** - accès en lecture seule à un groupe
- **Refuser** - accès refusé à un groupe

Si le même groupe de modèles/hôtes est ajouté sur plusieurs lignes avec des autorisations différentes, l'autorisation la plus restrictive sera appliquée.

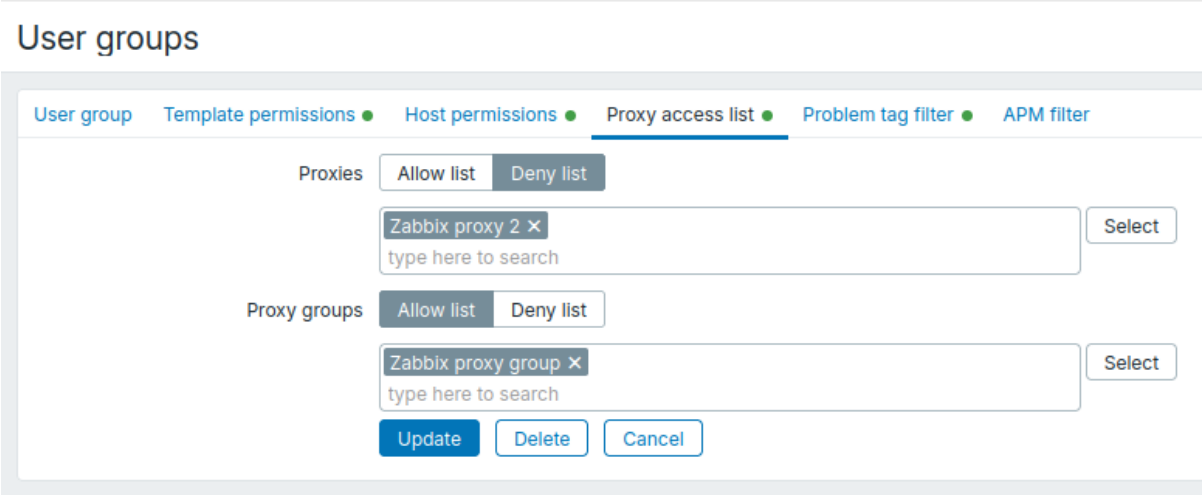
Notez qu'un utilisateur *Super administrateur* peut imposer aux groupes imbriqués d'avoir le même niveau d'autorisations que le groupe parent ; cela peut être configuré dans le formulaire de configuration du groupe d'hôtes/de modèles.

Les onglets **Autorisations des modèles** et **Autorisations des hôtes** prennent en charge le même ensemble de paramètres.

Les autorisations actuelles accordées aux groupes sont affichées dans le bloc *Autorisations* ; elles peuvent être modifiées ou supprimées.

Note:
Si un groupe d'utilisateurs dispose des autorisations **Lecture-écriture** sur un hôte et de l'autorisation **Refuser** ou d'aucune autorisation sur un modèle lié à cet hôte, les utilisateurs de ce groupe ne pourront pas modifier les éléments issus du modèle sur l'hôte, et le nom du modèle sera affiché comme *Modèle inaccessible*.

L'onglet **Liste d'accès aux proxy** permet de spécifier l'accès du groupe d'utilisateurs aux proxy, individuellement ou par l'intermédiaire de groupes de proxy :



Utilisez le champ *Groupes de proxy* pour appliquer l'autorisation aux groupes sélectionnés. Un proxy appartenant à un groupe ne peut être autorisé ou refusé que par l'intermédiaire de ce groupe. Utilisez le champ *Proxies* pour ajouter des proxy individuels qui n'appartiennent à aucun groupe de proxy.

Proxies et *Groupes de proxy* prennent en charge le même ensemble de paramètres.

Cliquez sur le bouton *Sélectionner* à côté du champ *Proxies* ou *Groupes de proxy* pour sélectionner respectivement des proxy ou des groupes de proxy. Vous pouvez également commencer à saisir le nom d'un proxy ou d'un groupe de proxy dans le champ correspondant. Une liste déroulante des proxy ou groupes de proxy correspondants apparaît.

Sélectionnez *Liste d'autorisation* pour autoriser le groupe d'utilisateurs à accéder uniquement aux proxy et groupes de proxy sélectionnés.
 Si la sélection *Proxies* est vide pour la *Liste d'autorisation*, le groupe d'utilisateurs n'a accès à aucun proxy n'appartenant pas à un groupe de proxy. Si la sélection *Groupes de proxy* est vide pour la *Liste d'autorisation*, le groupe d'utilisateurs n'a accès à aucun groupe de proxy.

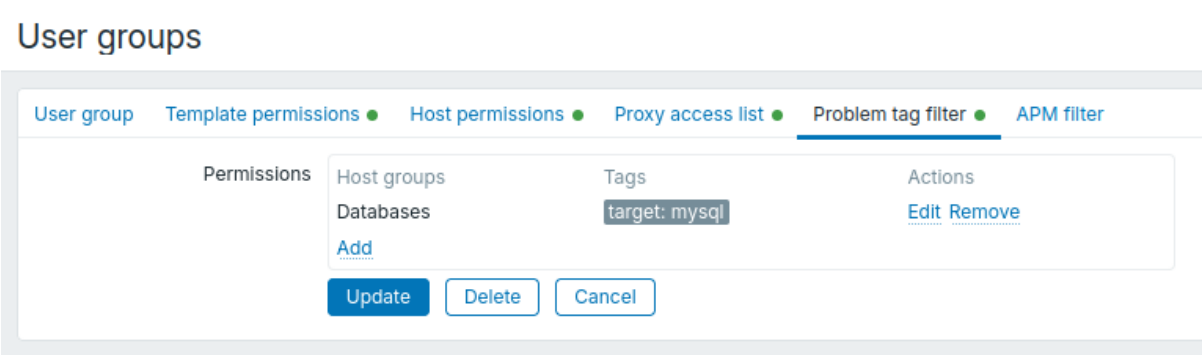
Sélectionnez *Liste de refus* pour refuser au groupe d'utilisateurs l'accès aux proxy et groupes de proxy sélectionnés. L'accès à tous les autres proxy et groupes de proxy est autorisé.
 Si la sélection *Proxies* est vide pour la *Liste de refus*, le groupe d'utilisateurs dispose d'un accès complet à tous les proxy n'appartenant pas à un groupe de proxy. Si la sélection *Groupes de proxy* est vide pour la *Liste de refus*, le groupe d'utilisateurs dispose d'un accès complet à tous les groupes de proxy.

Règles de priorité des autorisations pour les utilisateurs appartenant à plusieurs groupes d'utilisateurs :

- La *Liste de refus* est prioritaire sur la *Liste d'autorisation*. Si un groupe refuse l'accès à un proxy ou à un groupe de proxy, l'utilisateur ne peut pas y accéder.

- Les autorisations de la *Liste d'autorisation* sont cumulatives. Si plusieurs groupes autorisent l'accès, l'utilisateur peut accéder à tous les proxy autorisés par ces groupes.

L'onglet **Filtre des balises de problème** permet de définir des autorisations basées sur les balises pour que les groupes d'utilisateurs puissent voir les problèmes filtrés par nom et valeur de balise :

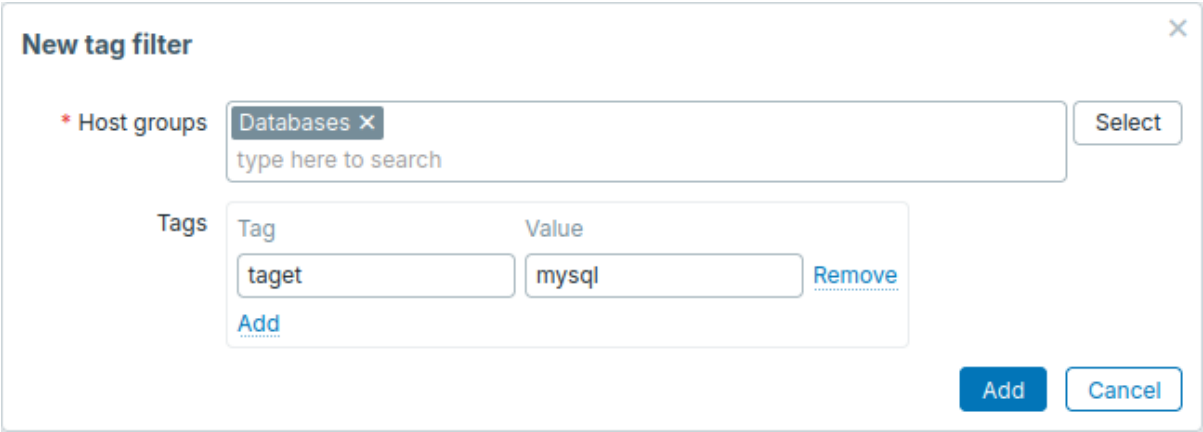


Cliquez sur Add pour choisir les groupes d'hôtes. Pour sélectionner un groupe d'hôtes auquel appliquer un filtre de balises, cliquez sur *Sélectionner* pour obtenir la liste complète des groupes d'hôtes existants, ou commencez à saisir le nom d'un groupe d'hôtes pour obtenir une liste déroulante des groupes correspondants. Seuls les groupes d'hôtes seront affichés, car un filtre de balises de problème ne peut pas être appliqué aux groupes de modèles.

Les noms de balises sans valeurs peuvent être ajoutés, mais les valeurs sans noms ne le peuvent pas. Seules les trois premières balises (avec leurs valeurs, le cas échéant) sont affichées dans le bloc *Autorisations* ; s'il y en a davantage, elles peuvent être consultées en cliquant sur l'icône ••• ou en la survolant.

Le filtre de balises permet de séparer l'accès au groupe d'hôtes de la possibilité de voir les problèmes.

Par exemple, si un administrateur de base de données doit voir uniquement les problèmes liés aux bases de données « MySQL », il faut d'abord créer un groupe d'utilisateurs pour les administrateurs de bases de données, puis spécifier le nom de balise « target » et la valeur « mysql ».



Si le nom de balise « target » est spécifié et que le champ de valeur est laissé vide, le groupe d'utilisateurs verra tous les problèmes portant le nom de balise « target » pour le groupe d'hôtes sélectionné.

Assurez-vous que le nom et la valeur de la balise sont correctement spécifiés ; sinon, le groupe d'utilisateurs ne verra aucun problème.

Examinons un exemple dans lequel un utilisateur est membre de plusieurs groupes d'utilisateurs sélectionnés. Dans ce cas, le filtrage utilisera une condition OR pour les balises.

Groupe d'utilisateurs A			**Groupe d'utilisateurs B**			**Résultat visible pour un utilisateur (membre) des deux groupes**
Filtre de balises						
Groupe d'hôtes	*Nom de balise*	*Valeur de balise*	*Groupe d'hôtes*	*Nom de balise*	*Valeur de balise*	

Bases de données	target	mysql	Bases de données	target	oracle	problèmes target:mysql ou target:oracle visibles
Non configuré dans le **Filtre des balises de problème**			Bases de données	target	oracle	problèmes target:oracle visibles

Accès depuis plusieurs groupes d'utilisateurs

Un utilisateur peut appartenir à un nombre quelconque de groupes d'utilisateurs. Ces groupes peuvent avoir des autorisations d'accès différentes aux hôtes ou aux modèles.

Il est donc important de savoir à quelles entités un utilisateur non privilégié pourra accéder au final. Dans l'exemple suivant, examinez comment l'accès à l'hôte **X** (dans le groupe d'hôtes 1) sera affecté dans différentes situations pour un utilisateur appartenant aux groupes d'utilisateurs A et B.

- Si le groupe A dispose uniquement d'un accès *Lecture* au groupe d'hôtes 1, mais que le groupe B dispose d'un accès *Lecture-écriture* au groupe d'hôtes 1, l'utilisateur obtiendra un accès **Lecture-écriture** à « X ».

Attention:
Les autorisations « Lecture-écriture » ont priorité sur les autorisations « Lecture ».

- Dans le même scénario que ci-dessus, si « X » appartient simultanément aussi au groupe d'hôtes 2 qui est **refusé** au groupe A ou B, l'accès à « X » sera **indisponible**, malgré un accès *Lecture-écriture* au groupe d'hôtes 1.
- Si aucune autorisation n'est définie pour le groupe A et que le groupe B dispose d'un accès *Lecture-écriture* au groupe d'hôtes 1, l'utilisateur obtiendra un accès **Lecture-écriture** à « X ».
- Si le groupe A dispose d'un accès *Refuser* au groupe d'hôtes 1 et que le groupe B dispose d'un accès *Lecture-écriture* au groupe d'hôtes 1, l'accès à « X » sera **refusé** à l'utilisateur.

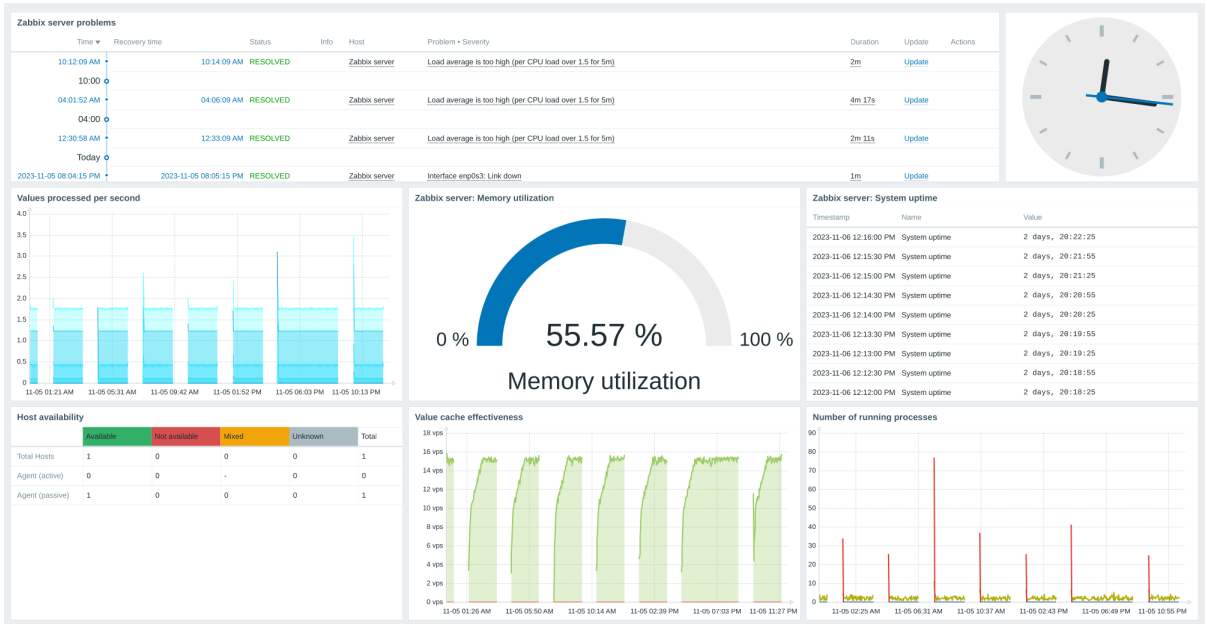
Autres détails

- Un utilisateur de niveau Admin disposant d'un accès *Lecture-écriture* à un hôte ne pourra pas lier/délier des modèles s'il n'a pas accès au groupe de modèles auquel ils appartiennent. Avec un accès *Lecture* au groupe de modèles, il pourra lier/délier des modèles à l'hôte, mais ne verra aucun modèle dans la liste des modèles et ne pourra pas effectuer d'opérations sur les modèles à d'autres endroits.
- Un utilisateur de niveau Admin disposant d'un accès *Lecture* à un hôte ne verra pas l'hôte dans la liste des hôtes de la section de configuration ; toutefois, les déclencheurs de l'hôte seront accessibles dans la configuration des services IT.
- Tout utilisateur autre qu'un Super Admin (y compris « guest ») peut voir les cartes réseau tant que la carte est vide ou ne contient que des images. Lorsque des hôtes, des groupes d'hôtes ou des déclencheurs sont ajoutés à la carte, les permissions sont respectées.
- Le serveur Zabbix n'enverra pas de notifications aux utilisateurs définis comme destinataires des opérations d'action si l'accès à l'hôte concerné est explicitement « refusé ».

Rapports planifiés

Présentation

Avec la fonctionnalité *Rapports planifiés*, vous pouvez configurer l'envoi d'une version PDF d'un tableau de bord donné aux destinataires spécifiés à intervalles réguliers.



Prérequis :

- Le service web Zabbix doit être installé et configuré correctement pour permettre la génération de rapports planifiés - consultez Configuration des rapports planifiés pour obtenir des instructions.
- Un utilisateur doit disposer d'un rôle utilisateur de type *Administrateur* ou *Super administrateur* avec les autorisations suivantes :
  - *Rapports planifiés* dans le bloc *Accès aux éléments de l'interface* (pour afficher les paramètres des rapports)
  - *Gérer les rapports planifiés* dans le bloc *Accès aux actions* (pour créer/modifier des rapports)

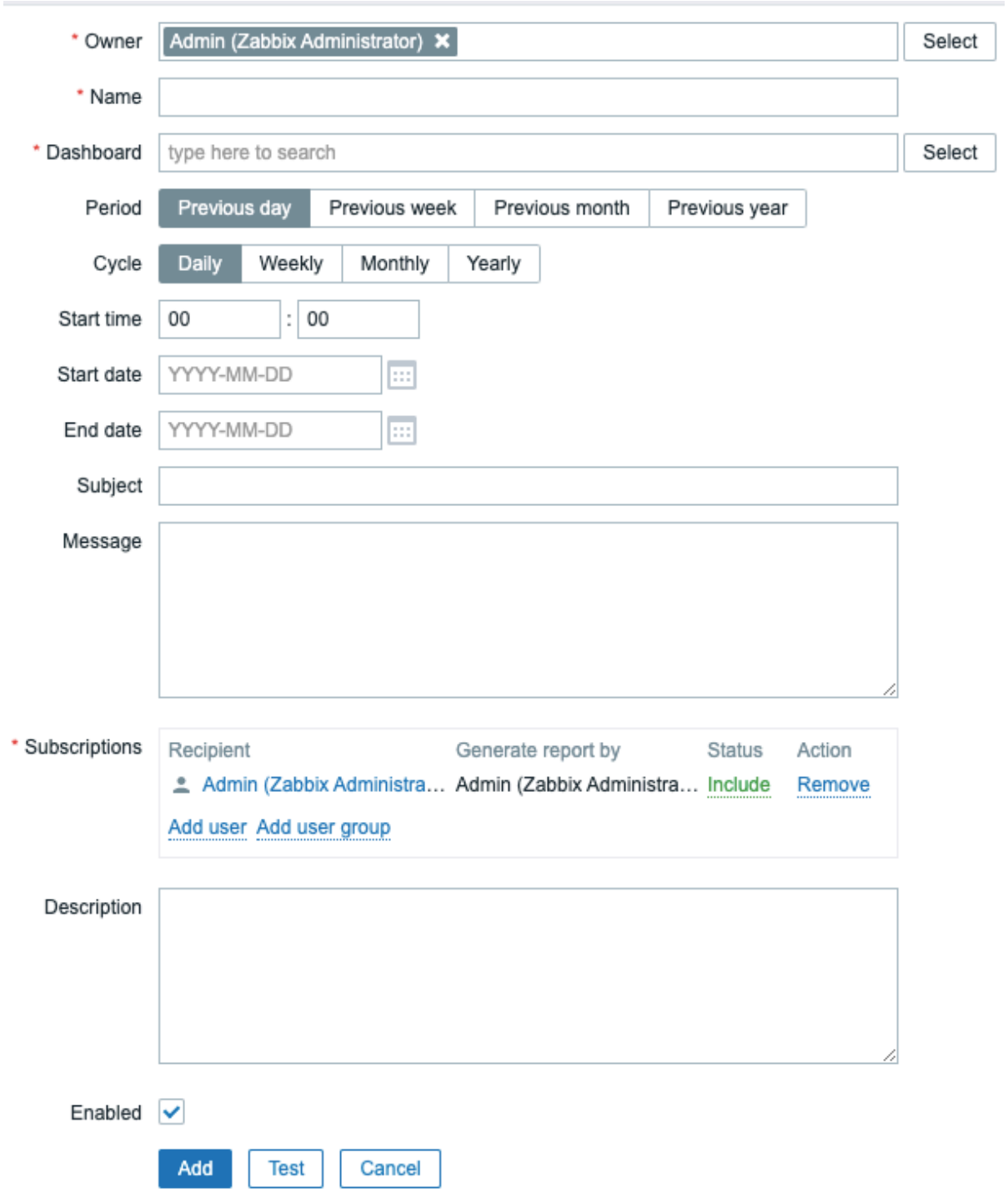
Pour créer un rapport planifié dans l'interface Zabbix, procédez comme suit :

1. Accédez à *Rapports* > *Rapports planifiés*.
2. Cliquez sur *Créer un rapport* dans le coin supérieur droit de l'écran.
3. Saisissez les paramètres du rapport dans le formulaire.

Vous pouvez également créer un rapport en ouvrant un rapport existant, en cliquant sur le bouton *Cloner*, puis en l'enregistrant sous un autre nom.

Configuration

L'onglet *Rapports planifiés* contient les attributs généraux du rapport.



Tous les champs de saisie obligatoires sont marqués d'un astérisque rouge.

Parameter	Description
Propriétaire	Utilisateur qui crée un rapport. Les utilisateurs de niveau *Super administrateur* sont autorisés à modifier le propriétaire. Pour les utilisateurs de niveau *Administrateur*, ce champ est en lecture seule.
Nom	Nom du rapport ; doit être unique.
Tableau de bord	Tableau de bord sur lequel le rapport est basé ; un seul tableau de bord peut être sélectionné à la fois. Pour sélectionner un tableau de bord, commencez à saisir son nom : une liste des tableaux de bord correspondants s'affiche ; faites défiler la liste pour effectuer votre sélection. Vous pouvez également cliquer sur *Sélectionner* à côté du champ et sélectionner un tableau de bord dans la liste affichée.
Période	Période pour laquelle le rapport sera préparé. Sélectionnez le jour, la semaine, le mois ou l'année précédente.
Cycle	Fréquence de génération du rapport. Les rapports peuvent être envoyés quotidiennement, hebdomadairement, mensuellement ou annuellement. Le mode « Hebdomadaire » permet de sélectionner les jours de la semaine auxquels le rapport sera envoyé.

Parameter	Description
Heure de début	Heure de la journée au format hh:mm à laquelle le rapport sera préparé. Notez que le fuseau horaire du serveur Zabbix sera utilisé.
Répéter le	Jours de la semaine auxquels le rapport sera envoyé. Ce champ est disponible uniquement si *Cycle* est défini sur « Hebdomadaire ».
Date de début	Date à laquelle la génération régulière des rapports doit commencer.
Date de fin	Date à laquelle la génération régulière des rapports doit s'arrêter.
Objet	Objet de l'e-mail du rapport. Macros prises en charge : {TIME}, {TIMESTAMP}.
Message	Corps de l'e-mail du rapport. Macros prises en charge : {TIME}, {TIMESTAMP}.
Abonnements	Liste des destinataires du rapport. Par défaut, elle inclut uniquement le propriétaire du rapport. Tout utilisateur Zabbix disposant d'un média e-mail configuré peut être spécifié comme destinataire du rapport. Cliquez sur *Ajouter un utilisateur* ou *Ajouter un groupe d'utilisateurs* pour ajouter d'autres destinataires. Cliquez sur le nom d'utilisateur pour modifier les paramètres : *Générer le rapport par* - indique si les données du rapport doivent être générées en fonction des permissions du tableau de bord de l'utilisateur actuel ou du destinataire. *Statut* - sélectionnez « Inclure » pour envoyer le rapport à l'utilisateur ou « Exclure » pour empêcher l'envoi du rapport à cet utilisateur. Au moins un utilisateur doit avoir le statut « Inclure ». Le statut « Exclure » peut être utilisé pour exclure des utilisateurs spécifiques d'un groupe d'utilisateurs inclus. Notez que les utilisateurs disposant de permissions insuffisantes (c'est-à-dire les utilisateurs ayant un rôle basé sur le type d'utilisateur *Administrateur* qui ne sont pas membres du même groupe d'utilisateurs que le destinataire ou le propriétaire du rapport) verront « Utilisateur inaccessible » ou « Groupe d'utilisateurs inaccessible » à la place des noms réels dans les champs *Destinataire* et *Générer le rapport par* ; les champs *Statut* et *Action* seront affichés en lecture seule.
Activé	Statut du rapport. Désélectionner cette case désactive le rapport.
Description	Description facultative du rapport. Cette description est destinée à un usage interne et ne sera pas envoyée aux destinataires du rapport.

Boutons du formulaire

Les boutons situés au bas du formulaire permettent d'effectuer plusieurs opérations.

Add	Ajouter un rapport. Ce bouton est disponible uniquement pour les nouveaux rapports.
Update	Mettre à jour les propriétés d'un rapport.
Clone	Créer un autre rapport à partir des propriétés du rapport actuel.
Test	Tester si la configuration du rapport est correcte en envoyant un rapport à l'utilisateur actuel.
Delete	Supprimer le rapport.
Cancel	Annuler la modification des propriétés du rapport.

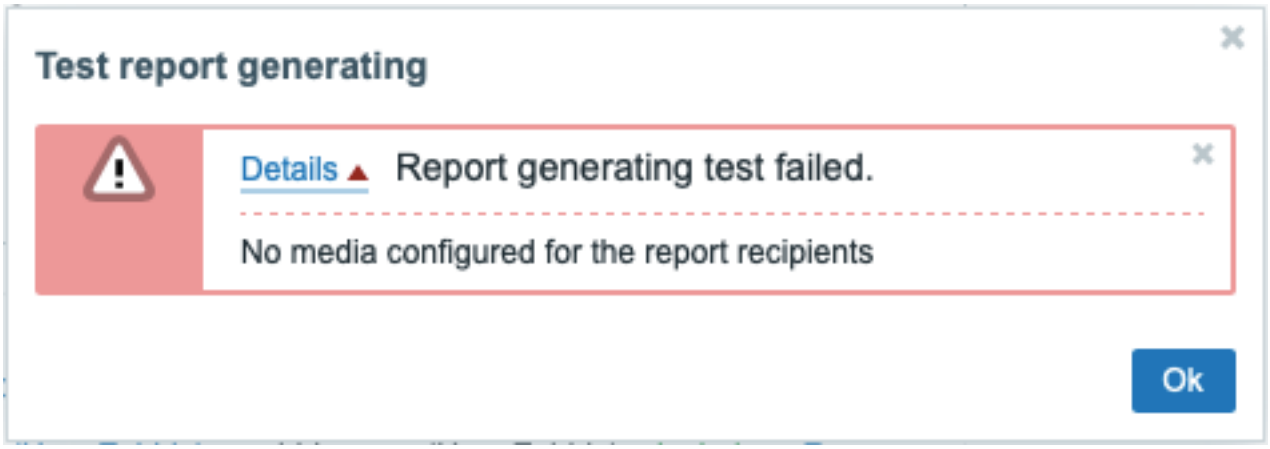
Test

Pour tester un rapport, cliquez sur le bouton *Test* en bas du formulaire de configuration du rapport.

> **Note:**
> Le bouton *Test* n'est pas disponible si le formulaire de configuration du rapport a été ouvert depuis le menu d'actions du tableau de bord.

Si la configuration est correcte, le rapport de test est immédiatement envoyé à l'utilisateur actuel. Pour les rapports de test, les paramètres des abonnés et de l'utilisateur *Générer le rapport par* sont ignorés.

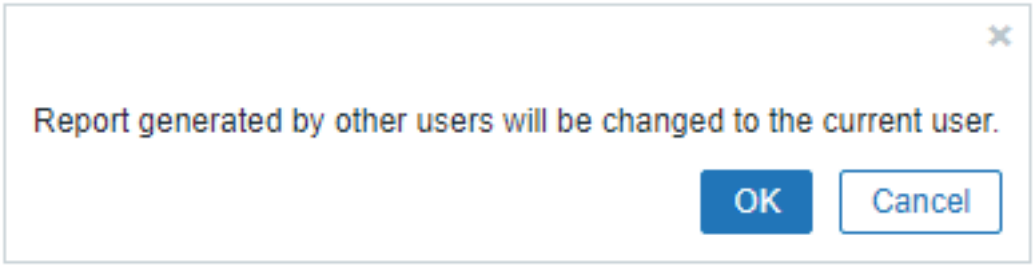
Si la configuration est incorrecte, un message d'erreur décrivant la cause possible s'affiche.



Mise à jour d'un rapport

Pour mettre à jour un rapport existant, cliquez sur le nom du rapport, effectuez les modifications de configuration nécessaires, puis cliquez sur le bouton *Update*.

Si un rapport existant est mis à jour par un autre utilisateur et que cet utilisateur modifie le tableau de bord, un message d'avertissement « Le rapport généré par d'autres utilisateurs sera remplacé par l'utilisateur actuel » s'affichera lorsque vous cliquerez sur le bouton *Update*.



Cliquer sur *OK* à cette étape entraînera les modifications suivantes :

- Les paramètres *Generate report by* seront mis à jour pour afficher l'utilisateur qui a modifié le rapport en dernier (sauf si *Generate report by* est défini sur le destinataire).
- Les utilisateurs affichés comme « Inaccessible user » ou « Inaccessible user group » seront supprimés de la liste des abonnés au rapport.

Cliquer sur *Cancel* fermera le formulaire de configuration et annulera la mise à jour du rapport.

Clonage d'un rapport

Pour cloner rapidement un rapport existant, cliquez sur le bouton *Cloner* en bas du formulaire de configuration d'un rapport existant. Lors du clonage d'un rapport créé par un autre utilisateur, l'utilisateur actuel devient le propriétaire du nouveau rapport.

Les paramètres du rapport seront copiés dans le nouveau formulaire de configuration du rapport en tenant compte des autorisations de l'utilisateur :

- Si l'utilisateur qui clone un rapport n'a pas les autorisations pour un tableau de bord, le champ *Dashboard* sera vidé.
- Si l'utilisateur qui clone un rapport n'a pas les autorisations pour certains utilisateurs ou groupes d'utilisateurs dans la liste *Subscriptions*, les destinataires inaccessibles ne seront pas clonés.
- Les paramètres *Generate report by* seront mis à jour pour afficher l'utilisateur actuel (sauf si *Generate report by* est défini sur le destinataire).

Modifiez les paramètres requis et le nom du rapport, puis cliquez sur *Ajouter*.

Exportation des données

Vue d'ensemble

Zabbix prend en charge l'exportation des données en temps réel de deux manières :

- exportation vers des fichiers
- diffusion en continu vers des systèmes externes

Les entités suivantes peuvent être exportées :

- événements de déclencheur
- valeurs d'élément
- tendances (exportation vers des fichiers uniquement)

Export vers des fichiers

Vue d'ensemble

Il est possible de configurer l'export en temps réel des événements de déclencheur, des valeurs d'élément et des tendances dans un format JSON délimité par des sauts de ligne.

L'export est effectué dans des fichiers, où chaque ligne du fichier d'export est un objet JSON. Les correspondances de valeurs ne sont pas appliquées.

En cas d'erreurs (les données ne peuvent pas être écrites dans le fichier d'export, ou le fichier d'export ne peut pas être renommé, ou un nouveau fichier ne peut pas être créé après son renommage), l'élément de données est ignoré et n'est jamais écrit dans le fichier d'export. Il est écrit uniquement dans la base de données Zabbix. L'écriture des données dans le fichier d'export reprend lorsque le problème d'écriture est résolu.

Le fichier d'export est créé avec des permissions de lecture et d'écriture uniquement pour le propriétaire du fichier. De plus, le fichier est lisible par le groupe du propriétaire. Toutes les autres permissions sont refusées.

Notez que l'hôte/l'élément peut ne contenir aucune métadonnée (groupes d'hôtes, nom d'hôte, nom de l'élément) si l'hôte/l'élément a été supprimé après la réception des données, mais avant que le serveur n'ait exporté les données.

Configuration

L'exportation en temps réel des événements de déclencheurs, des valeurs d'éléments et des tendances est configurée en spécifiant un répertoire pour les fichiers d'exportation. Consultez le paramètre ExportDir dans la configuration du serveur.

Deux autres paramètres sont disponibles :

- `ExportFileSize` peut être utilisé pour définir la taille maximale autorisée d'un fichier d'exportation individuel. Lorsqu'un processus doit écrire dans un fichier, il vérifie d'abord la taille de celui-ci. Si elle dépasse la limite configurée, le fichier est renommé en ajoutant `.old` à son nom, puis un nouveau fichier portant le nom d'origine est créé.

Attention:
Un fichier sera créé pour chaque processus qui écrira des données (soit environ 4 à 30 fichiers). Comme la taille par défaut de chaque fichier d'exportation est de 1G, la conservation de fichiers d'exportation volumineux peut rapidement épuiser l'espace disque.

- `ExportType` permet de spécifier les types d'entités (événements, historique, tendances) qui seront exportés.

Protocole d'exportation JSON délimité par des sauts de ligne

Cette section présente les détails du protocole d'exportation dans un format JSON délimité par des sauts de ligne, utilisé dans :

- exportation des données vers des fichiers
- diffusion vers des systèmes externes

Tous les fichiers ont une extension `.ndjson`. Chaque ligne du fichier d'exportation est un objet JSON.

Exportation d'un événement de déclencheur

Les informations suivantes sont exportées pour un événement de problème :

Field	Type	Description
clock	number	Nombre de secondes écoulées depuis l'époque Unix jusqu'au moment où le problème a été détecté (partie entière).
ns	number	Nombre de nanosecondes à ajouter à `clock` pour obtenir l'heure précise de détection du problème.
value	number	1 (toujours).
eventid	number	ID de l'événement de problème.
name	string	Nom de l'événement de problème.
severity	number	Niveau de gravité de l'événement de problème (0 - Non classé, 1 - Information, 2 - Avertissement, 3 - Moyen, 4 - Élevé, 5 - Désastre).
hosts	array	Liste des hôtes impliqués dans l'expression du déclencheur ; il doit y avoir au moins un élément dans le tableau.

Field		Type	Description
-		object	
	host	string	Nom de l'hôte.
	name	string	Nom visible de l'hôte.
groups		array	Liste des groupes d'hôtes de tous les hôtes impliqués dans l'expression du déclencheur ; il doit y avoir au moins un élément dans le tableau.
-		string	Nom du groupe d'hôtes.
tags		array	Liste des tags du problème (peut être vide).
-		object	
	tag	string	Nom du tag.
	value	string	Valeur du tag (peut être vide).

Les informations suivantes sont exportées pour un événement de récupération :

Field	Type	Description
clock	number	Nombre de secondes écoulées depuis l'époque Unix jusqu'au moment où le problème a été résolu (partie entière).
ns	number	Nombre de nanosecondes à ajouter à `clock` pour obtenir l'heure précise de résolution du problème.
value	number	0 (toujours).
eventid	number	ID de l'événement de récupération.
p_eventid	number	ID de l'événement de problème.

Exemples d'export d'événement de déclencheur

Problème :

```
{"clock":1519304285,"ns":123456789,"value":1,"name":"Either Zabbix agent is unreachable on Host B or polle
```

Rétablissement :

```
{"clock":1519304345,"ns":987654321,"value":0,"eventid":43,"p_eventid":42}
```

Problème (génération de plusieurs événements de problème) :

```
{"clock":1519304286,"ns":123456789,"value":1,"eventid":43,"name":"Either Zabbix agent is unreachable on Ho

{"clock":1519304286,"ns":123456789,"value":1,"eventid":43,"name":"Either Zabbix agent is unreachable on Ho
```

Rétablissement :

```
{"clock":1519304346,"ns":987654321,"value":0,"eventid":44,"p_eventid":43}

{"clock":1519304346,"ns":987654321,"value":0,"eventid":44,"p_eventid":42}
```

Export de la valeur d'élément

Les informations suivantes sont exportées pour une valeur d'élément collectée :

Field		Type	Description
host		object	Nom d'hôte de l'hôte de l'élément.
	host	string	Nom d'hôte.
	name	string	Nom visible de l'hôte.
groups		array	Liste des groupes d'hôtes de l'hôte de l'élément ; il doit y avoir au moins un élément dans le tableau.
	-	string	Nom du groupe d'hôtes.
item_tags		array	Liste des tags de l'élément (peut être vide).
	-	object	
	tag	string	Nom du tag.
	value	string	Valeur du tag (peut être vide).
itemid		number	ID de l'élément.
name		string	Nom visible de l'élément.

Field	Type	Description
clock	number	Nombre de secondes écoulées depuis l'Epoch au moment où la valeur a été collectée (partie entière).
ns	number	Nombre de nanosecondes à ajouter à `clock` pour obtenir une heure de collecte de valeur précise.
timestamp (*Log* only)	number	0 si non disponible.
source (*Log* only)	string	Chaîne vide si non disponible.
severity (*Log* only)	number	0 si non disponible.
eventid (*Log* only)	number	0 si non disponible.
value	number (for numeric items) or string (for text items)	Valeur de l'élément collectée.
type	number	Type de valeur collectée : 0 - flottant numérique, 1 - caractère, 2 - log, 3 - entier non signé, 4 - texte, 5 - binaire, 6 - JSON

Exemple d'export de valeur d'élément

Valeur numérique (entier non signé) :

```
{"host":{"host":"Host B","name":"Host B visible"},"groups":["Group X","Group Y","Group Z"],"item_tags":[{"
```

Valeur numérique (flottante) :

```
{"host":{"host":"Host B","name":"Host B visible"},"groups":["Group X","Group Y","Group Z"],"item_tags":[{"
```

Valeur de caractère, texte :

```
{"host":{"host":"Host B","name":"Host B visible"},"groups":["Group X","Group Y","Group Z"],"item_tags":[{"
```

Valeur de journal :

```
{"host":{"host":"Host A","name":"Host A visible"},"groups":["Group X","Group Y","Group Z"],"item_tags":[{"
```

Export des tendances

Les informations suivantes sont exportées pour une valeur de tendance calculée :

Field	Type	Description
host	object	Nom de l'hôte de l'élément.
host	string	Nom de l'hôte.
name	string	Nom visible de l'hôte.
groups	array	Liste des groupes d'hôtes de l'hôte de l'élément ; il doit y avoir au moins un élément dans le tableau.
-	string	Nom du groupe d'hôtes.
item_tags	array	Liste des balises de l'élément (peut être vide).
-	object	
tag	string	Nom de la balise.
value	string	Valeur de la balise (peut être vide).
itemid	number	ID de l'élément.
name	string	Nom visible de l'élément.
clock	number	Nombre de secondes écoulées depuis l'époque Unix au moment où la valeur a été collectée (partie entière).
count	number	Nombre de valeurs collectées pour une heure donnée.
min	number	Valeur minimale de l'élément pour une heure donnée.
avg	number	Valeur moyenne de l'élément pour une heure donnée.
max	number	Valeur maximale de l'élément pour une heure donnée.

Field	Type	Description
type	number	Type de valeur : 0 - flottant numérique, 3 - entier non signé

Exemples d'exportation des tendances

Valeur numérique (entier non signé) :


```
{"host":{"host":"Host B","name":"Host B visible"},"groups":["Group X","Group Y","Group Z"],"item_tags":[{"
```


Valeur numérique (flottante) :


```
{"host":{"host":"Host B","name":"Host B visible"},"groups":["Group X","Group Y","Group Z"],"item_tags":[{"
```


Diffusion vers des systèmes externes

Présentation générale

Il est possible de diffuser les valeurs des éléments et les événements de Zabbix vers des systèmes externes via HTTP (voir les détails du protocole).

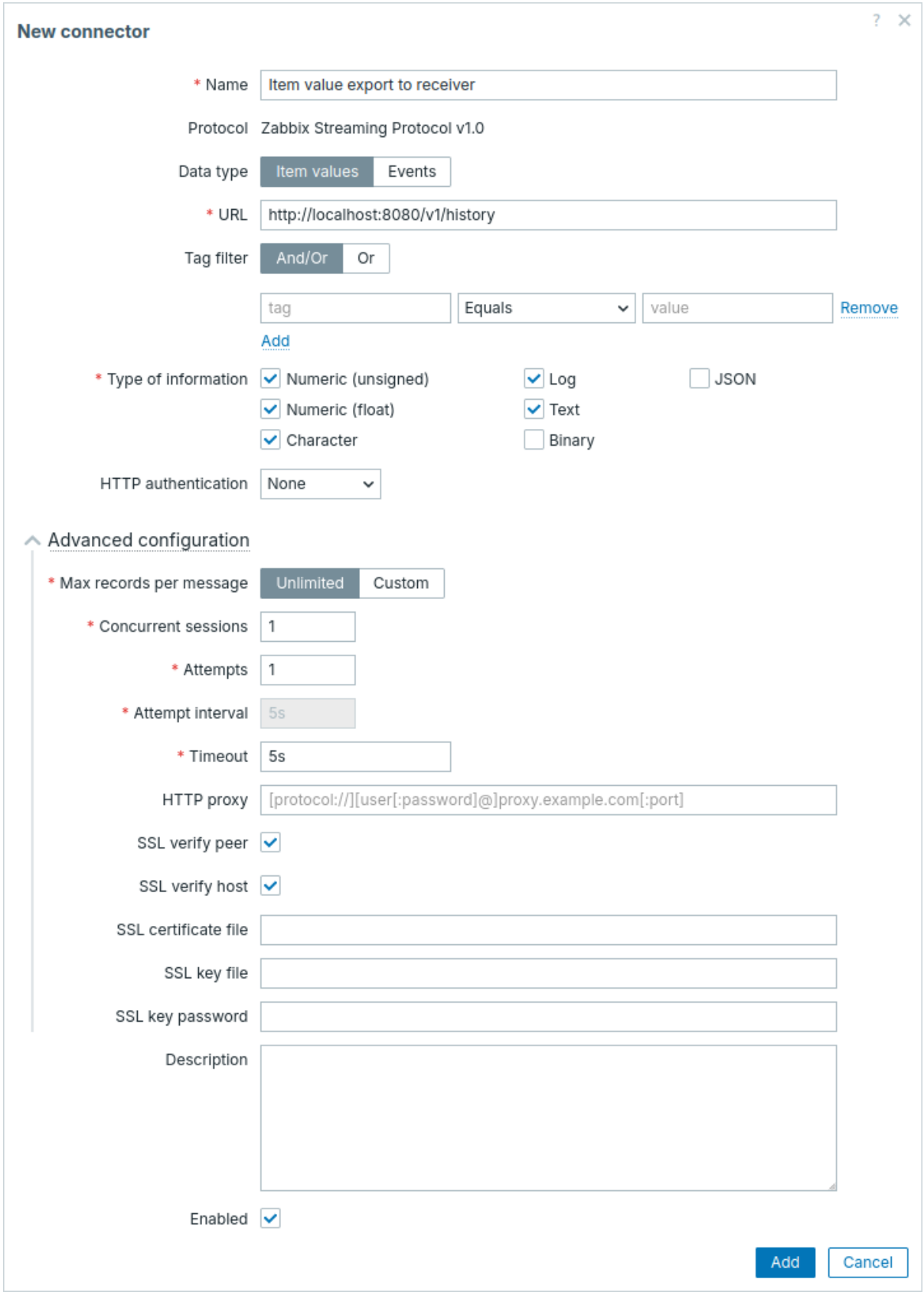
Le filtre par tags peut être utilisé pour diffuser des sous-ensembles de valeurs d'éléments ou d'événements.

Deux types de processus du serveur Zabbix sont responsables de la diffusion des données : `connector manager` et `connector worker`. Un élément interne Zabbix `zabbix[connector_queue]` vous permet de surveiller le nombre de valeurs mises en file d'attente dans la file d'attente du connecteur.

Configuration

Les étapes suivantes sont nécessaires pour configurer la diffusion des données vers un système externe :

1. Disposer d'un système distant configuré pour recevoir les données de Zabbix. À cette fin, les outils suivants sont disponibles :
   - Un exemple de récepteur simple qui consigne les informations reçues dans les fichiers `events.ndjson` et `history.ndjson`.
   - Connecteur Kafka pour le serveur Zabbix - un serveur léger écrit en Go, conçu pour transférer les valeurs des éléments et les événements d'un serveur Zabbix vers un broker Kafka.
2. Définir le nombre requis de workers de connecteur dans Zabbix en ajustant le paramètre `StartConnectors` dans `zabbix_server.conf`. Le nombre de workers de connecteur doit correspondre au nombre de connecteurs configuré dans l'interface Zabbix (ou lui être supérieur si le nombre de sessions simultanées est supérieur à 1). Redémarrer ensuite le serveur Zabbix.
3. Configurer un nouveau connecteur dans l'interface Zabbix (*Administration* > *Général* > *Connecteurs*) et recharger le cache du serveur à l'aide de la commande `zabbix_server -R config_cache_reload`.



Les champs obligatoires sont signalés par un astérisque.

Paramètre	Description
Nom	Saisir le nom du connecteur.
Type de données	Sélectionner le type de données à diffuser : **Valeurs des éléments** - diffuser les valeurs des éléments de Zabbix vers des systèmes externes ; **Événements** - diffuser les événements de Zabbix vers des systèmes externes.
URL	Saisir l'URL du récepteur. Les macros utilisateur sont prises en charge.

Paramètre	Description
Filtre de tags	Exporter uniquement les valeurs des éléments ou les événements correspondant au filtre de tags. Si ce champ n'est pas défini, tout est exporté. Il est possible d'inclure ou d'exclure des tags et des valeurs de tags spécifiques. Plusieurs conditions peuvent être définies. La correspondance des noms de tags est toujours sensible à la casse. Plusieurs opérateurs sont disponibles pour chaque condition : **Existe** - inclure les noms de tags spécifiés ; **Est égal à** - inclure les noms et les valeurs de tags spécifiés (sensible à la casse) ; **Contient** - inclure les noms de tags spécifiés lorsque les valeurs des tags contiennent la chaîne saisie (correspondance de sous-chaîne, insensible à la casse) ; **N'existe pas** - exclure les noms de tags spécifiés ; **N'est pas égal à** - exclure les noms et les valeurs de tags spécifiés (sensible à la casse) ; **Ne contient pas** - exclure les noms de tags spécifiés lorsque les valeurs des tags contiennent la chaîne saisie (correspondance de sous-chaîne, insensible à la casse). Deux types de calcul sont disponibles pour les conditions : **Et/Ou** - toutes les conditions doivent être remplies ; les conditions ayant le même nom de tag sont regroupées avec la condition Ou ; **Ou** - une seule condition remplie suffit.
Type d'information	Sélectionner le type d'information (numérique (non signé), numérique (flottant), caractère, etc.) utilisé pour filtrer les valeurs des éléments que le connecteur doit diffuser. Ce champ est disponible si *Type de données* est défini sur « Valeurs des éléments ».
Authentification HTTP	Sélectionner l'option d'authentification : **Aucune** - aucune authentification n'est utilisée ; **Basic** - l'authentification Basic est utilisée ; **NTLM** - l'authentification NTLM (Windows NT LAN Manager) est utilisée ; **Kerberos** - l'authentification Kerberos est utilisée (voir également : Configuration de Kerberos avec Zabbix) ; **Digest** - l'authentification Digest est utilisée ; **Bearer** - l'authentification Bearer est utilisée.
Nom d'utilisateur	Saisir le nom d'utilisateur (255 caractères maximum). Les macros utilisateur sont prises en charge. Ce champ est disponible si *Authentification HTTP* est défini sur « Basic », « NTLM », « Kerberos » ou « Digest ».
Mot de passe	Saisir le mot de passe utilisateur (255 caractères maximum). Les macros utilisateur sont prises en charge. Ce champ est disponible si *Authentification HTTP* est défini sur « Basic », « NTLM », « Kerberos » ou « Digest ».
Jeton Bearer	Saisir le jeton Bearer. Les macros utilisateur sont prises en charge. Ce champ est disponible et obligatoire si *Authentification HTTP* est défini sur « Bearer ».
Configuration avancée	Cliquer sur l'en-tête *Configuration avancée* pour afficher les options de configuration avancées (voir ci-dessous).
Nombre maximal d'enregistrements par message	Indiquer le nombre maximal de valeurs ou d'événements pouvant être diffusés dans un même message.
Sessions simultanées	Sélectionner le nombre de processus d'envoi à exécuter pour ce connecteur. Jusqu'à 100 sessions peuvent être spécifiées ; la valeur par défaut est « 1 ».
Tentatives	Nombre de tentatives de diffusion des données. Jusqu'à 5 tentatives peuvent être spécifiées ; la valeur par défaut est « 1 ».
Intervalle entre les tentatives	Indiquer le délai d'attente du connecteur après une tentative infructueuse de diffusion des données. Jusqu'à 10 s peuvent être spécifiées ; la valeur par défaut est « 5 s ». Ce champ est disponible si *Tentatives* est défini sur « 2 » ou plus. Les tentatives infructueuses sont celles pour lesquelles l'établissement d'une connexion a échoué ou pour lesquelles le code de réponse HTTP n'est pas 200, 201, 202, 203 ou 204. Les nouvelles tentatives sont **déclenchées** en cas d'erreurs de communication ou lorsque le code de réponse HTTP n'est pas 200, 201, 202, 203, 204, 400, 401, 403, 404, 405, 415 ou 422. Les redirections sont suivies : 302 -> 200 est donc une réponse positive, tandis que 302 -> 503 déclenchera une nouvelle tentative.
Délai d'expiration	Indiquer le délai d'expiration du message (1 à 60 secondes, valeur par défaut : 5 secondes). Les suffixes de temps sont pris en charge (par exemple, `30s`, `1m`). Les macros utilisateur sont prises en charge.

Paramètre	Description
Proxy HTTP	Vous pouvez spécifier un proxy HTTP au format suivant : `[protocol://][username[:password]@]proxy.example.com[:port]` Les macros utilisateur sont prises en charge. Le préfixe facultatif `protocol://` peut être utilisé pour spécifier d'autres protocoles de proxy (la prise en charge du préfixe de protocole a été ajoutée dans cURL 7.21.7). Si aucun protocole n'est spécifié, le proxy est traité comme un proxy HTTP. Le port 1080 est utilisé par défaut. Si *Proxy HTTP* est spécifié, le proxy remplace les variables d'environnement associées au proxy, telles que `http_proxy` et `HTTPS_PROXY`. S'il n'est pas spécifié, le proxy ne remplace pas les variables d'environnement associées au proxy. La valeur saisie est transmise telle quelle, sans aucune vérification de cohérence. Vous pouvez également saisir l'adresse d'un proxy SOCKS. Si vous spécifiez un protocole incorrect, le connecteur ne pourra pas diffuser les valeurs des éléments ou les événements de Zabbix. Notez que seule l'authentification simple est prise en charge avec un proxy HTTP.
Vérifier le pair SSL	Cocher la case pour vérifier le certificat SSL du serveur web. Le certificat du serveur est vérifié par rapport aux certificats des autorités de certification (CA) approuvées situés dans l'emplacement système des CA. Vous pouvez remplacer l'emplacement des fichiers CA à l'aide du paramètre de configuration du serveur Zabbix ou du proxy `SSLCALocation`.
Vérifier l'hôte SSL	Cocher la case pour vérifier que le champ *Common Name* ou le champ *Subject Alternate Name* du certificat du serveur web correspond. Cette option définit l'option cURL `CURLOPT_SSL_VERIFYHOST`.
Fichier de certificat SSL	Nom du fichier de certificat SSL utilisé pour l'authentification du client. Le fichier de certificat doit être au format PEM[1]. Les macros utilisateur sont prises en charge. Si le fichier de certificat contient également la clé privée, laisser le champ *Fichier de clé SSL* vide. Si la clé est chiffrée, spécifier le mot de passe dans le champ *Mot de passe de la clé SSL*. Le répertoire contenant ce fichier est spécifié par le paramètre de configuration du serveur Zabbix ou du proxy `SSLCertLocation`.
Fichier de clé SSL	Nom du fichier de clé privée SSL utilisé pour l'authentification du client. Le fichier de clé privée doit être au format PEM[1]. Les macros utilisateur sont prises en charge. Le répertoire contenant ce fichier est spécifié par le paramètre de configuration du serveur Zabbix ou du proxy `SSLKeyLocation`.
Mot de passe de la clé SSL	Mot de passe du fichier de clé privée SSL. Les macros utilisateur sont prises en charge.
Description	Saisir la description du connecteur.
Activé	Cocher la case pour activer le connecteur.

Lorsque le connecteur Kafka est configuré avec une liste d'adresses de brokers d'amorçage séparées par des virgules (par exemple, `Kafka.URL=kafka1.example.com:9093,kafka2.example.com:9093`), le client Kafka se connecte au(x) broker(s) qui répond(ent) en premier et utilise leurs métadonnées de cluster. Si la liste contient des adresses provenant de différents clusters Kafka, seul le cluster répondant le plus rapidement sera utilisé et les autres adresses seront consignées comme indisponibles ; par conséquent, des avertissements au démarrage tels que le suivant peuvent apparaître même si le connecteur est connecté :

```
kafka cluster connected, but broker(s) "kafka1.example.com:9093, kafka2.example.com:9093" unavailable; wil
```

Dans certains environnements (réseaux privés, réseaux de conteneurs ou configurations DNS/hosts non standard), les noms d'hôte ou les adresses IP peuvent être résolus en adresses de bouclage (par exemple, `127.0.0.1/localhost`) ou normalisés par le client, ce qui peut rendre ces avertissements trompeurs. Pour réduire la confusion, assurez-vous que toutes les adresses `Kafka.URL` appartiennent au même cluster Kafka, vérifiez la résolution DNS depuis l'hôte du connecteur ainsi que les `advertised.listeners` des brokers, et privilégiez les adresses qui se résolvent vers l'adresse annoncée du broker.

Protocole

La communication entre le serveur et le receiver s'effectue via HTTP en utilisant l'API REST, NDJSON, "Content-Type: application/x-ndjson".

Pour plus de détails, consultez Newline-delimited JSON export protocol.

Requête du serveur

Exemple de diffusion en continu des valeurs d'élément :

```
POST /v1/history HTTP/1.1
Host: localhost:8080
Accept: */*
Accept-Encoding: deflate, gzip, br, zstd
Content-Length: 628
Content-Type: application/x-ndjson

{"host":{"host":"Zabbix server","name":"Zabbix server"},"groups":["Zabbix servers"],"item_tags":[{"tag":"f
{"host":{"host":"Zabbix server","name":"Zabbix server"},"groups":["Zabbix servers"],"item_tags":[{"tag":"f
{"host":{"host":"Zabbix server","name":"Zabbix server"},"groups":["Zabbix servers"],"item_tags":[{"tag":"b
```

Exemple de diffusion en continu des événements :

```
POST /v1/events HTTP/1.1
Host: localhost:8080
Accept: */*
Accept-Encoding: deflate, gzip, br, zstd
Content-Length: 333
Content-Type: application/x-ndjson

{"clock":1673454303,"ns":800155804,"value":1,"eventid":5,"name":"trigger for foo being 0","severity":0,"ho
{"clock":1673454303,"ns":832290669,"value":0,"eventid":6,"p_eventid":5}
```

Réponse du récepteur

La réponse se compose du code d'état de la réponse HTTP et de la charge utile JSON. Le code d'état de la réponse HTTP doit être "200", "201", "202", "203" ou "204" pour les requêtes traitées avec succès, sinon il indique un échec de la requête.

Exemple de réussite :

```
HTTP/1.1 200 OK
Content-Type: application/json
X-Content-Type-Options: nosniff
Date: Tue, 21 Apr 2026 10:13:04 GMT
Content-Length: 23

{"response":"success"}
```

Exemple avec des erreurs :

```
HTTP/1.1 422 Unprocessable Entity
Content-Type: application/json
X-Content-Type-Options: nosniff
Date: Tue, 21 Apr 2026 12:15:01 GMT
Content-Length: 55

{"error":"invalid character '{' after top-level value"}
```

Passerelle SNMP

Aperçu

La passerelle SNMP de Zabbix est une extension AgentX pour snmpd prenant en charge à la fois l'interrogation SNMP et les traps SNMP.

Avec la passerelle SNMP de Zabbix, il est possible d'utiliser le protocole SNMP pour récupérer :

- les données des déclencheurs ;
- les données des déclencheurs de problème ;
- l'état des groupes d'hôtes (nombre de déclencheurs par état de déclencheur pour chaque groupe)

Les données sont récupérées par l'OID, qui est une combinaison d'une base commune et d'un suffixe spécifique. La **base** commune est définie dans le fichier de configuration de la passerelle SNMP, par exemple :

- `BaseOID=1.3.6.1.4.1.3043.7.55` - pour toutes les données de déclencheur ;
- `ProblemBaseOID=1.3.6.1.4.1.3047.7.55` - pour les données des déclencheurs de problème ;
- `BaseOID=1.3.6.1.4.1.3046.7.55` - pour l'état des groupes d'hôtes.

Le **suffixe** OID est défini dans la configuration des déclencheurs d'hôte en tant que **tag** (par exemple, `OIDSuffix:3`) dans le frontend.

Dans ce cas, toutes les informations du déclencheur seront disponibles sous `OID=1.3.6.1.4.1.3043.7.55.X.3`. Ici, « X » correspondra au numéro des champs de données du déclencheur (c'est-à-dire 1 - suffixe, 2 - ID, 3 - expression, 4 - description, etc.).

Pour une description plus détaillée et un exemple de fichier de configuration, consultez le fichier readme de la passerelle SNMP.

Installation et configuration

Consultez le fichier readme de la passerelle SNMP pour obtenir des instructions sur :

- l'installation et la configuration de snmpd ;
- l'activation de la prise en charge d'AgentX ;
- la configuration de la passerelle SNMP Zabbix ;
- la configuration des traps SNMP pour les changements d'état des déclencheurs.

Récupération des données

Une fois que tout est correctement configuré, vous pouvez utiliser les commandes `snmpwalk` et `snmpget` pour récupérer des données :

```
[user@localhost ~]# snmpget -v2c -c public 127.0.0.1 1.3.6.1.4.1.3043.7.55.2.3
SNMPv2-SMI::enterprises.3043.7.55.2.3 = INTEGER: 15247
```

```
[user@localhost ~]# snmpwalk -v2c -c public 127.0.0.1 1.3.6.1.4.1.3043.7.55
SNMPv2-SMI::enterprises.3043.7.55.1.1 = INTEGER: 1
SNMPv2-SMI::enterprises.3043.7.55.1.3 = INTEGER: 3
SNMPv2-SMI::enterprises.3043.7.55.1.4 = INTEGER: 4
SNMPv2-SMI::enterprises.3043.7.55.1.5 = INTEGER: 5
SNMPv2-SMI::enterprises.3043.7.55.1.6 = INTEGER: 6
SNMPv2-SMI::enterprises.3043.7.55.1.10 = INTEGER: 10
SNMPv2-SMI::enterprises.3043.7.55.2.1 = INTEGER: 15367
SNMPv2-SMI::enterprises.3043.7.55.2.3 = INTEGER: 15247
SNMPv2-SMI::enterprises.3043.7.55.2.4 = INTEGER: 15365
SNMPv2-SMI::enterprises.3043.7.55.2.5 = INTEGER: 15366
SNMPv2-SMI::enterprises.3043.7.55.2.6 = INTEGER: 13493
SNMPv2-SMI::enterprises.3043.7.55.2.10 = INTEGER: 13503
...
```

Options de filtrage

Vous pouvez limiter les informations sur les déclencheurs de problèmes dans la configuration de la passerelle SNMP :

- par gravité (par défaut `ProblemMinSeverity=-1`)
- en masquant les problèmes acquittés (par défaut `ProblemHideAck=false`)

Vous pouvez limiter le nombre de problèmes par groupe d'hôtes dans la configuration de la passerelle SNMP :

- par déclencheurs à l'état inconnu (par défaut `CountUnknown=false`)
- par déclencheurs avec des problèmes acquittés/non acquittés/tous les problèmes (par défaut `CountAcknowledgeStatus=all`)

Export/import de configuration

Vue d'ensemble

La fonctionnalité d'export/import de Zabbix permet d'échanger diverses entités de configuration entre un système Zabbix et un autre.

Cas d'utilisation typiques de cette fonctionnalité :

- partager des modèles ou des cartes réseau - les utilisateurs de Zabbix peuvent partager leurs paramètres de configuration
- téléverser un modèle vers Zabbix Community templates. Ensuite, d'autres peuvent télécharger le modèle et importer le fichier dans Zabbix.
- s'intégrer à des outils tiers - les formats universels YAML, XML et JSON rendent possible l'intégration ainsi que l'import/export de données avec des outils et applications tiers

Ce qui peut être exporté/importé

Les objets pouvant être exportés/importés sont :

- Groupes d'hôtes (*via l'API Zabbix uniquement*)
- Groupes de modèles (*via l'API Zabbix uniquement*)
- Modèles
- Hôtes
- Cartes réseau
- Types de média
- Images
- Tableaux de bord
- Expressions régulières

Format d'export

Les données peuvent être exportées à l'aide de l'interface web Zabbix ou de l'API Zabbix. Les formats d'export pris en charge sont YAML, XML et JSON.

Détails sur l'export

- Tous les éléments pris en charge sont exportés dans un seul fichier.
- Les entités de hôte et de modèle (éléments, déclencheurs, graphiques, règles de découverte) héritées des modèles liés ne sont pas exportées. Toute modification apportée à ces entités au niveau d'un hôte (comme un intervalle d'élément modifié, une expression régulière modifiée ou des prototypes ajoutés à la règle de découverte de bas niveau) sera perdue lors de l'export ; lors de l'import, toutes les entités provenant des modèles liés sont recréées comme sur le modèle lié d'origine.
- Les entités créées par la découverte de bas niveau et toutes les entités qui en dépendent ne sont pas exportées. Par exemple, un déclencheur créé pour un élément généré par une règle LLD ne sera pas exporté.
- Lorsque l'hôte/le modèle exporté contient des entités prenant en charge les délais d'attente, les valeurs de délai d'attente seront exportées si ces entités ont leurs propres délais d'attente configurés.

Détails sur l'import

- L'import s'arrête à la première erreur.
- Lors de la mise à jour d'images existantes pendant l'import d'images, le champ "imagetype" est ignoré, c'est-à-dire qu'il est impossible de modifier le type d'image via l'import.
- Lors de l'import d'hôtes/modèles avec l'option "Delete missing", les macros d'hôte/modèle absentes du fichier d'import seront supprimées de l'hôte/modèle après l'import.
- Les balises vides pour les éléments, déclencheurs, graphiques, discoveryRules, itemPrototypes, triggerPrototypes, graphPrototypes n'ont aucune signification, c'est-à-dire que c'est comme si elles étaient absentes.
- Si les entités de l'hôte/modèle importé ont leurs propres délais d'attente configurés, ils seront appliqués ; sinon, les délais d'attente du proxy/global seront appliqués.
- L'import prend en charge YAML, XML et JSON, le fichier d'import doit avoir une extension de fichier correcte : .yaml et .yml pour YAML, .xml pour XML et .json pour JSON. Voir les informations de compatibilité concernant les versions XML prises en charge.
- L'import prend en charge les fichiers de configuration uniquement en encodage UTF-8 (avec ou sans BOM) ; les autres encodages (UTF16LE, UTF16BE, UTF32LE, UTF32BE, etc.) entraîneront une erreur de conversion lors de l'import.

Format de base YAML

Le format d'export YAML contient les nœuds suivants :

- Nœud racine pour l'export YAML de Zabbix
- Version d'export

```
zabbix_export:
  version: '8.0'
```

Les autres nœuds dépendent des objets exportés.

Format XML

Le format d'export XML contient les balises suivantes :

- En-tête par défaut pour les documents XML
- Balise racine pour l'export XML de Zabbix
- Version d'export

```
<?xml version="1.0" encoding="UTF-8"?>
<zabbix_export>
```

```
    <version>8.0</version>
</zabbix_export>
```

Les autres balises dépendent des objets exportés.

Format JSON

Le format d'export JSON contient les objets suivants :

- Objet racine pour l'export JSON de Zabbix
- Version d'export

```
{
    "zabbix_export": {
        "version": "8.0"
    }
}
```

Les autres objets dépendent des objets exportés.

Groupes de modèles

Vue d'ensemble

Dans l'interface, les groupes de modèles peuvent être exportés uniquement avec l'export de modèle. Lorsqu'un modèle est exporté, tous les groupes auxquels il appartient sont exportés automatiquement avec lui.

L'API permet d'exporter les groupes de modèles indépendamment des modèles.

Format d'export

```
  template_groups:
    - uuid: 36bff6c29af64692839d077febfc7079
      name: 'Network devices'
```

Éléments exportés

Element	Type	Description
uuid	string	Identifiant unique de ce groupe de modèles.
name	string	Nom du groupe.

Groupes d'hôtes

Vue d'ensemble

Dans l'interface, les groupes d'hôtes peuvent être exportés uniquement avec l'export de l'hôte. Lorsqu'un hôte est exporté, tous les groupes auxquels il appartient sont exportés automatiquement avec lui.

L'API permet d'exporter les groupes d'hôtes indépendamment des hôtes.

Format d'export

```
  host_groups:
    - uuid: 6f6799aa69e844b4b3918f779f2abf08
      name: 'Serveurs Zabbix'
```

Éléments exportés

Element	Type	Description
uuid	string	Identifiant unique pour ce groupe d'hôtes.
name	string	Nom du groupe.

Modèles

Vue d'ensemble

Les modèles sont exportés avec de nombreux objets associés et relations entre objets.

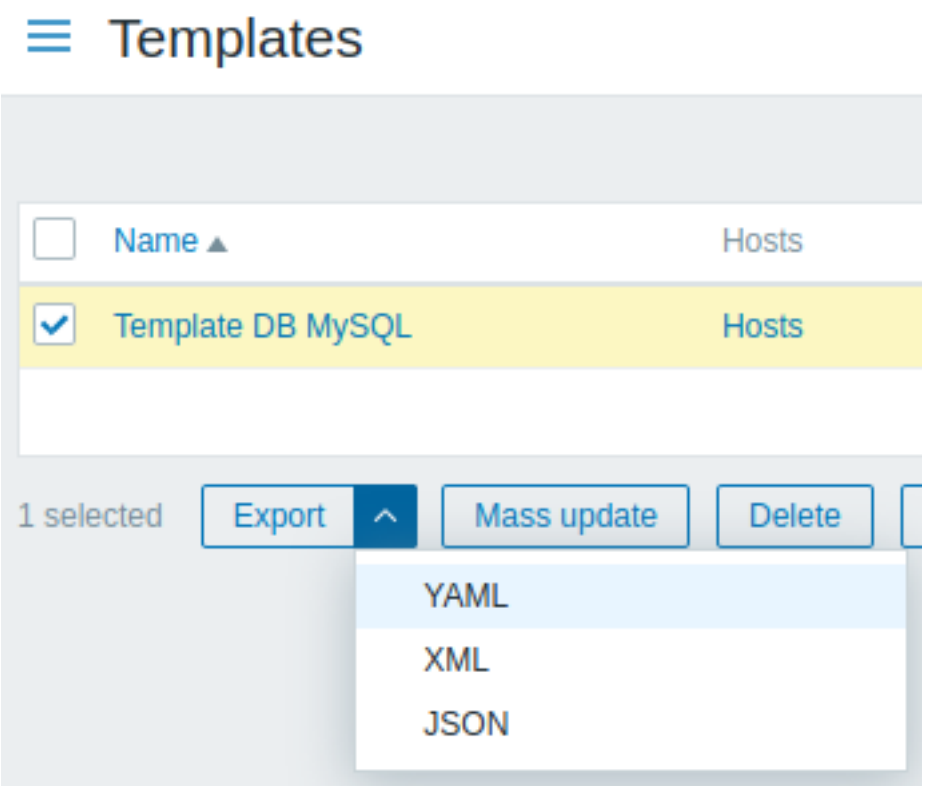
L'export du modèle contient :

- Groupes de modèles liés
- Groupes d'hôtes liés (si utilisés dans la configuration de prototype d'hôte)
- Données du modèle
- Liaison à d'autres modèles
- Liaison à des groupes de modèles
- Éléments directement liés
- Déclencheurs directement liés
- Graphiques directement liés
- Tableaux de bord directement liés
- Règles de découverte directement liées avec tous les prototypes
- Scénarios web directement liés
- Correspondances de valeurs

Exportation

Pour exporter des modèles, procédez comme suit :

1. Accédez à *Collecte de données* > *Modèles*.
2. Cochez les cases des modèles à exporter.
3. Cliquez sur *Exporter* sous la liste pour exporter au format YAML par défaut, ou cliquez sur la flèche pour sélectionner un autre format d'exportation.



Selon le format sélectionné, les modèles sont exportés vers un fichier local avec un nom par défaut :

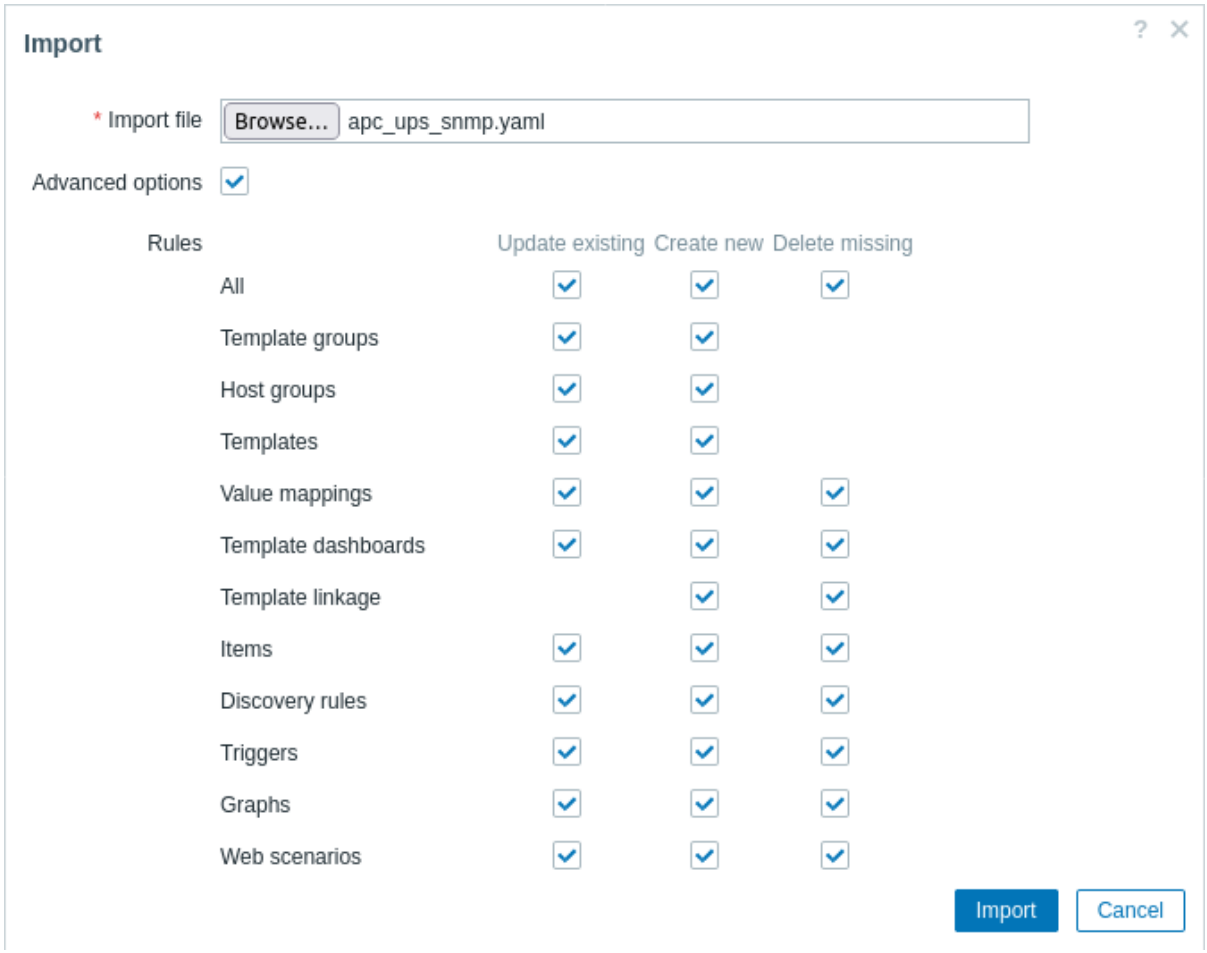
- `zabbix_export_templates.yaml` - en export YAML ;
- `zabbix_export_templates.xml` - en export XML ;
- `zabbix_export_templates.json` - en export JSON.

Importation

Pour importer des modèles, procédez comme suit :

1. Accédez à *Collecte de données* > *Modèles*.
2. Cliquez sur *Importer* dans le coin supérieur droit.

3. Sélectionnez le fichier d'importation.
4. Cochez les cases correspondant aux règles d'importation que vous souhaitez appliquer (voir ci-dessous).
5. Cliquez sur *Importer* dans le coin inférieur droit du formulaire de configuration.



Si vous cochez la case *Options avancées*, une liste détaillée de tous les éléments pouvant être importés s'affiche. Cochez ou décochez chaque règle d'importation selon vos besoins.

Si vous cliquez sur la case de la ligne *Tous*, tous les éléments situés en dessous seront cochés ou décochés.

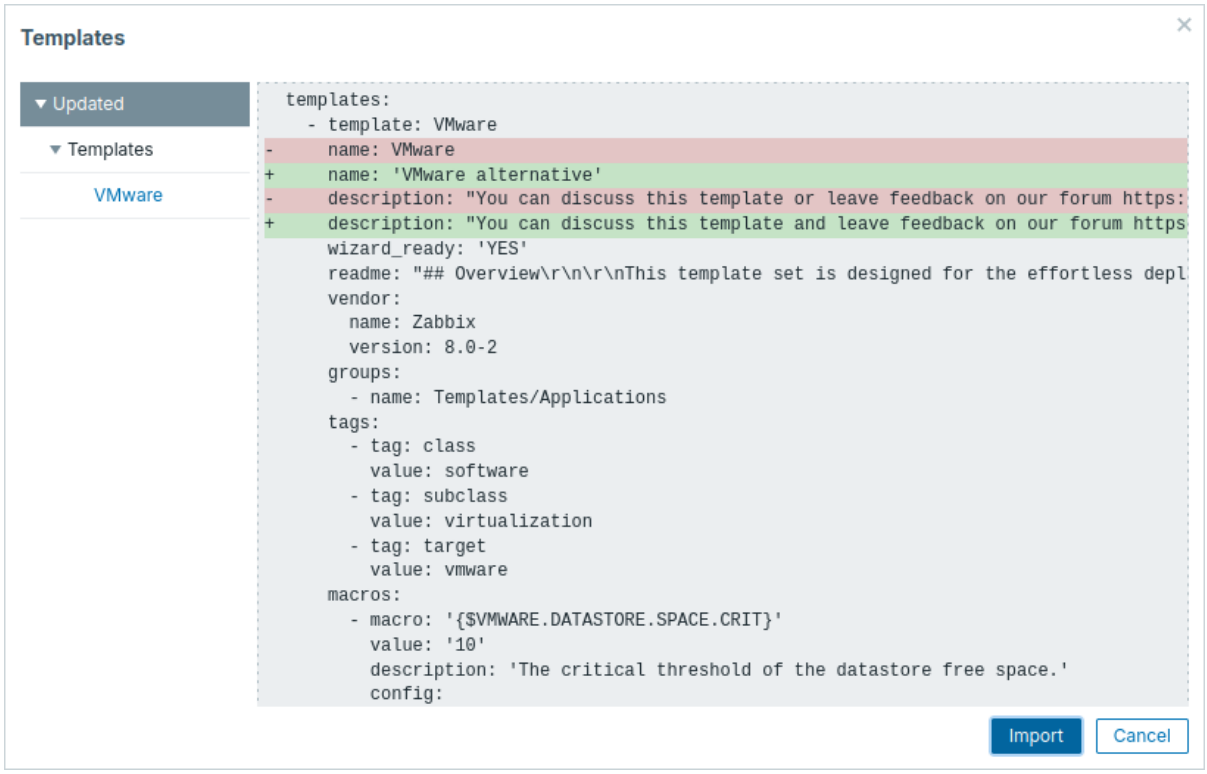
Règles d'importation :

Règle	Description
Mettre à jour les éléments existants	Les éléments existants seront mis à jour à l'aide des données du fichier d'importation. Sinon, ils ne seront pas mis à jour.
Créer les nouveaux éléments	Les nouveaux éléments seront créés à l'aide des données du fichier d'importation. Sinon, ils ne seront pas créés.
Supprimer les éléments manquants	Les éléments existants qui ne figurent pas dans le fichier d'importation seront supprimés. Sinon, ils ne seront pas supprimés. Si *Supprimer les éléments manquants* est sélectionné pour la *liaison de modèles*, les liaisons actuelles avec les modèles qui ne figurent pas dans le fichier d'importation seront supprimées. Les entités (éléments, déclencheurs, graphiques, etc.) héritées des modèles dont la liaison a été supprimée ne seront pas supprimées (sauf si l'option *Supprimer les éléments manquants* est également sélectionnée pour chaque entité).

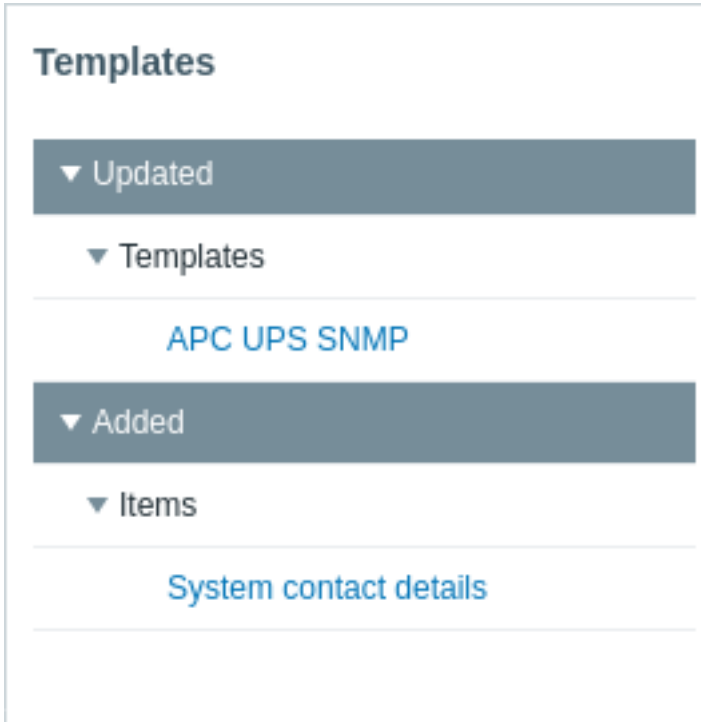
Attention:
Si des modèles portant les mêmes noms existent déjà, les options *Supprimer les éléments manquants* doivent être sélectionnées lors de l'importation afin d'obtenir une importation propre. Ainsi, les anciens éléments qui ne figurent plus dans le modèle mis à jour seront supprimés (notez que cela entraînera la perte de l'historique de ces anciens éléments).

L'écran suivant vous permet d'afficher le contenu d'un modèle en cours d'importation. S'il s'agit d'un nouveau modèle, tous les éléments seront affichés en vert. Lors de la mise à jour d'un modèle existant, les nouveaux éléments du modèle seront mis en

évidence en vert, les éléments supprimés seront mis en évidence en rouge et les éléments qui n'ont pas changé seront affichés sur un arrière-plan gris.



Le menu situé à gauche permet de parcourir la liste des modifications. La section *Mis à jour* met en évidence toutes les modifications apportées aux éléments existants du modèle. La section *Ajoutés* répertorie les nouveaux éléments du modèle. Les éléments de chaque section sont regroupés par type d'élément ; cliquez sur la flèche grise pour développer ou réduire le groupe d'éléments.



Vérifiez les modifications apportées au modèle, puis cliquez sur *Importer* pour effectuer l'importation du modèle. Un message indiquant la réussite ou l'échec de l'importation s'affichera dans l'interface.

Format d'export Lorsqu'un modèle est exporté, Zabbix génère un format YAML, JSON ou XML structuré. L'export inclut les éléments du modèle, tels que les métadonnées du modèle, les éléments, les macros, les déclencheurs, les tableaux de bord, et plus encore.

Chaque élément remplit une fonction spécifique et peut contenir des éléments imbriqués.

Les sections suivantes décrivent chaque élément dans le format d'export. Les exemples utilisent le modèle Linux by Zabbix agent.

Une ellipse (`...`) indique des éléments omis par souci de concision. La note (`voir le tableau ci-dessous`) est utilisée lorsque l'élément est expliqué plus en détail dans les sections suivantes.

```
zabbix_export:
  version: '8.0'
  template_groups:
    - uuid: 846977d1dfed4968bc5f8bdb363285bc
      name: 'Templates/Operating systems'
  templates: (see table below)
  triggers: (see table below)
  graphs: (see table below)
```

Métadonnées du modèle

Element	Type	Description
version	string	(required) Version d'export.
template_groups		(required) Élément racine pour les groupes de modèles.
uuid	string	(required) Identifiant unique pour ce groupe de modèles.
name	string	(required) Nom du groupe de modèles.
host_groups		Élément racine pour les groupes d'hôtes utilisés par les prototypes d'hôte.
uuid	string	(required) Identifiant unique pour ce groupe d'hôtes.
name	string	(required) Nom du groupe d'hôtes.
templates		Élément racine pour les modèles.
triggers		Élément racine pour les éléments de déclencheur de modèle, qui sont les mêmes que pour les déclencheurs d'éléments de modèle.
graphs		Élément racine pour les éléments de graphique de modèle, qui sont les mêmes que pour les graphiques d'hôte.

Modèles

```
zabbix_export:
  (...)
  templates:
    - uuid: f8f7908280354f2abeed07dc788c3747
      template: 'Linux by Zabbix agent'
      name: 'Linux by Zabbix agent'
      description: |
        Il s'agit d'un modèle Linux officiel. Il nécessite Zabbix agent 8.0 ou une version plus récente. (
      wizard_ready: 'YES'
      readme: |
        ## Vue d'ensemble

        Il s'agit d'un modèle Linux officiel. Il nécessite Zabbix agent 8.0 ou une version plus récente. (
      vendor:
        name: Zabbix
        version: 8.0-2
      groups:
        - name: 'Templates/Operating systems'
      items: (see table below)
      discovery_rules: (see table below)
      tags:
        - tag: class
          value: os
        - tag: target
          value: linux
      macros:
        - macro: '{$AGENT.TIMEOUT}'
          value: 3m
          description: 'Délai d''expiration après lequel l''agent est considéré comme indisponible. Fonctio
          config: (see table below)
        - macro: '{$CPU.UTIL.CRIT}'
          value: '90'
```

```
        description: 'Seuil critique d''utilisation du CPU exprimé en %.'
        config: (see table below)
      (...)
    dashboards: (see table below)
    valuemaps: (see table below)
  (...)
```

Element	Type	Description
uuid	string	(required) Identifiant unique pour ce modèle.
template	string	(required) Nom unique du modèle.
name	string	Nom visible du modèle.
description	text	Description du modèle.
wizard_ready	text	Indique si le modèle est disponible pour la sélection dans l'assistant d'hôte. Valeurs possibles:[1] NO (0, par défaut), YES (1). Voir aussi: Objet modèle (`wizard_ready`).
readme	text	Instructions de configuration spécifiques au modèle à afficher dans l'assistant d'hôte. Prend en charge le formatage Markdown.
vendor		Élément racine pour le fournisseur du modèle (présent si le modèle exporté contient des données de fournisseur).
name	string	(required) Nom du fournisseur du modèle.
version	string	(required) Version du modèle. Pour les modèles prêts à l'emploi, la version s'affiche comme suit: version majeure de Zabbix, séparateur ("-"), numéro de révision (augmenté à chaque nouvelle version du modèle et réinitialisé à chaque version majeure de Zabbix). Par exemple, 7.0-0, 7.0-3, 8.0-0, 8.0-3.
templates		Élément racine pour les modèles liés.
name	string	(required) Nom du modèle.
groups		Élément racine pour les groupes de modèles.
name	string	(required) Nom du groupe de modèles.
items		Élément racine pour les éléments de modèle.
discovery_rules		Élément racine pour les règles de découverte de bas niveau du modèle.
httptests		Élément racine pour les scénarios web du modèle.
tags		Élément racine pour les balises du modèle.
tag	string	(required) Nom de la balise.
value	string	Valeur de la balise.
macros		Élément racine pour les macros utilisateur du modèle.
macro	string	(required) Nom de la macro utilisateur.
type	string	Type de macro utilisateur. Valeurs possibles:[1] TEXT (0, par défaut), SECRET_TEXT (1), VAULT (2). Voir aussi: Objet macro utilisateur (`type`).
value	string	Valeur de la macro utilisateur.
description	string	Description de la macro utilisateur.
config		Élément racine pour la configuration des macros du modèle, responsable de la manière dont la macro est affichée dans l'assistant d'hôte.
dashboards		Élément racine pour les tableaux de bord du modèle.
valuemaps		Élément racine pour les mappages de valeurs du modèle.

Éléments du modèle

```
zabbix_export:
  (...)
  templates:
    (...)
    items:
      - uuid: f94f9f4699e94c369e6c98b2a2f485ce
        name: 'Zabbix agent ping'
        key: agent.ping
        description: 'L'agent renvoie toujours « 1 » pour cet élément. Peut être utilisé avec `nodata()`
        valuemap:
          name: 'Zabbix agent ping status'
        tags:
          - tag: component
```

```
            value: system
      (...)
      - uuid: 58818005e76d46dda14d6592f601ab00
        name: 'Nombre de paquets installés'
        key: system.sw.packages.get
        delay: 1h
        preprocessing: (voir le tableau ci-dessous)
        tags:
          - tag: component
            value: os
        triggers: (voir le tableau ci-dessous)
      - uuid: 403cebed115441369e94d35d070ca7b8
        name: 'Utilisation de la mémoire'
        type: DEPENDENT
        key: vm.memory.utilization
        value_type: FLOAT
        units: '%'
        description: 'Le pourcentage de mémoire utilisée est calculé comme `100-pavailable`.'
        preprocessing: (voir le tableau ci-dessous)
        master_item:
          key: 'vm.memory.size[pavailable]'
        tags:
          - tag: component
            value: memory
        triggers: (voir le tableau ci-dessous)
      (...)
```

Élément	Type	Description
uuid	string	(obligatoire) Identifiant unique de cet élément.
name	string	(obligatoire) Nom de l'élément.
type	string	Type d'élément. Valeurs possibles :[1] ZABBIX_PASSIVE (0, par défaut), TRAP (2), SIMPLE (3), INTERNAL (5), ZABBIX_ACTIVE (7), EXTERNAL (10), ODBC (11), IPMI (12), SSH (13), TELNET (14), CALCULATED (15), JMX (16), SNMP_TRAP (17), DEPENDENT (18), HTTP_AGENT (19), SNMP_AGENT (20), ITEM_TYPE_SCRIPT (21), ITEM_TYPE_BROWSER (22), TELEMETRY_QUERY (24).
snmp_oid	string	(obligatoire pour les éléments SNMP_AGENT) Identifiant d'objet SNMP.
key	string	(obligatoire) Clé de l'élément.
delay	string	Intervalle de mise à jour de l'élément. Valeur par défaut : 1m. La valeur est toujours 0 pour les éléments TRAP.
history	string	Période (utilisant un suffixe temporel, une macro utilisateur ou une macro LLD) pendant laquelle les données d'historique doivent être conservées. Valeur par défaut : 31d.
trends	string	Période (utilisant un suffixe temporel, une macro utilisateur ou une macro LLD) pendant laquelle les données de tendance doivent être conservées. Valeur par défaut : 365d.
status	string	État de l'élément. Valeurs possibles :[1] ENABLED (0, par défaut), DISABLED (1).
value_type	string	Type de valeur reçue. Valeurs possibles :[1] FLOAT (0), CHAR (1), LOG (2), UNSIGNED (3, par défaut), TEXT (4), BINARY (5), JSON (6).
allowed_hosts	string	Liste d'adresses IP d'hôtes, séparées par des virgules, autorisées à envoyer des données pour l'élément. Pris en charge pour les éléments TRAP et HTTP_AGENT.
units	string	Unités de la valeur reçue (bps, B, etc.).
params	text	Paramètres supplémentaires selon le type de l'élément (script exécuté pour les éléments SSH et TELNET ; requête SQL pour les éléments ODBC ; formule pour les éléments CALCULATED ; script pour les éléments ITEM_TYPE_SCRIPT et ITEM_TYPE_BROWSER).
ipmi_sensor	string	Capteur IPMI. Pris en charge pour les éléments IPMI.

Élément	Type	Description
authtype	string	Type d'authentification. Pris en charge pour les éléments SSH et HTTP_AGENT. Valeurs possibles pour les éléments SSH :[1] PASSWORD (0, par défaut), PUBLIC_KEY (1). Valeurs possibles pour les éléments HTTP_AGENT :[1] NONE (0, par défaut), BASIC (1), NTLM (2), Kerberos (3) ou Digest (4).
username	string	(obligatoire pour les éléments SSH et TELNET) Nom d'utilisateur utilisé pour l'authentification. Pris en charge pour les éléments SIMPLE, ODBC, JMX et HTTP_AGENT. Lorsqu'il est utilisé pour les éléments JMX, `password` (voir ci-dessous) doit également être indiqué, ou les deux éléments doivent rester vides.
password	string	(obligatoire pour les éléments SSH et TELNET) Mot de passe utilisé pour l'authentification. Pris en charge pour les éléments SIMPLE, ODBC, JMX et HTTP_AGENT. Lorsqu'il est utilisé pour les éléments JMX, `username` (voir ci-dessus) doit également être indiqué, ou les deux éléments doivent rester vides.
publickey	string	(obligatoire pour les éléments SSH) Nom du fichier de clé publique.
privatekey	string	(obligatoire pour les éléments SSH) Nom du fichier de clé privée.
description	text	Description de l'élément.
inventory_link	string	Champ d'inventaire de l'hôte alimenté par l'élément. Valeurs possibles :[1] NONE (0), ALIAS (4), etc. (voir Inventaire de l'hôte pour les champs pris en charge).
valuemap		Élément racine des mappages de valeurs de l'élément.
name	string	(obligatoire) Nom du mappage de valeurs à utiliser pour l'élément.
logtimefmt	string	Format de l'heure dans les entrées de journal. Pris en charge pour les éléments de type de valeur LOG.
preprocessing		Élément racine du prétraitement des valeurs de l'élément.
step		Élément racine des étapes de prétraitement des valeurs des éléments de modèle.
jmx_endpoint	string	Point de terminaison JMX. Pris en charge pour les éléments JMX.
master_item		(obligatoire pour les éléments DEPENDENT) Élément racine de l'élément maître de l'élément dépendant.
key	string	(obligatoire) Clé de l'élément maître de l'élément dépendant.
timeout	string	Délai d'expiration de la requête d'interrogation des données de l'élément. Pris en charge pour la liste des types d'éléments dans Délais d'expiration.
url	string	(obligatoire pour les éléments HTTP_AGENT) Chaîne d'URL.
query_fields		Élément racine des paramètres de requête. Pris en charge pour les éléments HTTP_AGENT.
name	string	(obligatoire pour les éléments HTTP_AGENT) Nom du paramètre de requête.
value	string	Valeur du paramètre de requête. Pris en charge pour les éléments HTTP_AGENT.
parameters		Élément racine des paramètres définis par l'utilisateur. Pris en charge pour les éléments ITEM_TYPE_SCRIPT et ITEM_TYPE_BROWSER.
name	string	(obligatoire pour les éléments ITEM_TYPE_SCRIPT et ITEM_TYPE_BROWSER) Nom du paramètre défini par l'utilisateur.
value	string	Valeur du paramètre défini par l'utilisateur. Pris en charge pour les éléments ITEM_TYPE_SCRIPT et ITEM_TYPE_BROWSER.
posts	string	Données du corps de la requête HTTP(S). Pris en charge pour les éléments HTTP_AGENT.
status_codes	string	Plages de codes d'état HTTP requis, séparées par des virgules. Pris en charge pour les éléments HTTP_AGENT.
follow_redirects	string	Suivre les redirections de la réponse lors de l'interrogation des données. Pris en charge pour les éléments HTTP_AGENT. Valeurs possibles :[1] NO (0), YES (1, par défaut).
post_type	string	Type du corps des données envoyées. Pris en charge pour les éléments HTTP_AGENT. Valeurs possibles :[1] RAW (0, par défaut), JSON (2), XML (3).
http_proxy	string	Chaîne de connexion au proxy HTTP(S). Pris en charge pour les éléments HTTP_AGENT.
headers		Élément racine des en-têtes de requête HTTP(S). Pris en charge pour les éléments HTTP_AGENT.
name	string	(obligatoire pour les éléments HTTP_AGENT) Nom de l'en-tête.

Élément	Type	Description
value	string	(obligatoire pour les éléments HTTP_AGENT) Valeur de l'en-tête.
retrieve_mode	string	Partie de la réponse à stocker. Pris en charge pour les éléments HTTP_AGENT. Valeurs possibles :[1] BODY (0, par défaut), HEADERS (1), BOTH (2).
request_method	string	Type de méthode de requête. Pris en charge pour les éléments HTTP_AGENT. Valeurs possibles :[1] GET (0, par défaut), POST (1), PUT (2), HEAD (3).
output_format	string	Méthode de traitement de la réponse. Pris en charge pour les éléments HTTP_AGENT. Valeurs possibles :[1] RAW (0, par défaut), JSON (1).
allow_traps	string	Autoriser le remplissage de la valeur de la même manière que pour l'élément trapper. Pris en charge pour les éléments HTTP_AGENT. Valeurs possibles :[1] NO (0, par défaut), YES (1).
ssl_cert_file	string	Chemin du fichier de clé SSL publique. Pris en charge pour les éléments HTTP_AGENT.
ssl_key_file	string	Chemin du fichier de clé SSL privée. Pris en charge pour les éléments HTTP_AGENT.
ssl_key_password	string	Mot de passe du fichier de clé SSL. Pris en charge pour les éléments HTTP_AGENT.
verify_peer	string	Indique s'il faut vérifier que le certificat de l'hôte est authentique. Pris en charge pour les éléments HTTP_AGENT. Valeurs possibles :[1] NO (0, par défaut), YES (1).
verify_host	string	Indique s'il faut vérifier que le nom d'hôte de la connexion correspond à celui indiqué dans le certificat de l'hôte. Pris en charge pour les éléments HTTP_AGENT. Valeurs possibles :[1] NO (0, par défaut), YES (1).
time_shift	string	Décalage vers le passé (avant l'heure actuelle) correspondant à la fin de la plage temporelle interrogée par l'élément. Pris en charge pour les éléments TELEMETRY_QUERY.
lookback_limit	string	Décalage maximal vers le passé (avant l'heure actuelle) pendant lequel l'élément recherche des données lorsqu'il n'existe aucune requête précédente à poursuivre (par exemple, lors de la première exécution de l'élément), ou lorsque la requête précédente est antérieure à `delay` (par exemple, parce que le serveur Zabbix était arrêté). Pris en charge pour les éléments TELEMETRY_QUERY.
granularity	string	Durée d'un segment en lequel la plage temporelle complète de la requête est divisée. Pris en charge pour les éléments TELEMETRY_QUERY.
query		(obligatoire pour les éléments TELEMETRY_QUERY) Élément racine de la requête de télémétrie. Définit ce que l'élément interroge dans la source de données, la manière dont il filtre les données et la manière dont il agrège le résultat.
signal_type	string	(obligatoire) Type de signal interrogé par l'élément. Pris en charge pour les éléments TELEMETRY_QUERY. Valeurs possibles :[1] TRACES (0), METRICS (1), LOGS (2).
metric_point_type	string	(obligatoire pour le type de signal METRICS) Type de point de mesure interrogé par l'élément. Pris en charge pour les éléments TELEMETRY_QUERY. Valeurs possibles :[1] SUM (0), GAUGE (1), HISTOGRAM (2), EXPONENTIAL_HISTOGRAM (3).
columns		(obligatoire) Élément racine des colonnes de données de télémétrie interrogées par l'élément. Cet élément contient les éléments suivants : `column (string)` - (obligatoire) Nom de la colonne de données de télémétrie interrogée par l'élément. `attribute_key (string)` - (obligatoire pour les colonnes contenant des paires clé-valeur) Nom de la clé recherchée par l'élément dans la colonne contenant les paires clé-valeur.

Élément		Type	Description
	aggregated_columns		(obligatoire) Élément racine des colonnes agrégées de données de télémétrie interrogées par l'élément. Cet élément contient les éléments suivants : `column` (`string`) - (obligatoire pour les fonctions MIN, MAX, AVG, SUM et PERCENTILE) Nom de la colonne de données de télémétrie interrogée par l'élément. `function` (`string`) - (obligatoire) Fonction appliquée par l'élément aux enregistrements de la colonne. Valeurs possibles :[1] MIN (1), MAX (2), AVG (3), COUNT (4), SUM (5), PERCENTILE (8). `parameters` (`array`) - (obligatoire pour les fonctions PERCENTILE) Centile calculé par l'élément à partir des enregistrements de la colonne. `alias` (`string`) - (obligatoire) Nom unique de la valeur calculée, utilisé pour l'identifier dans la valeur de l'élément.
	filter		(obligatoire) Élément racine du filtre de requête de télémétrie appliqué aux enregistrements de colonne dans la requête. Cet élément contient les éléments suivants : `evaltype` (`string`) - (obligatoire) Méthode d'évaluation des conditions du filtre. Valeurs possibles :[1] AND_OR (0), AND (1), OR (2), FORMULA (3). `formula` (`string`) - (obligatoire pour les méthodes d'évaluation FORMULA) Expression définie par l'utilisateur utilisée pour évaluer les conditions des filtres avec une expression personnalisée. `conditions` (`string`) - (obligatoire) Élément racine des conditions de filtrage des données de télémétrie auxquelles un enregistrement d'une colonne de données de télémétrie doit satisfaire. L'élément `conditions` contient les éléments suivants : `column` (`string`) - (obligatoire) Nom de la colonne de données de télémétrie à laquelle s'appliquent les conditions du filtre. `attribute_key` (`string`) - (obligatoire pour les colonnes contenant des paires clé-valeur) Nom de la clé à laquelle la condition s'applique, dans la colonne contenant les paires clé-valeur. `operator` (`string`) - (obligatoire) Opérateur de condition du filtre. Valeurs possibles pour les colonnes à valeur unique :[1] EQUAL (0), NOT_EQUAL (1), LIKE (2), NOT_LIKE (3). Valeurs possibles pour les colonnes contenant des paires clé-valeur :[1] EQUAL (0), NOT_EQUAL (1), EXISTS (12). `formulaid` (`string`) - (obligatoire pour les méthodes d'évaluation FORMULA) Identifiant unique arbitraire utilisé pour référencer la condition depuis une expression personnalisée. `value` (`string`) - (obligatoire pour les opérations EQUAL, NOT_EQUAL, LIKE et NOT LIKE) Valeur à comparer.
tags			Élément racine des étiquettes de l'élément.
	tag	string	(obligatoire) Nom de l'étiquette.
	value	string	Valeur de l'étiquette.
triggers			Élément racine des déclencheurs des éléments de modèle.

Note:
Voir également : Objet élément (consultez la propriété correspondante portant le même nom).

Étapes de prétraitement de la valeur de l'élément du modèle

```
zabbix_export:
  (...)
  templates:
    (...)
      items:
        (...)
        - uuid: 58818005e76d46dda14d6592f601ab00
          (...)
          preprocessing:
            - type: JSONPATH
```

```
            parameters:
              - $.length()
          - type: DISCARD_UNCHANGED_HEARTBEAT
            parameters:
              - 12h
        (...)
```

Element	Type	Description
type	string	(obligatoire) Type de l'étape de prétraitement de la valeur de l'élément. Valeurs possibles :[1] MULTIPLIER (1), RTRIM (2), LTRIM (3), TRIM (4), REGEX (5), BOOL_TO_DECIMAL (6), OCTAL_TO_DECIMAL (7), HEX_TO_DECIMAL (8), SIMPLE_CHANGE (9, calculé : valeur reçue - valeur précédente), CHANGE_PER_SECOND (10, calculé : (valeur reçue - valeur précédente)/(heure actuelle - heure de la dernière vérification)), XMLPATH (11), JSONPATH (12), IN_RANGE (13), MATCHES_REGEX (14), NOT_MATCHES_REGEX (15), CHECK_JSON_ERROR (16), CHECK_XML_ERROR (17), CHECK_REGEX_ERROR (18), DISCARD_UNCHANGED (19), DISCARD_UNCHANGED_HEARTBEAT (20), JAVASCRIPT (21), PROMETHEUS_PATTERN (22), PROMETHEUS_TO_JSON (23), CSV_TO_JSON (24), STR_REPLACE (25), CHECK_NOT_SUPPORTED (26), XML_TO_JSON (27), SNMP_WALK_VALUE (28), SNMP_WALK_TO_JSON (29), SNMP_GET_VALUE (30).
parameters		(obligatoire) Élément racine pour les paramètres de l'étape de prétraitement de la valeur de l'élément.
parameter	string	Paramètre individuel de l'étape de prétraitement de la valeur de l'élément.
error_handler	string	Type d'action utilisé en cas d'échec de l'étape de prétraitement. Valeurs possibles :[1] ORIGINAL_ERROR (0, par défaut), DISCARD_VALUE (1), CUSTOM_VALUE (2), CUSTOM_ERROR (3).
error_handler_params	string	Paramètres du gestionnaire d'erreurs.

Note:
Voir aussi : Objet de prétraitement de l'élément (se référer à la propriété correspondante portant le même nom).

Déclencheurs d'élément de modèle

```
zabbix_export:
  (...)
  templates:
    (...)
      items:
        (...)
        - uuid: 58818005e76d46dda14d6592f601ab00
          (...)
          triggers:
            - uuid: b950c306394f4b3c902060a8273cbcde
              expression: 'change(/Linux by Zabbix agent/system.sw.packages.get)<>0'
              name: 'Linux: Le nombre de paquets installés a changé'
              priority: WARNING
              manual_close: 'YES'
              tags:
                - tag: scope
                  value: notice
        (...)
        - uuid: 403cebed115441369e94d35d070ca7b8
          (...)
          triggers:
            - uuid: cfd395b1cde74ef18a5e5f840bd5142a
              expression: 'min(/Linux by Zabbix agent/vm.memory.utilization,5m)>{$MEMORY.UTIL.MAX}'
              name: 'Linux: Utilisation mémoire élevée'
              event_name: 'Linux: Utilisation mémoire élevée (>{$MEMORY.UTIL.MAX}% pendant 5m)'
              priority: AVERAGE
              description: 'Le système manque de mémoire libre.'
              dependencies:
```

```
            - name: 'Linux: Manque de mémoire disponible'
              expression: 'max(/Linux by Zabbix agent/vm.memory.size[available],5m)<{$MEMORY.AVAILABLE
          tags:
            - tag: scope
              value: capacity
            - tag: scope
              value: performance
    (...)
```

Element	Type	Description
uuid	string	(obligatoire) Identifiant unique de ce déclencheur.
expression	string	(obligatoire) Expression du déclencheur.
recovery_mode	string	Base de génération des événements OK. Valeurs possibles :[1] EXPRESSION (0, par défaut), RECOVERY_EXPRESSION (1), NONE (2).
recovery_expression	string	Expression de récupération du déclencheur.
correlation_mode	string	Mode de corrélation (aucune corrélation d'événements ou corrélation par tag). Valeurs possibles :[1] DISABLED (0, par défaut), TAG_VALUE (1).
correlation_tag	string	Nom du tag à utiliser pour la corrélation des événements.
name	string	(obligatoire) Nom du déclencheur.
event_name	string	Nom de l'événement.
opdata	string	Données opérationnelles.
url_name	string	Libellé de l'URL associée au déclencheur.
url	string	URL associée au déclencheur.
status	string	État du déclencheur. Valeurs possibles :[1] ENABLED (0, par défaut), DISABLED (1).
priority	string	Sévérité du déclencheur. Valeurs possibles :[1] NOT_CLASSIFIED (0, par défaut), INFO (1), WARNING (2), AVERAGE (3), HIGH (4), DISASTER (5).
description	text	Description du déclencheur.
type	string	Type de génération d'événements (un seul événement de problème ou plusieurs événements de problème). Valeurs possibles :[1] SINGLE (0, par défaut), MULTIPLE (1).
manual_close	string	Fermeture manuelle des événements de problème. Valeurs possibles :[1] NO (0, par défaut), YES (1).
dependencies		Élément racine pour les dépendances.
name	string	(obligatoire) Nom du déclencheur dépendant.
expression	string	(obligatoire) Expression du déclencheur dépendant.
recovery_expression	string	Expression de récupération du déclencheur dépendant.
tags		Élément racine pour les tags du déclencheur.
tag	string	(obligatoire) Nom du tag.
value	string	Valeur du tag.

Note:
Voir aussi : Objet Trigger (reportez-vous à la propriété correspondante portant le même nom).

Règles de découverte de bas niveau du modèle

```
zabbix_export:
  (...)
  templates:
    (...)
      discovery_rules:
        - uuid: acfdea9c46ef48c68e6636f43b8f96a2
          name: 'Network interface discovery'
          key: net.if.discovery
          delay: 1h
          filter: (see table below)
          description: 'The discovery of network interfaces.'
          item_prototypes: (see table below)
          trigger_prototypes: (see table below)
          graph_prototypes: (see table below)
```

```
(...)
```

Attention:
La plupart des éléments des règles de découverte de bas niveau du modèle sont identiques à ceux des éléments du modèle. Le tableau ci-dessous décrit les éléments qui diffèrent des éléments du modèle.

Element	Type	Description
type	string	Type d'élément. Valeurs possibles:[1] ZABBIX_PASSIVE (0, par défaut), TRAP (2), SIMPLE (3), INTERNAL (5), ZABBIX_ACTIVE (7), EXTERNAL (10), ODBC (11), IPMI (12), SSH (13), TELNET (14), JMX (16), DEPENDENT (18), HTTP_AGENT (19), SNMP_AGENT (20), ITEM_TYPE_SCRIPT (21), ITEM_TYPE_BROWSER (22).
key	string	(obligatoire) La clé de la règle de découverte de bas niveau.
filter		Élément racine pour les filtres des règles de découverte de bas niveau du modèle.
lifetime	string	Période de temps (en secondes, suffixe de temps, ou macro utilisateur), après laquelle les ressources découvertes qui n'existent plus seront supprimées. Par défaut: 7d.
lifetime_type	string	Scénario de suppression des ressources LLD perdues. Valeurs possibles: DELETE_NEVER, DELETE_IMMEDIATELY, DELETE_AFTER.
enabled_lifetime	string	Période de temps (en secondes, suffixe de temps, ou macro utilisateur), après laquelle les ressources découvertes qui n'existent plus seront désactivées.
enabled_lifetime_type	string	Scénario de désactivation des ressources LLD perdues. Valeurs possibles: DISABLE_NEVER, DISABLE_IMMEDIATELY, DISABLE_AFTER.
item_prototypes		Élément racine pour les éléments prototypes du modèle, qui sont identiques à ceux des éléments du modèle.
trigger_prototypes		Élément racine pour les prototypes de déclencheur du modèle, qui sont identiques à ceux des déclencheurs d'élément du modèle.
graph_prototypes		Élément racine pour les prototypes de graphique du modèle, qui sont identiques à ceux des graphiques d'hôte.
host_prototypes		Élément racine pour les prototypes d'hôte du modèle, qui sont identiques à ceux des hôtes.
parent_discovery_rule		Élément racine pour la règle de découverte de bas niveau parente (ou le prototype de règle) du prototype de règle de découverte de bas niveau. Cette propriété indique qu'il s'agit d'un prototype de règle LLD, enfant direct de la règle/prototype de règle référencé.
key	string	(obligatoire) La clé de la règle de découverte de bas niveau parente (ou du prototype de règle).
master_item	string	(obligatoire pour les règles DEPENDENT) Élément racine pour l'élément maître de la règle dépendante.
lld_macro_paths		Élément racine pour les chemins de macro de la règle de découverte de bas niveau.
lld_macro	string	(obligatoire) Nom de la macro de la règle de découverte de bas niveau.
path	string	(obligatoire) Sélecteur de la valeur qui sera attribuée à la macro correspondante.
preprocessing		Élément racine pour le prétraitement de la valeur de la règle de découverte de bas niveau.
step		Élément racine pour les éléments d'étape de prétraitement de la valeur de la règle de découverte de bas niveau, qui sont identiques à ceux des étapes de prétraitement de la valeur d'élément du modèle, mais avec moins de valeurs possibles. Voir aussi: objet de prétraitement de la règle LLD.
overrides		Élément racine pour les règles de remplacement de la règle de découverte de bas niveau.
name	string	(obligatoire) Nom unique du remplacement.
step	string	(obligatoire) Numéro d'ordre unique du remplacement.
stop	string	Arrêter le traitement des remplacements suivants si correspondance.
filter		Élément racine pour les éléments de filtre de la règle de remplacement de la règle de découverte de bas niveau du modèle, qui sont identiques à ceux des filtres des règles de découverte de bas niveau du modèle.
operations		Élément racine pour les opérations de remplacement des règles de découverte de bas niveau du modèle.

Note:
Voir aussi: objet de règle LLD (reportez-vous à la propriété correspondante portant le même nom).

Filtres de règle de découverte bas niveau du modèle

```
zabbix_export:
  (...)
  templates:
    (...)
      discovery_rules:
        - uuid: acfdea9c46ef48c68e6636f43b8f96a2
          (...)
          filter:
            evaltype: AND
            conditions:
              - macro: '{#IFNAME}'
                value: '{$NET.IF.IFNAME.MATCHES}'
              - macro: '{#IFNAME}'
                value: '{$NET.IF.IFNAME.NOT_MATCHES}'
                operator: NOT_MATCHES_REGEX
          (...)
        - uuid: 2bbdc79f082d4c618e01bec625e9c90a
          (...)
          filter:
            evaltype: AND
            conditions:
              - macro: '{#DEVNAME}'
                value: '{$VFS.DEV.DEVNAME.MATCHES}'
              - macro: '{#DEVNAME}'
                value: '{$VFS.DEV.DEVNAME.NOT_MATCHES}'
                operator: NOT_MATCHES_REGEX
              - macro: '{#DEVTYPE}'
                value: disk
          (...)
```

Element	Type	Description
evaltype	string	Remplace la méthode d'évaluation des conditions du filtre. Valeurs possibles :[1] AND_OR (0, par défaut), AND (1), OR (2), FORMULA (3).
formula	string	Formule de calcul personnalisée pour les conditions du filtre.
conditions		Élément racine pour les conditions du filtre.
macro	string	(obligatoire) Nom de la macro de découverte bas niveau sur laquelle effectuer la vérification.
value	string	Valeur à comparer.
operator	string	Opérateur de condition. Valeurs possibles :[1] MATCHES_REGEX (8, par défaut), NOT_MATCHES_REGEX (9).
formulaid	string	(obligatoire) ID unique arbitraire utilisé pour référencer une condition dans l'expression personnalisée. Peut contenir uniquement des lettres majuscules. L'ID doit être défini par l'utilisateur lors de la modification des conditions du filtre, mais sera généré à nouveau lors d'une requête ultérieure.

Note:
Voir aussi : objet de filtre de règle LLD (reportez-vous à la propriété correspondante portant le même nom).

Opérations de remplacement de règle de découverte de bas niveau du modèle

Element	Type	Description
operationobject	string	Objet auquel l'opération est appliquée. Valeurs possibles:[1] ITEM_PROTOTYPE (0), TRIGGER_PROTOTYPE (1), GRAPH_PROTOTYPE (2), HOST_PROTOTYPE (3).

Element	Type	Description
operator	string	Opérateur de condition de remplacement. Valeurs possibles:[1] EQUAL (1), NOT_EQUAL (2), LIKE (3), NOT_LIKE (4), REGEXP (5), NOT_REGEXP (6).
value	string	Une expression régulière ou une chaîne pour l'opérateur de condition de remplacement.
status	string	Statut de l'objet lors de l'opération de remplacement.
discover	string	Indique si l'objet est ajouté à la suite de la découverte.
delay	string	Intervalle de mise à jour défini pour le prototype d'élément lors de l'opération de remplacement.
history	string	Période de conservation de l'historique définie pour le prototype d'élément lors de l'opération de remplacement.
trends	string	Période de conservation des tendances définie pour le prototype d'élément lors de l'opération de remplacement.
severity	string	Niveau de gravité du prototype de déclencheur défini lors de l'opération de remplacement.
tags		Élément racine pour les tags définis pour l'objet lors de l'opération de remplacement.
tag	string	(obligatoire) Nom du tag.
value	string	Valeur du tag.
templates		Élément racine pour les modèles liés au prototype d'hôte lors de l'opération de remplacement.
name	string	(obligatoire) Nom du modèle.
inventory_mode	string	Mode d'inventaire du prototype d'hôte défini lors de l'opération de remplacement.

Note:
Voir aussi: Objet d'opération de remplacement de règle LLD (reportez-vous à la propriété correspondante portant le même nom).

Modèles de scénarios web

Element	Type	Description
uuid	string	(obligatoire) Identifiant unique de ce scénario web.
name	string	(obligatoire) Nom du scénario web.
delay	string	Fréquence (en secondes, suffixe de temps, ou macro utilisateur) d'exécution du scénario web. Par défaut: 1m.
attempts	integer	Nombre de tentatives pour l'exécution des étapes du scénario web. Valeurs possibles:[1] 1-10 (par défaut: 1).
agent	string	Agent client. Zabbix se fera passer pour le navigateur sélectionné. Cela est utile lorsqu'un site web renvoie un contenu différent selon le navigateur. Par défaut: Zabbix.
http_proxy	string	Proxy qui sera utilisé par le scénario web, au format: `http://[username[:password]@]proxy.example.com[:port]`
variables		Élément racine pour les variables du scénario web pouvant être utilisées dans les étapes du scénario.
name	string	(obligatoire) Nom de la variable.
value	text	(obligatoire) Valeur de la variable.
headers		Élément racine pour les en-têtes HTTP à envoyer lors de l'exécution d'une requête. Les en-têtes doivent être listés en utilisant la même syntaxe que celle qu'ils auraient dans le protocole HTTP.
name	string	(obligatoire) Nom de l'en-tête.
value	text	(obligatoire) Valeur de l'en-tête.
status	string	Statut du scénario web. Valeurs possibles:[1] ENABLED (0, par défaut), DISABLED (1).
authentication	string	Méthode d'authentification. Valeurs possibles:[1] NONE (0, par défaut), BASIC (1), NTLM (2), Kerberos (3), ou Digest (4).
http_user	string	Nom d'utilisateur utilisé pour l'authentification BASIC (HTTP), NTLM, Kerberos ou Digest.
http_password	string	Mot de passe utilisé pour l'authentification BASIC (HTTP), NTLM, Kerberos ou Digest.
verify_peer	string	Vérifier le certificat SSL du serveur web. Valeurs possibles:[1] NO (0, par défaut), YES (1).

Element	Type	Description
verify_host	string	Vérifier que le champ *Common Name* ou le champ *Subject Alternate Name* du certificat du serveur web correspond. Valeurs possibles:[1] NO (0, par défaut), YES (1).
ssl_cert_file	string	Nom du fichier de certificat SSL utilisé pour l'authentification du client (doit être au format PEM).
ssl_key_file	string	Nom du fichier de clé privée SSL utilisé pour l'authentification du client (doit être au format PEM).
ssl_key_password	string	Mot de passe du fichier de clé privée SSL.
steps		(obligatoire) Élément racine pour les étapes du scénario web du modèle.
tags		Élément racine pour les tags du scénario web.
tag	string	(obligatoire) Nom du tag.
value	string	Valeur du tag.

Note:
Voir aussi: Objet scénario web (se référer à la propriété correspondante portant le même nom).

Étapes du scénario web

Element	Type	Description
name	string	(required) Nom de l'étape du scénario web.
url	string	(required) URL à surveiller.
query_fields		Élément racine pour les paramètres de requête (un tableau de champs HTTP à ajouter à l'URL lors de l'exécution d'une requête).
name	string	(required) Nom du paramètre de requête.
value	string	Valeur du paramètre de requête.
posts		Élément racine pour les variables HTTP POST (une chaîne (données POST brutes) ou un tableau de champs HTTP (données de champs de formulaire)).
name	string	(required) Nom du champ POST.
value	string	(required) Valeur du champ POST.
variables		Élément racine des variables au niveau de l'étape (macros) qui doivent être appliquées après cette étape. Si la valeur de la variable a un préfixe 'regex:', alors sa valeur est extraite des données renvoyées par cette étape selon le modèle d'expression régulière suivant le préfixe 'regex:'
name	string	(required) Nom de la variable.
value	text	(required) Valeur de la variable.
headers		Élément racine pour les en-têtes HTTP à envoyer lors de l'exécution d'une requête.
name	string	(required) Nom de l'en-tête.
value	text	(required) Valeur de l'en-tête.
follow_redirects	string	Suivre les redirections HTTP. Valeurs possibles:[1] NO (0), YES (1, default).
retrieve_mode	string	Mode de récupération de la réponse HTTP. Valeurs possibles:[1] BODY (0, default), HEADERS (1), BOTH (2).
timeout	string	Délai d'attente (en secondes, suffixe de temps, ou macro utilisateur) de l'exécution de l'étape. Par défaut: 15s.
required	string	Texte qui doit être présent dans la réponse (ignoré si vide).
status_codes	string	Liste de codes d'état HTTP acceptés, séparés par des virgules (par exemple, 200-201,210-299; ignoré si vide).

Note:
Voir aussi: Objet d'étape de scénario web (se référer à la propriété correspondante portant le même nom).

Configuration des macros de modèle

```
zabbix_export:
  (...)
  templates:
```

```yaml
  - uuid: f8f7908280354f2abeed07dc788c3747
    (...)
    macros:
      - macro: '{$AGENT.TIMEOUT}'
        (...)
        config:
          type: TEXT
          priority: '1'
          label: 'Secondes depuis la dernière détection de l agent Zabbix'
          description: 'Délai après lequel l agent est considéré comme indisponible.'
      - macro: '{$CPU.UTIL.CRIT}'
        (...)
        config:
          type: TEXT
          priority: '2'
          section_name: Thresholds
          label: 'Seuil d utilisation CPU exprimé'
          description: 'Seuil critique d utilisation CPU exprimé en %. Dans la plage de 0 à 100 inclus.'
          regex: '^-?([0-9]+|(([0-9]+)\.([0-9]+)))$'
      (...)
      - macro: '{$IFCONTROL}'
        (...)
        config:
          type: CHECKBOX
          priority: '19'
          label: 'Contrôle des interfaces'
          description: 'Déclencher un déclencheur lorsque l état opérationnel de l interface change à "D
          options:
            - checked: '1'
              unchecked: '0'
      (...)
```

Element	Type	Description
type	string	(obligatoire) Type du champ de saisie de la macro. Valeurs possibles :[1] NOCONF (0), TEXT (1), LIST (2), CHECKBOX (3).
priority	string	Position de la macro dans la liste des macros.
section_name	string	Libellé de la section repliable dans laquelle la macro est regroupée.
label	string	(obligatoire pour les macros TEXT, LIST, CHECKBOX) Libellé de la macro.
description	text	Texte d aide de la macro. Prend en charge le formatage Markdown.
required	string	Indique que la macro est obligatoire. Valeurs possibles :[1] NO (0), YES (1). Pris en charge pour les macros TEXT et LIST.
regex	string	Expression régulière permettant de valider la saisie de l utilisateur dans un champ de texte. Pris en charge pour les macros TEXT.
options		Élément racine pour les éléments LIST ou les valeurs CHECKBOX.
value	string	(obligatoire pour les macros LIST) Valeur de l élément LIST.
text	string	(obligatoire pour les macros LIST) Texte de l élément LIST.
checked	string	(obligatoire pour les macros CHECKBOX) Valeur représentant l état coché. Valeurs possibles :[1] 0 (false), 1 (true). Pris en charge pour les macros CHECKBOX.
unchecked	string	(obligatoire pour les macros CHECKBOX) Valeur représentant l état non coché. Valeurs possibles :[1] 0 (false), 1 (true). Pris en charge pour les macros CHECKBOX.

Note:
Voir aussi : Objet de configuration des macros (reportez-vous à la propriété correspondante portant le même nom).

Tableaux de bord de modèle

```
zabbix_export:
  (...)
  templates:
    (...)
      dashboards:
        - uuid: c689ad3115fd46a4b927d1f70ee2e5a4
          name: Filesystems
          pages:
            - name: Overview
              widgets: (see table below)
        (...)
```

Element	Type	Description
uuid	string	(required) Identifiant unique pour ce tableau de bord.
name	string	(required) Nom du tableau de bord du modèle.
display pe- riod	integer	Période d'affichage des pages du tableau de bord.
auto_start	string	Démarrage automatique du diaporama. Valeurs possibles:[1] NO (0), YES (1, default).
pages		Élément racine pour les pages du tableau de bord du modèle.
name	string	Nom de la page.
display period	integer	Intervalle du diaporama.
sortorder	integer	Ordre de tri des pages.
widgets		Élément racine pour les widgets du tableau de bord du modèle.

Note:
Voir aussi: Objet de tableau de bord de modèle (se référer à la propriété correspondante portant le même nom).

Widgets du tableau de bord du modèle

```
zabbix_export:
  (...)
  templates:
    (...)
      dashboards:
        - uuid: c689ad3115fd46a4b927d1f70ee2e5a4
          name: Filesystems
          pages:
            - name: Overview
              widgets:
                - type: graphprototype
                  width: '72'
                  height: '5'
                  fields:
                    - type: INTEGER
                      name: columns
                      value: '3'
                    - type: GRAPH_PROTOTYPE
                      name: graphid.0
                      value:
                        host: 'Linux by Zabbix agent'
                        name: 'FS [{#FSTYPE}({#FSNAME})]: Space utilization chart (relative to total)'
                    - type: STRING
                      name: reference
                      value: AAAEB
                - type: graphprototype
                  'y': '5'
                  width: '72'
                  height: '5'
```

```
                  fields:
                    - type: INTEGER
                      name: columns
                      value: '1'
                    - type: GRAPH_PROTOTYPE
                      name: graphid.0
                      value:
                        host: 'Linux by Zabbix agent'
                        name: 'FS [{#FSTYPE}({#FSNAME})]: Space usage graph, in % (relative to max availab
                    - type: STRING
                      name: reference
                      value: AAAEC
        (...)
```

Element	Type	Description
type	string	(obligatoire) Type de widget.
name	string	Nom du widget.
x	integer	Position horizontale à partir du côté gauche du tableau de bord du modèle. Valeurs possibles:[1] 0-71.
y	integer	Position verticale à partir du haut du tableau de bord du modèle. Valeurs possibles:[1] 0-63.
width	integer	Largeur du widget. Valeurs possibles:[1] 1-72.
height	integer	Hauteur du widget. Valeurs possibles:[1] 1-64.
hide_header	string	Masquer l'en-tête du widget. Valeurs possibles:[1] NO (0, par défaut), YES (1).
fields		Élément racine pour les champs du widget du tableau de bord du modèle.
type	string	(obligatoire) Type de champ du widget. Valeurs possibles:[1] INTEGER (0), STRING (1), ITEM (4), ITEM_PROTOTYPE (5), GRAPH (6), GRAPH_PROTOTYPE (7), MAP (8), SERVICE (9), SLA (10), USER (11), ACTION (12), MEDIA_TYPE (13).
name	string	(obligatoire) Nom du champ du widget.
value	mixed	(obligatoire) Valeur du champ du widget, selon le type de champ.

Note:
Voir aussi: Objet widget du tableau de bord du modèle (reportez-vous à la propriété correspondante portant le même nom).

Modèles de correspondance de valeurs

```
zabbix_export:
  (...)
  templates:
    (...)
      valuemaps:
        - uuid: 8c048c6cca8248f2860c208e8db0f59e
          name: ifOperStatus
          mappings:
            - value: '0'
              newvalue: unknown
            - value: '1'
              newvalue: notpresent
            - value: '2'
              newvalue: down
            - value: '3'
              newvalue: lowerlayerdown
            - value: '4'
              newvalue: testing
            - value: '5'
              newvalue: dormant
            - value: '6'
```

```
            newvalue: up
        (...)
```

Element	Type	Description
uuid	string	(obligatoire) Identifiant unique de cette correspondance de valeurs.
name	string	(obligatoire) Nom de la correspondance de valeurs.
mapping		Élément racine pour les correspondances.
type	string	Type de correspondance de la valeur. Valeurs possibles:[1] EQUAL (0, par défaut), GREATER_OR_EQUAL (2), LESS_OR_EQUAL (3), IN_RANGE (4), REGEXP (5), DEFAULT (6).
value	string	Valeur d'origine.
newvalue	string	(obligatoire) Valeur vers laquelle la valeur d'origine est mappée.

Note:
Voir aussi: Objet Value map (reportez-vous à la propriété correspondante portant le même nom).

Notes de bas de page [1] Les valeurs entières de l'API entre parenthèses, par exemple ENABLED (0), sont mentionnées uniquement à titre de référence. Pour plus d'informations, consultez la page de l'objet API liée dans l'entrée du tableau ou à la fin de chaque section.

Hôtes

Vue d'ensemble

Les hôtes sont exportés avec de nombreux objets associés et relations entre objets.

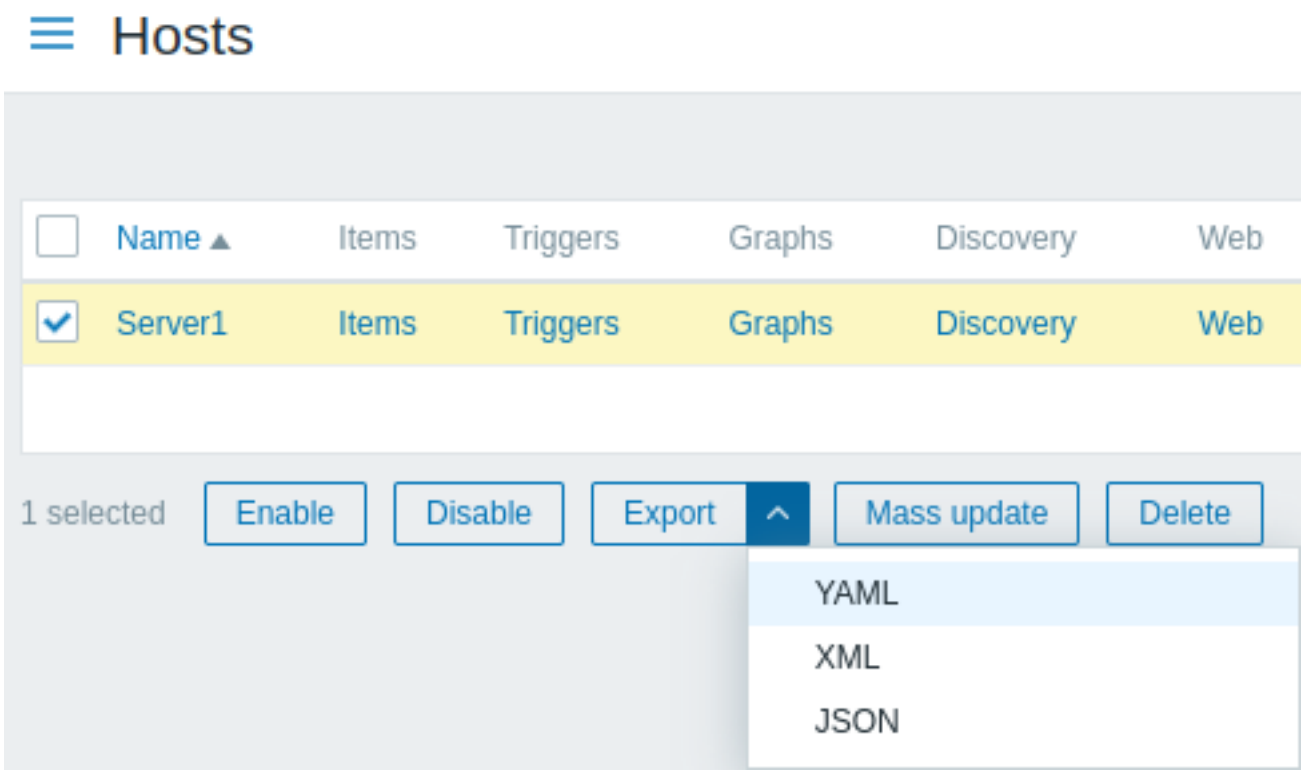
L'export d'hôte contient :

- Groupes d'hôtes liés
- Données de l'hôte
- Liaison de modèle
- Liaison de groupe d'hôtes
- Interfaces de l'hôte
- Éléments directement liés
- Déclencheurs directement liés
- Règles de découverte directement liées avec tous les prototypes
- Scénarios web directement liés
- Macros de l'hôte
- Données d'inventaire de l'hôte
- Correspondances de valeurs
- Graphiques liés

Exportation

Pour exporter des hôtes, procédez comme suit :

1. Accédez à *Collecte de données* > *Hôtes*.
2. Cochez les cases des hôtes à exporter.
3. Cliquez sur *Exporter* sous la liste pour exporter au format YAML par défaut, ou cliquez sur la flèche pour sélectionner un autre format d'exportation.



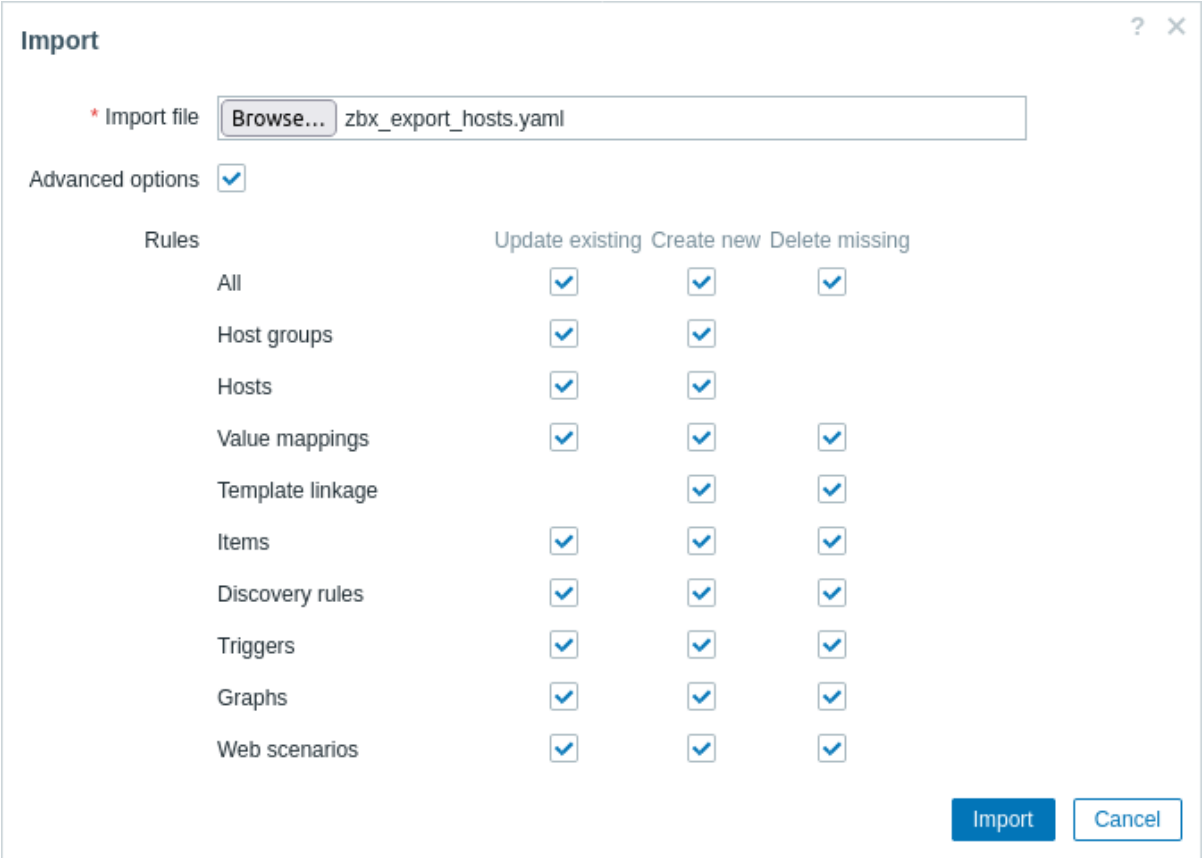
Selon le format sélectionné, les hôtes sont exportés vers un fichier local avec un nom par défaut :

- `zabbix_export_hosts.yaml` - à l'export YAML ;
- `zabbix_export_hosts.xml` - à l'export XML ;
- `zabbix_export_hosts.json` - à l'export JSON.

Importation

Pour importer des hôtes, procédez comme suit :

1. Accédez à *Collecte de données* > *Hôtes*.
2. Cliquez sur *Importer* dans le coin supérieur droit.
3. Sélectionnez le fichier d'importation.
4. Cochez les cases correspondant aux règles d'importation que vous souhaitez appliquer (voir ci-dessous).
5. Cliquez sur *Importer* dans le coin inférieur droit du formulaire de configuration.



Si vous cochez la case *Options avancées*, une liste détaillée de tous les éléments pouvant être importés s'affiche. Cochez ou décochez chaque règle d'importation selon vos besoins.

Si vous cliquez sur la case de la ligne *Tous*, tous les éléments situés en dessous seront cochés ou décochés.

Règles d'importation :

Règle	Description
Mettre à jour les éléments existants	Les éléments existants seront mis à jour à l'aide des données du fichier d'importation. Sinon, ils ne seront pas mis à jour.
Créer les nouveaux éléments	Les nouveaux éléments seront créés à l'aide des données du fichier d'importation. Sinon, ils ne seront pas créés.
Supprimer les éléments manquants	Les éléments existants qui ne figurent pas dans le fichier d'importation seront supprimés. Sinon, ils ne seront pas supprimés. Si *Supprimer les éléments manquants* est sélectionné pour *Liaison de modèles*, les liaisons actuelles avec les modèles qui ne figurent pas dans le fichier d'importation seront supprimées. Les entités (éléments, déclencheurs, graphiques, etc.) héritées des modèles dont la liaison a été supprimée ne seront pas supprimées (sauf si l'option *Supprimer les éléments manquants* est également sélectionnée pour chaque entité).

Vous ne pouvez pas examiner les modifications apportées aux hôtes pendant l'importation. Un message indiquant la réussite ou l'échec de l'importation s'affichera dans l'interface.

Format d'export

Lorsqu'un hôte est exporté, Zabbix génère un format YAML, JSON ou XML structuré. L'export inclut des éléments d'hôte, tels que les métadonnées de l'hôte, les éléments, les macros, les déclencheurs, et plus encore.

Chaque élément a une fonction spécifique et peut contenir des éléments imbriqués.

Les sections suivantes décrivent chaque élément dans le format d'export. Les exemples utilisent un hôte avec le modèle MySQL by Zabbix agent 2. À des fins de démonstration des éléments, le modèle est dissocié après la création de l'hôte. La dissociation conserve tous les éléments dans la configuration (contrairement à la dissociation et à la suppression, qui les supprime).

Une ellipse `(...)` indique des éléments omis par souci de concision. La note `(voir le tableau ci-dessous)` est utilisée lorsque l'élément est expliqué plus en détail dans les sections suivantes.

```yaml
zabbix_export:
  version: '8.0'
  host_groups:
    - uuid: 748ad4d098d447d492bb935c907f652f
      name: Databases
  hosts: (voir le tableau ci-dessous)
  graphs: (voir le tableau ci-dessous)
```

Métadonnées de l'hôte

Element		Type	Description
version		string	(obligatoire) Version de Zabbix.
host_groups			(obligatoire) Élément racine pour les groupes d'hôtes.
	uuid	string	(obligatoire) Identifiant unique pour ce groupe d'hôtes.
	name	string	(obligatoire) Nom du groupe d'hôtes.
hosts			Élément racine pour les hôtes.
graphs			Élément racine pour les graphes d'hôte.

Hôtes

```yaml
zabbix_export:
  (...)
  hosts:
    - host: 'MySQL server'
      name: 'MySQL server'
      groups:
        - name: Databases
      interfaces: (see table below)
      items: (see table below)
      discovery_rules: (see table below)
      tags:
        - tag: location
          value: Riga
      macros:
        (...)
        - macro: '{$MYSQL.DSN}'
          value: 192.0.2.1
        (...)
        - macro: '{$MYSQL.PASSWORD}'
          type: SECRET_TEXT
          description: 'Mot de passe de l'utilisateur MySQL.'
        (...)
      valuemaps: (see table below)
  (...)
```

Element	Type	Description
host	string	(required) Nom d'hôte unique.
name	string	Nom d'hôte visible.
description	text	Description de l'hôte.
monitored_by	string	Mode de surveillance de l'hôte. Valeurs possibles:[1] SERVER (0, par défaut), PROXY (1) ou PROXY_GROUP (2).
proxy		Élément racine pour le proxy.
name	string	(required) Nom du proxy (le cas échéant) qui surveille l'hôte.
proxy_group		Élément racine pour le groupe de proxy.
name	string	(required) Nom du groupe de proxy (le cas échéant) utilisé pour surveiller l'hôte.
status	string	Statut de l'hôte. Valeurs possibles:[1] ENABLED (0, par défaut), DISABLED (1).
ipmi_authtype	string	Type d'authentification de session IPMI. Valeurs possibles:[1] DEFAULT (-1, par défaut), NONE (0), MD2 (1), MD5 (2), STRAIGHT (4), OEM (5), RMCP_PLUS (6).

Element	Type	Description
ipmi_privilege	string	Niveau de privilège de session IPMI. Valeurs possibles:[1] CALLBACK (1), USER (2, par défaut), OPERATOR (3), ADMIN (4), OEM (5).
ipmi_username	string	Nom d'utilisateur pour les vérifications IPMI.
ipmi_password	string	Mot de passe pour les vérifications IPMI.
templates		Élément racine pour les modèles liés.
name	string	(required) Nom du modèle.
groups		Élément racine pour les groupes d'hôtes auxquels l'hôte appartient.
name	string	(required) Nom du groupe d'hôtes.
interfaces		Élément racine pour les interfaces d'hôte.
items		Élément racine pour les éléments d'hôte.
discovery_rules		Élément racine pour les règles de découverte de bas niveau de l'hôte.
http_tests		Élément racine pour les scénarios web de l'hôte.
tags		Élément racine pour les tags de l'hôte.
tag	string	(required) Nom du tag.
value	string	Valeur du tag.
macros		Élément racine pour les macros de l'hôte.
macro	string	(required) Nom de la macro utilisateur.
type	string	Type de macro utilisateur. Valeurs possibles:[1] TEXT (0, par défaut), SECRET_TEXT (1), VAULT (2).
value	string	Valeur de la macro utilisateur.
description	string	Description de la macro utilisateur.
inventory		Élément racine pour l'inventaire de l'hôte.
<inventory_property>	string	Propriété d'inventaire. Toutes les propriétés ont leur propre élément (`type`, `name`, `os`, etc.; par exemple, voir Format d'export).
inventory_mode	string	Mode d'inventaire. Valeurs possibles:[1] DISABLED (-1), MANUAL (0, par défaut), AUTOMATIC (1).
valuemaps		Élément racine pour les mappages de valeurs de l'hôte.

Note:
Voir aussi: Objet Hôte (reportez-vous à la propriété correspondante portant le même nom).

Interfaces d'hôte

```
zabbix_export:
  (...)
  hosts:
    - host: 'MySQL server'
      (...)
      interfaces:
        - ip: 192.0.2.1
          interface_ref: if1
      (...)
```

Element	Type	Description
default	string	Indique s'il s'agit de l'interface principale de l'hôte. Notez qu'il ne peut y avoir qu'une seule interface principale d'un même type sur un hôte. Valeurs possibles:[1] NO (0), YES (1, par défaut).
type	string	Type d'interface. Valeurs possibles:[1] ZABBIX (1, par défaut), SNMP (2), IPMI (3), JMX (4).
useip	string	Indique s'il faut utiliser l'adresse IP comme interface pour se connecter à l'hôte (sinon, le DNS sera utilisé). Valeurs possibles:[1] NO (0), YES (1, par défaut).
ip	string	(requis pour les connexions IP) Adresse IP (IPv4 ou IPv6).
dns	string	(requis pour les connexions DNS) Nom DNS.
port	string	Numéro de port.
details		Élément racine pour les détails de l'interface.
version	string	Utiliser cette version SNMP. Valeurs possibles:[1] SNMPV1 (1), SNMP_V2C (2, par défaut), SNMP_V3 (3).
community	string	(requis pour les éléments SNMPv1 et SNMPv2) Communauté SNMP.

Element	Type	Description
max_repetitions	string	Valeur maximale de répétition pour les requêtes SNMP bulk natives (GetBulkRequest-PDUs). Pris en charge pour les éléments SNMPv2 et SNMPv3 (`discovery[]` et `walk[]`). Par défaut: 10.
contextname	string	Nom de contexte SNMPv3. Pris en charge pour les éléments SNMPv3.
securityname	string	Nom de sécurité SNMPv3. Pris en charge pour les éléments SNMPv3.
securitylevel	string	Niveau de sécurité SNMPv3. Pris en charge pour les éléments SNMPv3. Valeurs possibles:[1] NOAUTHNOPRIV (0, par défaut), AUTHNOPRIV (1), AUTHPRIV (2).
authprotocol	string	Protocole d'authentification SNMPv3. Pris en charge pour les éléments SNMPv3. Valeurs possibles:[1] MD5 (0, par défaut), SHA1 (1), SHA224 (2), SHA256 (3), SHA384 (4), SHA512 (5).
authpassphrase	string	Phrase secrète d'authentification SNMPv3. Pris en charge pour les éléments SNMPv3.
privprotocol	string	Protocole de confidentialité SNMPv3. Pris en charge pour les éléments SNMPv3. Valeurs possibles:[1] DES (0, par défaut), AES128 (1), AES192 (2), AES256 (3), AES192C (4), AES256C (5).
privpassphrase	string	Phrase secrète de confidentialité SNMPv3. Pris en charge pour les éléments SNMPv3.
bulk	string	Utiliser des requêtes bulk pour SNMP. Valeurs possibles:[1] NO (0), YES (1, par défaut).
interface_ref	string	Nom de référence de l'interface à utiliser dans les éléments (format: `if<N>`).

Note:
Voir aussi: Objet interface d'hôte (se référer à la propriété correspondante portant le même nom).

Éléments d'hôte

```
zabbix_export:
  (...)
  hosts:
    - host: 'Serveur MySQL'
      (...)
      items:
        (...)
        - name: 'Utilisation du cache binaire sur disque'
          type: DEPENDENT
          key: mysql.binlog_cache_disk_use
          value_type: FLOAT
          description: 'Nombre de transactions ayant utilisé un cache disque temporaire parce qu'elles ne p
          preprocessing: (voir le tableau ci-dessous)
          master_item:
            key: 'mysql.get_status_variables["{$MYSQL.DSN}","{$MYSQL.USER}","{$MYSQL.PASSWORD}"]'
          tags:
            - tag: component
              value: cache
        (...)
        - name: 'Utilisation du pool de tampons'
          type: CALCULATED
          key: mysql.buffer_pool_utilization
          value_type: FLOAT
          units: '%'
          params: |
            ( last(//mysql.innodb_buffer_pool_pages_total) -
            last(//mysql.innodb_buffer_pool_pages_free) ) /
            ( last(//mysql.innodb_buffer_pool_pages_total) +
            ( last(//mysql.innodb_buffer_pool_pages_total) = 0 ) ) * 100 *
```

```
        ( last(//mysql.innodb_buffer_pool_pages_total) > 0 )
      description: 'Rapport entre le nombre de pages utilisées et le nombre total de pages dans le poo
      tags:
        - tag: component
          value: memory
      triggers: (voir le tableau ci-dessous)
    (...)
    - name: Disponibilité
      type: DEPENDENT
      key: mysql.uptime
      units: uptime
      description: 'Nombre de secondes pendant lesquelles le serveur a été actif.'
      preprocessing: (voir le tableau ci-dessous)
      master_item:
        key: 'mysql.get_status_variables["{$MYSQL.DSN}","{$MYSQL.USER}","{$MYSQL.PASSWORD}"]'
      tags:
        - tag: component
          value: application
      triggers: (voir le tableau ci-dessous)
    (...)
```

Élément	Type	Description
name	string	(requis) Nom de l'élément.
type	string	Type d'élément. Valeurs possibles :[1] ZABBIX_PASSIVE (0, par défaut), TRAP (2), SIMPLE (3), INTERNAL (5), ZABBIX_ACTIVE (7), EXTERNAL (10), ODBC (11), IPMI (12), SSH (13), TELNET (14), CALCULATED (15), JMX (16), SNMP_TRAP (17), DEPENDENT (18), HTTP_AGENT (19), SNMP_AGENT (20), ITEM_TYPE_SCRIPT (21), ITEM_TYPE_BROWSER (22), ITEM_TYPE_NESTED (23), TELEMETRY_QUERY (24).
snmp_oid	string	(requis pour les éléments SNMP_AGENT) Identifiant d'objet SNMP.
key	string	(requis) Clé de l'élément.
delay	string	Intervalle de mise à jour de l'élément. Valeur par défaut : 1m. La valeur est toujours 0 pour les éléments TRAP.
history	string	Période (utilisant un suffixe temporel, une macro utilisateur ou une macro LLD) pendant laquelle les données d'historique doivent être conservées. Valeur par défaut : 31d.
trends	string	Période (utilisant un suffixe temporel, une macro utilisateur ou une macro LLD) pendant laquelle les données de tendances doivent être conservées. Valeur par défaut : 365d.
status	string	État de l'élément. Valeurs possibles :[1] ENABLED (0, par défaut), DISABLED (1).
value_type	string	Type de valeur reçue. Valeurs possibles :[1] FLOAT (0), CHAR (1), LOG (2), UNSIGNED (3, par défaut), TEXT (4), BINARY (5), JSON (6).
allowed_hosts	string	Liste d'adresses IP, séparées par des virgules, des hôtes autorisés à envoyer des données pour l'élément. Pris en charge pour les éléments TRAP et HTTP_AGENT.
units	string	Unités de la valeur reçue (bps, B, etc.).
params	text	Paramètres supplémentaires selon le type de l'élément (script exécuté pour les éléments SSH et TELNET ; requête SQL pour les éléments ODBC ; formule pour les éléments CALCULATED ; script pour les éléments ITEM_TYPE_SCRIPT et ITEM_TYPE_BROWSER).
ipmi_sensor	string	Capteur IPMI. Pris en charge pour les éléments IPMI.
authtype	string	Type d'authentification. Pris en charge pour les éléments SSH et HTTP_AGENT. Valeurs possibles pour les éléments SSH :[1] PASSWORD (0, par défaut), PUBLIC_KEY (1). Valeurs possibles pour les éléments HTTP_AGENT :[1] NONE (0, par défaut), BASIC (1), NTLM (2), Kerberos (3) ou Digest (4).

Élément	Type	Description
username	string	(requis pour les éléments SSH et TELNET) Nom d'utilisateur utilisé pour l'authentification. Pris en charge pour les éléments SIMPLE, ODBC, JMX et HTTP_AGENT. Lorsqu'il est utilisé pour les éléments JMX, `password` (voir ci-dessous) doit également être indiqué, ou les deux éléments doivent être laissés vides.
password	string	(requis pour les éléments SSH et TELNET) Mot de passe utilisé pour l'authentification. Pris en charge pour les éléments SIMPLE, ODBC, JMX et HTTP_AGENT. Lorsqu'il est utilisé pour les éléments JMX, `username` (voir ci-dessus) doit également être indiqué, ou les deux éléments doivent être laissés vides.
publickey	string	(requis pour les éléments SSH) Nom du fichier de clé publique.
privatekey	string	(requis pour les éléments SSH) Nom du fichier de clé privée.
description	text	Description de l'élément.
inventory_link	string	Champ d'inventaire de l'hôte alimenté par l'élément. Valeurs possibles :[1] NONE (0), ALIAS (4), etc. (voir Inventaire de l'hôte pour les champs pris en charge).
valuemap		Élément racine des mappages de valeurs de l'élément.
name	string	(requis) Nom du mappage de valeurs à utiliser pour l'élément.
logtimefmt	string	Format de l'heure dans les entrées de journal. Pris en charge pour les éléments de type de valeur LOG.
preprocessing		Élément racine du prétraitement des valeurs de l'élément.
step		Élément racine des étapes de prétraitement des valeurs des éléments d'hôte.
interface_ref	string	Référence à l'interface de l'hôte (format : `if<N>`).
jmx_endpoint	string	Point de terminaison JMX. Pris en charge pour les éléments JMX.
master_item		(requis pour les éléments DEPENDENT) Élément racine de l'élément maître de l'élément dépendant.
key	string	(requis) Clé de l'élément maître de l'élément dépendant.
timeout	string	Délai d'expiration de la demande d'interrogation des données de l'élément. Pris en charge pour la liste Délais d'expiration des types d'éléments.
url	string	(requis pour les éléments HTTP_AGENT) Chaîne d'URL.
query_fields		Élément racine des paramètres de requête. Pris en charge pour les éléments HTTP_AGENT.
name	string	(requis pour les éléments HTTP_AGENT) Nom du paramètre de requête.
value	string	Valeur du paramètre de requête. Pris en charge pour les éléments HTTP_AGENT.
parameters		Élément racine des paramètres définis par l'utilisateur. Pris en charge pour les éléments ITEM_TYPE_SCRIPT et ITEM_TYPE_BROWSER.
name	string	(requis pour les éléments ITEM_TYPE_SCRIPT et ITEM_TYPE_BROWSER) Nom du paramètre défini par l'utilisateur.
value	string	Valeur du paramètre défini par l'utilisateur. Pris en charge pour les éléments ITEM_TYPE_SCRIPT et ITEM_TYPE_BROWSER.
posts	string	Données du corps de la requête HTTP(S). Pris en charge pour les éléments HTTP_AGENT.
status_codes	string	Plages de codes d'état HTTP requis, séparées par des virgules. Pris en charge pour les éléments HTTP_AGENT.
follow_redirects	string	Suivre les redirections de la réponse lors de l'interrogation des données. Pris en charge pour les éléments HTTP_AGENT. Valeurs possibles :[1] NO (0), YES (1, par défaut).
post_type	string	Type du corps des données envoyées. Pris en charge pour les éléments HTTP_AGENT. Valeurs possibles :[1] RAW (0, par défaut), JSON (2), XML (3).
http_proxy	string	Chaîne de connexion au proxy HTTP(S). Pris en charge pour les éléments HTTP_AGENT.
headers		Élément racine des en-têtes de requête HTTP(S). Pris en charge pour les éléments HTTP_AGENT.
name	string	(requis pour les éléments HTTP_AGENT) Nom de l'en-tête.
value	string	(requis pour les éléments HTTP_AGENT) Valeur de l'en-tête.
retrieve_mode	string	Partie de la réponse à stocker. Pris en charge pour les éléments HTTP_AGENT. Valeurs possibles :[1] BODY (0, par défaut), HEADERS (1), BOTH (2).

Élément	Type	Description
request_method	string	Type de méthode de requête. Pris en charge pour les éléments HTTP_AGENT. Valeurs possibles :[1] GET (0, par défaut), POST (1), PUT (2), HEAD (3).
output_format	string	Méthode de traitement de la réponse. Pris en charge pour les éléments HTTP_AGENT. Valeurs possibles :[1] RAW (0, par défaut), JSON (1).
allow_traps	string	Autoriser le remplissage de la valeur de la même manière que pour un élément trapper. Pris en charge pour les éléments HTTP_AGENT. Valeurs possibles :[1] NO (0, par défaut), YES (1).
ssl_cert_file	string	Chemin du fichier de clé SSL publique. Pris en charge pour les éléments HTTP_AGENT.
ssl_key_file	string	Chemin du fichier de clé SSL privée. Pris en charge pour les éléments HTTP_AGENT.
ssl_key_password	string	Mot de passe du fichier de clé SSL. Pris en charge pour les éléments HTTP_AGENT.
verify_peer	string	Indique s'il faut vérifier que le certificat de l'hôte est authentique. Pris en charge pour les éléments HTTP_AGENT. Valeurs possibles :[1] NO (0, par défaut), YES (1).
verify_host	string	Indique s'il faut vérifier que le nom d'hôte de la connexion correspond à celui indiqué dans le certificat de l'hôte. Pris en charge pour les éléments HTTP_AGENT. Valeurs possibles :[1] NO (0, par défaut), YES (1).
time_shift	string	Décalage vers le passé (avant l'heure actuelle) de la fin de la plage temporelle interrogée par l'élément. Pris en charge pour les éléments TELEMETRY_QUERY.
lookback_limit	string	Période vers le passé (avant l'heure actuelle) pendant laquelle l'élément recherche des données lorsqu'il n'existe aucune requête précédente à poursuivre (par exemple, lors de la première exécution de l'élément), ou lorsque la requête précédente est antérieure à `delay` (par exemple, parce que le serveur Zabbix était arrêté). Pris en charge pour les éléments TELEMETRY_QUERY.
granularity	string	Durée d'une portion en laquelle la plage temporelle complète de la requête est divisée. Pris en charge pour les éléments TELEMETRY_QUERY.
query		(requis pour les éléments TELEMETRY_QUERY) Élément racine de la requête de télémétrie. Définit les données interrogées par l'élément auprès de la source de données, la manière dont elles sont filtrées et la manière dont le résultat est agrégé.
signal_type	string	(requis) Type de signal interrogé par l'élément. Pris en charge pour les éléments TELEMETRY_QUERY. Valeurs possibles :[1] TRACES (0), METRICS (1), LOGS (2).
metric_point_type	string	(requis pour le type de signal METRICS) Type de point de métrique interrogé par l'élément. Pris en charge pour les éléments TELEMETRY_QUERY. Valeurs possibles :[1] SUM (0), GAUGE (1), HISTOGRAM (2), EXPONENTIAL_HISTOGRAM (3).
columns		(requis) Élément racine des colonnes de données de télémétrie interrogées par l'élément. Cet élément contient les éléments suivants : `column (string)` - (requis) Nom de la colonne de données de télémétrie interrogée par l'élément. `attribute_key (string)` - (requis pour les colonnes contenant des paires clé-valeur) Nom de la clé interrogée par l'élément dans la colonne contenant des paires clé-valeur.

Élément	Type	Description
aggregated_columns		(requis) Élément racine des colonnes agrégées de données de télémétrie interrogées par l'élément. Cet élément contient les éléments suivants : `column` (`string`) - (requis pour les fonctions MIN, MAX, AVG, SUM et PERCENTILE) Nom de la colonne de données de télémétrie interrogée par l'élément. `function` (`string`) - (requis) Fonction appliquée par l'élément aux enregistrements de la colonne. Valeurs possibles :[1] MIN (1), MAX (2), AVG (3), COUNT (4), SUM (5), PERCENTILE (8). `parameters` (`array`) - (requis pour les fonctions PERCENTILE) Centile calculé par l'élément à partir des enregistrements de la colonne. `alias` (`string`) - (requis) Nom unique de la valeur calculée, utilisé pour l'identifier dans la valeur de l'élément.
filter		(requis) Élément racine du filtre de requête de télémétrie appliqué aux enregistrements de colonnes dans la requête. Cet élément contient les éléments suivants : `evaltype` (`string`) - (requis) Méthode d'évaluation de la condition de filtrage. Valeurs possibles :[1] AND_OR (0), AND (1), OR (2), FORMULA (3). `formula` (`string`) - (requis pour les méthodes d'évaluation FORMULA) Expression définie par l'utilisateur utilisée pour évaluer les conditions des filtres avec une expression personnalisée. `conditions` (`string`) - (requis) Élément racine des conditions de filtrage des données de télémétrie auxquelles un enregistrement d'une colonne de données de télémétrie doit satisfaire. L'élément `conditions` contient les éléments suivants : `column` (`string`) - (requis) Nom de la colonne de données de télémétrie à laquelle les conditions de filtrage s'appliquent. `attribute_key` (`string`) - (requis pour les colonnes contenant des paires clé-valeur) Nom de la clé à laquelle la condition s'applique, dans la colonne contenant des paires clé-valeur. `operator` (`string`) - (requis) Opérateur de la condition de filtrage. Valeurs possibles pour les colonnes à valeur unique :[1] EQUAL (0), NOT_EQUAL (1), LIKE (2), NOT_LIKE (3). Valeurs possibles pour les colonnes contenant des paires clé-valeur :[1] EQUAL (0), NOT_EQUAL (1), EXISTS (12). `formulaid` (`string`) - (requis pour les méthodes d'évaluation FORMULA) Identifiant unique arbitraire utilisé pour référencer la condition depuis une expression personnalisée. `value` (`string`) - (requis pour les opérations EQUAL, NOT_EQUAL, LIKE et NOT LIKE) Valeur à comparer.
tags		Élément racine des étiquettes de l'élément.
tag	string	(requis) Nom de l'étiquette.
value	string	Valeur de l'étiquette.
triggers		Élément racine des déclencheurs des éléments d'hôte.

Note:
Voir également : Objet item (consultez la propriété correspondante portant le même nom).

Étapes de prétraitement de la valeur de l'élément de l'hôte

```
zabbix_export:
  (...)
  hosts:
    - host: 'MySQL server'
      (...)
      items:
        (...)
        - name: 'Binlog cache disk use'
          (...)
          preprocessing:
```

```
            - type: JSONPATH
              parameters:
                - $.Binlog_cache_disk_use
            - type: DISCARD_UNCHANGED_HEARTBEAT
              parameters:
                - 6h
          (...)
```

Element	Type	Description
type	string	(obligatoire) Type de l'étape de prétraitement de la valeur de l'élément. Valeurs possibles:[1] MULTIPLIER (1), RTRIM (2), LTRIM (3), TRIM (4), REGEX (5), BOOL_TO_DECIMAL (6), OCTAL_TO_DECIMAL (7), HEX_TO_DECIMAL (8), SIMPLE_CHANGE (9, calculé: valeur reçue - valeur précédente), CHANGE_PER_SECOND (10, calculé: (valeur reçue - valeur précédente)/(heure actuelle - heure de la dernière vérification)), XMLPATH (11), JSONPATH (12), IN_RANGE (13), MATCHES_REGEX (14), NOT_MATCHES_REGEX (15), CHECK_JSON_ERROR (16), CHECK_XML_ERROR (17), CHECK_REGEX_ERROR (18), DISCARD_UNCHANGED (19), DISCARD_UNCHANGED_HEARTBEAT (20), JAVASCRIPT (21), PROMETHEUS_PATTERN (22), PROMETHEUS_TO_JSON (23), CSV_TO_JSON (24), STR_REPLACE (25), CHECK_NOT_SUPPORTED (26), XML_TO_JSON (27), SNMP_WALK_VALUE (28), SNMP_WALK_TO_JSON (29), SNMP_GET_VALUE (30).
parameters		(obligatoire) Élément racine pour les paramètres de l'étape de prétraitement de la valeur de l'élément.
parameter	string	Paramètre individuel de l'étape de prétraitement de la valeur de l'élément.
error_handler	string	Type d'action utilisé en cas d'échec de l'étape de prétraitement. Valeurs possibles:[1] ORIGINAL_ERROR (0, par défaut), DISCARD_VALUE (1), CUSTOM_VALUE (2), CUSTOM_ERROR (3).
error_handler_params	string	Paramètres du gestionnaire d'erreurs.

Note:
Voir aussi: Objet de prétraitement d'élément (reportez-vous à la propriété correspondante portant le même nom).

Déclencheurs d'élément d'hôte

```
zabbix_export:
  (...)
  hosts:
    - host: 'MySQL server'
      (...)
      items:
        (...)
        - name: Uptime
          (...)
          triggers:
            - expression: 'nodata(/MySQL server/mysql.uptime,30m)=1'
              name: 'MySQL: Failed to fetch info data'
              event_name: 'MySQL: Failed to fetch info data (or no data for 30m)'
              priority: INFO
              description: 'Zabbix n'a reçu aucune donnée pour les éléments au cours des 30 dernières minu
              dependencies:
                - name: 'MySQL: Service is down'
                  expression: 'last(/MySQL server/mysql.ping["{$MYSQL.DSN}","{$MYSQL.USER}","{$MYSQL.PASSW
              tags:
                - tag: scope
                  value: availability
            - expression: 'last(/MySQL server/mysql.uptime)<10m'
              name: 'MySQL: Service has been restarted'
              event_name: 'MySQL: Service has been restarted (uptime < 10m)'
              priority: INFO
              description: 'Le temps de fonctionnement de MySQL est inférieur à 10 minutes.'
              tags:
```

```
            - tag: scope
              value: notice
      (...)
```

Element	Type	Description
uuid	string	(required) Identifiant unique de ce déclencheur.
expression	string	(required) Expression du déclencheur.
recovery_mode	string	Base de génération des événements OK. Valeurs possibles:[1] EXPRESSION (0, default), RECOVERY_EXPRESSION (1), NONE (2).
recovery_expression	string	Expression de récupération du déclencheur.
correlation_mode	string	Mode de corrélation (aucune corrélation d'événements ou corrélation d'événements par tag). Valeurs possibles:[1] DISABLED (0, default), TAG_VALUE (1).
correlation_tag	string	Nom du tag à utiliser pour la corrélation des événements.
name	string	(required) Nom du déclencheur.
event_name	string	Nom de l'événement.
opdata	string	Données opérationnelles.
url_name	string	Libellé de l'URL associée au déclencheur.
url	string	URL associée au déclencheur.
status	string	Statut du déclencheur. Valeurs possibles:[1] ENABLED (0, default), DISABLED (1).
priority	string	Sévérité du déclencheur. Valeurs possibles:[1] NOT_CLASSIFIED (0, default), INFO (1), WARNING (2), AVERAGE (3), HIGH (4), DISASTER (5).
description	text	Description du déclencheur.
type	string	Type de génération d'événements (un seul événement de problème ou plusieurs événements de problème). Valeurs possibles:[1] SINGLE (0, default), MULTIPLE (1).
manual_close	string	Fermeture manuelle des événements de problème. Valeurs possibles:[1] NO (0, default), YES (1).
dependencies		Élément racine pour les dépendances.
name	string	(required) Nom du déclencheur dépendant.
expression	string	(required) Expression du déclencheur dépendant.
recovery_expression	string	Expression de récupération du déclencheur dépendant.
tags		Élément racine pour les tags du déclencheur.
tag	string	(required) Nom du tag.
value	string	Valeur du tag.

Note:
Voir aussi: Objet Trigger (reportez-vous à la propriété correspondante portant le même nom).

Règles de découverte de bas niveau de l'hôte

```
zabbix_export:
  (...)
  hosts:
    - host: 'MySQL server'
      (...)
      discovery_rules:
        - name: 'Database discovery'
          key: 'mysql.db.discovery["{$MYSQL.DSN}","{$MYSQL.USER}","{$MYSQL.PASSWORD}"]'
          delay: 1h
          filter: (see table below)
          description: 'Scanning databases in DBMS.'
          interface_ref: if1
          item_prototypes:
            - name: 'Size of database {#DATABASE}'
              key: 'mysql.db.size["{$MYSQL.DSN}","{$MYSQL.USER}","{$MYSQL.PASSWORD}","{#DATABASE}"]'
              delay: 5m
              units: B
              description: 'Database size.'
```

```
            preprocessing:
              - type: DISCARD_UNCHANGED_HEARTBEAT
                parameters:
                  - 1h
            interface_ref: if1
            tags:
              - tag: component
                value: storage
              - tag: database
                value: '{#DATABASE}'
          lld_macro_paths:
            - lld_macro: '{#DATABASE}'
              path: $.Database
          preprocessing:
            - type: DISCARD_UNCHANGED_HEARTBEAT
              parameters:
                - 1d
        (...)
```

Attention:
La plupart des éléments des règles de découverte de bas niveau de l'hôte sont identiques à ceux des éléments d'hôte. Le tableau ci-dessous décrit les éléments qui diffèrent des éléments d'hôte.

Element	Type	Description
type	string	Type d'élément. Valeurs possibles:[1] ZABBIX_PASSIVE (0, par défaut), TRAP (2), SIMPLE (3), INTERNAL (5), ZABBIX_ACTIVE (7), EXTERNAL (10), ODBC (11), IPMI (12), SSH (13), TELNET (14), JMX (16), DEPENDENT (18), HTTP_AGENT (19), SNMP_AGENT (20), ITEM_TYPE_SCRIPT (21), ITEM_TYPE_BROWSER (22).
key	string	Clé de la règle de découverte de bas niveau.
filter		Élément racine pour les filtres des règles de découverte de bas niveau de l'hôte.
lifetime	string	Période de temps (en secondes, suffixe de temps, ou macro utilisateur), après laquelle les ressources découvertes qui n'existent plus seront supprimées. Par défaut: 7d.
lifetime_type	string	Scénario de suppression des ressources LLD perdues. Valeurs possibles: DELETE_NEVER, DELETE_IMMEDIATELY, DELETE_AFTER.
enabled_lifetime	string	Période de temps (en secondes, suffixe de temps, ou macro utilisateur), après laquelle les ressources découvertes qui n'existent plus seront désactivées.
enabled_lifetime_type	string	Scénario de désactivation des ressources LLD perdues. Valeurs possibles: DISABLE_NEVER, DISABLE_IMMEDIATELY, DISABLE_AFTER.
item_prototypes		Élément racine pour les prototypes d'éléments d'hôte, qui sont identiques à ceux des éléments d'hôte.
trigger_prototypes		Élément racine pour les prototypes de déclencheurs d'hôte, qui sont identiques à ceux des déclencheurs d'éléments d'hôte.
graph_prototypes		Élément racine pour les prototypes de graphiques d'hôte, qui sont identiques à ceux des graphiques d'hôte.
host_prototypes		Élément racine pour les prototypes d'hôtes, qui sont identiques à ceux des hôtes.
parent_discovery_rule		Élément racine pour la règle de découverte de bas niveau parente (ou le prototype de règle) du prototype de règle de découverte de bas niveau. Cette propriété indique qu'il s'agit d'un prototype de règle LLD, enfant direct de la règle ou du prototype de règle référencé.
key	string	Clé de la règle de découverte de bas niveau parente (ou du prototype de règle).
master_item	string	(obligatoire pour les règles DEPENDENT) Élément racine pour l'élément maître de la règle dépendante.
lld_macro_paths		Élément racine pour les chemins des macros de la règle de découverte de bas niveau.
lld_macro	string	(obligatoire) Nom de la macro de la règle de découverte de bas niveau.
path	string	(obligatoire) Sélecteur de la valeur qui sera attribuée à la macro correspondante.
preprocessing		Élément racine pour le prétraitement de la valeur de la règle de découverte de bas niveau.

Element	Type	Description
step		Élément racine pour les éléments d'étape de prétraitement de la valeur de la règle de découverte de bas niveau, qui sont identiques à ceux des étapes de prétraitement de la valeur d'élément d'hôte, mais avec moins de valeurs possibles. Voir aussi: objet de prétraitement de règle LLD.
overrides		Élément racine pour les règles de remplacement de la règle de découverte de bas niveau.
name	string	(obligatoire) Nom unique du remplacement.
step	string	(obligatoire) Numéro d'ordre unique du remplacement.
stop	string	Arrêter le traitement des remplacements suivants si correspond.
filter		Élément racine pour les éléments de filtre des règles de remplacement de la règle de découverte de bas niveau, qui sont identiques à ceux des filtres des règles de découverte de bas niveau de l'hôte.
operations		Élément racine pour les opérations de remplacement des règles de découverte de bas niveau de l'hôte.

Note:
Voir aussi: objet de règle LLD (reportez-vous à la propriété correspondante portant le même nom).

Filtres de règle de découverte bas niveau de l'hôte

```
zabbix_export:
  (...)
  hosts:
    - host: 'MySQL server'
      (...)
      discovery_rules:
        - name: 'Database discovery'
          (...)
          filter:
            evaltype: AND
            conditions:
              - macro: '{#DATABASE}'
                value: '{$MYSQL.DBNAME.MATCHES}'
              - macro: '{#DATABASE}'
                value: '{$MYSQL.DBNAME.NOT_MATCHES}'
                operator: NOT_MATCHES_REGEX
          (...)
```

Element	Type	Description
evaltype	string	Méthode d'évaluation de la condition de filtre à remplacer. Valeurs possibles :[1] AND_OR (0, par défaut), AND (1), OR (2), FORMULA (3).
formula	string	Formule de calcul personnalisée pour les conditions de filtre.
conditions		Élément racine pour les conditions de filtre.
macro	string	(obligatoire) Nom de la macro de découverte bas niveau sur laquelle effectuer la vérification.
value	string	Valeur à comparer.
operator	string	Opérateur de condition. Valeurs possibles :[1] MATCHES_REGEX (8, par défaut), NOT_MATCHES_REGEX (9).
formulaid	string	(obligatoire) ID unique arbitraire utilisé pour référencer une condition depuis l'expression personnalisée. Ne peut contenir que des lettres majuscules. L'ID doit être défini par l'utilisateur lors de la modification des conditions de filtre, mais sera généré à nouveau lors d'une demande ultérieure.

Note:
Voir aussi : Objet de filtre de règle LLD (reportez-vous à la propriété correspondante portant le même nom).

Opérations de remplacement de règle de découverte bas niveau de hôte

Element	Type	Description
operationobject	string	Objet auquel l'opération est appliquée. Valeurs possibles:[1] ITEM_PROTOTYPE (0), TRIGGER_PROTOTYPE (1), GRAPH_PROTOTYPE (2), HOST_PROTOTYPE (3), DISCOVERY_PROTOTYPE (4).
operator	string	Opérateur de condition de remplacement. Valeurs possibles:[1] EQUAL (1), NOT_EQUAL (2), LIKE (3), NOT_LIKE (4), REGEXP (5), NOT_REGEXP (6).
value	string	Expression régulière ou chaîne pour l'opérateur de condition de remplacement.
status	string	Statut de l'objet après l'opération de remplacement.
discover	string	Indique si l'objet est ajouté à la suite de la découverte.
delay	string	Intervalle de mise à jour défini pour le prototype d'élément après l'opération de remplacement.
history	string	Période de conservation de l'historique définie pour le prototype d'élément après l'opération de remplacement.
trends	string	Période de conservation des tendances définie pour le prototype d'élément après l'opération de remplacement.
severity	string	Niveau de gravité du prototype de déclencheur défini après l'opération de remplacement.
tags		Élément racine pour les balises définies pour l'objet après l'opération de remplacement.
tag	string	(obligatoire) Nom de la balise.
value	string	Valeur de la balise.
templates		Élément racine pour les modèles liés au prototype d'hôte après l'opération de remplacement.
name	string	(obligatoire) Nom du modèle.
inventory_mode	string	Mode d'inventaire du prototype d'hôte défini après l'opération de remplacement.

Note:
Voir aussi: Objet d'opération de remplacement de règle LLD (reportez-vous à la propriété correspondante portant le même nom).

Scénarios web de l'hôte

Element	Type	Description
uuid	string	(required) Identifiant unique de ce scénario web.
name	string	(required) Nom du scénario web.
delay	string	Fréquence (en utilisant des secondes, un suffixe de temps, ou une macro utilisateur) d'exécution du scénario web. Par défaut: 1m.
attempts	integer	Nombre de tentatives pour l'exécution des étapes du scénario web. Valeurs possibles: 1-10 (par défaut: 1).
agent	string	Agent client. Zabbix se fera passer pour le navigateur sélectionné. Cela est utile lorsqu'un site web renvoie un contenu différent selon les navigateurs. Par défaut: Zabbix.
http_proxy	string	Proxy qui sera utilisé par le scénario web, indiqué sous la forme: `http://[username[:password]@]proxy.example.com[:port]`
variables		Élément racine pour les variables du scénario web pouvant être utilisées dans les étapes du scénario.
name	string	(required) Nom de la variable.
value	text	(required) Valeur de la variable.
headers		Élément racine pour les en-têtes HTTP à envoyer lors de l'exécution d'une requête. Les en-têtes doivent être listés en utilisant la même syntaxe que celle qu'ils auraient dans le protocole HTTP.
name	string	(required) Nom de l'en-tête.
value	text	(required) Valeur de l'en-tête.
status	string	Statut du scénario web. Valeurs possibles:[1] ENABLED (0, par défaut), DISABLED (1).
authentication	string	Méthode d'authentification. Valeurs possibles:[1] NONE (0, par défaut), BASIC (1), NTLM (2), Kerberos (3), ou Digest (4).
http_user	string	Nom d'utilisateur utilisé pour l'authentification BASIC (HTTP), NTLM, Kerberos ou Digest.

Element	Type	Description
http_password	string	Mot de passe utilisé pour l'authentification BASIC (HTTP), NTLM, Kerberos ou Digest.
verify_peer	string	Vérifier le certificat SSL du serveur web. Valeurs possibles:[1] NO (0, par défaut), YES (1).
verify_host	string	Vérifier que le champ *Common Name* ou le champ *Subject Alternate Name* du certificat du serveur web correspond. Valeurs possibles:[1] NO (0, par défaut), YES (1).
ssl_cert_file	string	Nom du fichier de certificat SSL utilisé pour l'authentification du client (doit être au format PEM).
ssl_key_file	string	Nom du fichier de clé privée SSL utilisé pour l'authentification du client (doit être au format PEM).
ssl_key_password	string	Mot de passe du fichier de clé privée SSL.
steps		Élément racine (obligatoire) pour les étapes du scénario web de l'hôte.
tags		Élément racine pour les tags du scénario web.
tag	string	(required) Nom du tag.
value	string	Valeur du tag.

Note:
Voir aussi: Objet scénario web (reportez-vous à la propriété correspondante portant le même nom).

Étapes du scénario web de l'hôte

Element	Type	Description
name	string	(obligatoire) Nom de l'étape du scénario web.
url	string	(obligatoire) URL à surveiller.
query_fields		Élément racine pour les paramètres de requête (un tableau de champs HTTP à ajouter à l'URL lors de l'exécution d'une requête).
name	string	(obligatoire) Nom du paramètre de requête.
value	string	Valeur du paramètre de requête.
posts		Élément racine pour les variables HTTP POST (une chaîne (données brutes du POST) ou un tableau de champs HTTP (données de champs de formulaire)).
name	string	(obligatoire) Nom du champ POST.
value	string	(obligatoire) Valeur du champ POST.
variables		Élément racine des variables au niveau de l'étape (macros) qui doivent être appliquées après cette étape. Si la valeur de la variable a le préfixe 'regex:', alors sa valeur est extraite des données renvoyées par cette étape selon le modèle d'expression régulière suivant le préfixe 'regex:'
name	string	(obligatoire) Nom de la variable.
value	text	(obligatoire) Valeur de la variable.
headers		Élément racine pour les en-têtes HTTP à envoyer lors de l'exécution d'une requête.
name	string	(obligatoire) Nom de l'en-tête.
value	text	(obligatoire) Valeur de l'en-tête.
follow_redirects	string	Suivre les redirections HTTP. Valeurs possibles :[1] NO (0), YES (1, par défaut).
retrieve_mode	string	Mode de récupération de la réponse HTTP. Valeurs possibles :[1] BODY (0, par défaut), HEADERS (1), BOTH (2).
timeout	string	Délai d'attente (en secondes, suffixe de temps, ou macro utilisateur) de l'exécution de l'étape. Par défaut : 15s.
required	string	Texte qui doit être présent dans la réponse (ignoré si vide).
status_codes	string	Liste de codes d'état HTTP acceptés, séparés par des virgules (par exemple, 200-201,210-299; ignoré si vide).

Note:
Voir aussi : Objet d'étape de scénario web (reportez-vous à la propriété correspondante portant le même nom).

Graphiques de l'hôte

```
zabbix_export:
  (...)
  graphs:
    - name: 'MySQL: Bandwidth'
      graph_items: (see table below)
    (...)
```

Element	Type	Description
uuid	string	Identifiant unique du graphique.
name	string	(obligatoire) Nom du graphique.
width	integer	Largeur du graphique, en pixels. Utilisé pour l'aperçu et pour les graphiques en camembert/explosés. Valeurs possibles:[1] 20-65535 (par défaut: 900).
height	integer	Hauteur du graphique, en pixels. Utilisé pour l'aperçu et pour les graphiques en camembert/explosés. Valeurs possibles:[1] 20-65535 (par défaut: 900).
yaxismin	double	Valeur minimale de l'axe Y. Pris en charge pour la valeur minimale FIXED de l'axe Y. Par défaut: 0.
yaxismax	double	Valeur maximale de l'axe Y. Pris en charge pour la valeur minimale FIXED de l'axe X. Par défaut: 0.
show_work_period	string	Mettre en évidence les heures non ouvrées. Pris en charge pour les graphiques NORMAL et STACKED. Valeurs possibles:[1] NO (0), YES (1, par défaut).
show_triggers	string	Afficher les valeurs simples des déclencheurs sous forme de ligne. Pris en charge pour les graphiques NORMAL et STACKED. Valeurs possibles:[1] NO (0), YES (1, par défaut).
type	string	Type de graphique. Valeurs possibles:[1] NORMAL (0, par défaut), STACKED (1), PIE (2), EXPLODED (3).
show_legend	string	Afficher la légende du graphique. Valeurs possibles:[1] NO (0), YES (1, par défaut).
show_3d	string	Activer le style 3D. Pris en charge pour les graphiques NORMAL et STACKED. Valeurs possibles:[1] NO (0, par défaut), YES (1).
percent_left	double	Afficher la ligne de percentile pour l'axe gauche. Pris en charge pour les graphiques NORMAL. Par défaut: 0.
percent_right	double	Afficher la ligne de percentile pour l'axe droit. Pris en charge pour les graphiques NORMAL. Par défaut: 0.
ymin_type_1	string	Valeur minimale de l'axe Y. Pris en charge pour les graphiques NORMAL et STACKED. Valeurs possibles:[1] CALCULATED (0, par défaut), FIXED (1), ITEM (2).
ymin_item_1		(obligatoire si `ymin_type_1` est défini sur `ITEM`) Élément racine pour les détails de chaque élément.
host	string	(obligatoire) Hôte de l'élément.
key	string	(obligatoire) Clé de l'élément.
ymax_type_1	string	Valeur maximale de l'axe Y. Pris en charge pour les graphiques NORMAL et STACKED. Valeurs possibles:[1] CALCULATED (0, par défaut), FIXED (1), ITEM (2).
ymax_item_1		(obligatoire si `ymax_type_1` est défini sur `ITEM`) Élément racine pour les détails de chaque élément.
host	string	(obligatoire) Hôte de l'élément.
key	string	(obligatoire) Clé de l'élément.
graph_items		(obligatoire) Élément racine pour les éléments de graphique de l'hôte.

Note:
Voir aussi: Objet Graph (reportez-vous à la propriété correspondante portant le même nom).

Éléments du graphique de l'hôte

```
zabbix_export:
  (...)
  graphs:
    - name: 'MySQL: Bandwidth'
      graph_items:
        - drawtype: GRADIENT_LINE
          color: 199C0D
          item:
            host: 'MySQL server'
            key: mysql.bytes_received.rate
        - sortorder: '1'
          drawtype: GRADIENT_LINE
          color: F63100
          item:
            host: 'MySQL server'
            key: mysql.bytes_sent.rate
    (...)
```

Élément	Type	Description
sortorder	integer	Ordre de dessin. La valeur la plus petite est dessinée en premier. Peut être utilisé pour dessiner des lignes ou des zones derrière (ou devant) une autre.
drawtype	string	Style de dessin de l'élément du graphique. Pris en charge pour les graphiques NORMAL. Valeurs possibles :[1] SINGLE_LINE (0, par défaut), FILLED_REGION (1), BOLD_LINE (2), DOTTED_LINE (3), DASHED_LINE (4), GRADIENT_LINE (5).
color	string	Couleur de l'élément (6 symboles, hexadécimal).
yaxisside	string	Côté du graphique où l'échelle Y de l'élément du graphique sera dessinée. Pris en charge pour les graphiques NORMAL et STACKED. Valeurs possibles :[1] LEFT (0, par défaut), RIGHT (1).
calc_fnc	string	Données à dessiner si plus d'une valeur existe pour un élément. Valeurs possibles :[1] MIN (1), AVG (2, par défaut), MAX (4), ALL (7; minimum, moyenne et maximum; pris en charge pour les graphiques simples), LAST (9, pris en charge pour les graphiques en camembert/explosés).
type	string	Type d'élément du graphique. Valeurs possibles :[1] SIMPLE (0, par défaut), GRAPH_SUM (2; la valeur de l'élément représente l'ensemble du camembert; pris en charge pour les graphiques en camembert/explosés).
item		(obligatoire) Élément individuel.
host	string	(obligatoire) Hôte de l'élément.
key	string	(obligatoire) Clé de l'élément.

Note:
Voir aussi: Objet Graph item (reportez-vous à la propriété correspondante portant le même nom).

Cartes de valeurs de l'hôte

```
zabbix_export:
  (...)
  hosts:
    - host: 'MySQL server'
      (...)
      valuemaps:
        - name: Example value map
          mappings:
            - value: '1'
              newvalue: Example value
  (...)
```

Element	Type	Description
uuid	string	(required) Identifiant unique pour cette carte de valeurs.

Element	Type	Description
name	string	(required) Nom de la carte de valeurs.
mapping		Élément racine pour les mappages.
type	string	Type de correspondance du mappage. Valeurs possibles:[1] EQUAL (0, par défaut), GREATER_OR_EQUAL (2), LESS_OR_EQUAL (3), IN_RANGE (4), REGEXP (5), DEFAULT (6).
value	string	Valeur d'origine.
newvalue	string	(required) Valeur vers laquelle la valeur d'origine est mappée.

Note:
Voir aussi: Objet carte de valeurs (se référer à la propriété correspondante portant le même nom).

Notes de bas de page [1] Les valeurs entières de l'API entre parenthèses, par exemple ENABLED (0), sont mentionnées uniquement à titre de référence. Pour plus d'informations, consultez la page de l'objet API liée dans l'entrée du tableau ou à la fin de chaque section.

Cartes réseau

Vue d'ensemble

La carte réseau export contient :

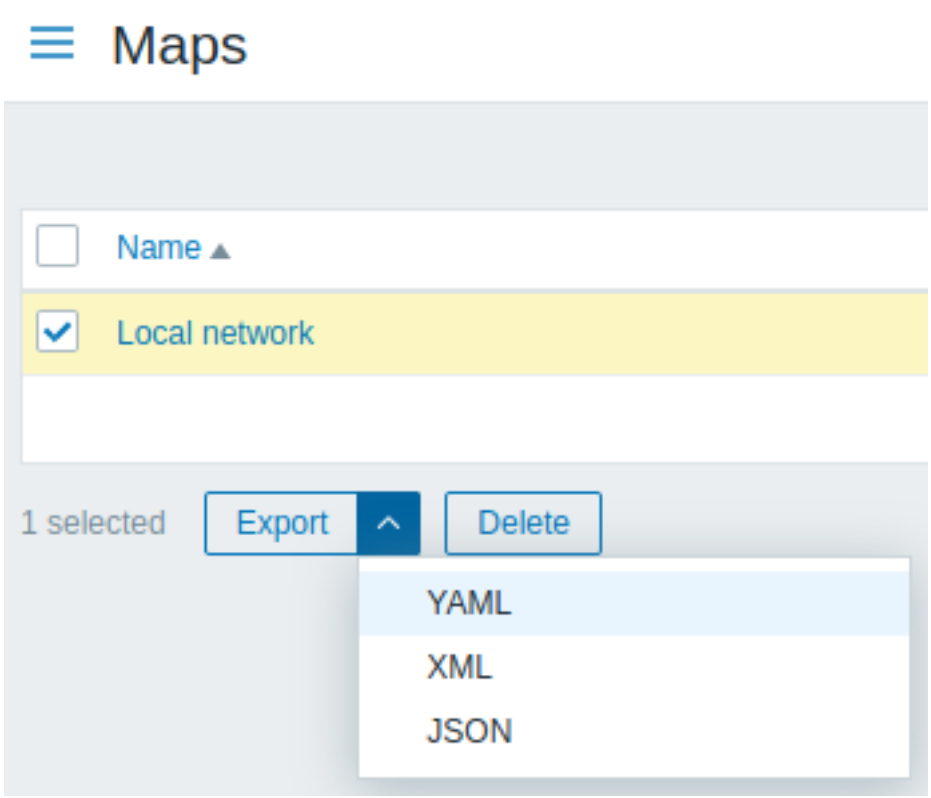
- Toutes les images associées
- La structure de la carte (tous les paramètres de la carte, tous les éléments qu'elle contient avec leurs paramètres, les liens de carte et les indicateurs d'état des liens de carte)

Warning:
Aucun groupe d'hôtes, hôte, déclencheur, autre carte ou autre élément pouvant être associé à la carte exportée n'est exporté. Ainsi, si au moins l'un des éléments auxquels la carte fait référence est manquant, son importation échouera.

Exportation

Pour exporter des cartes réseau, procédez comme suit :

1. Accédez à *Monitoring* > *Maps*.
2. Cochez les cases des cartes réseau à exporter.
3. Cliquez sur *Export* sous la liste pour exporter au format YAML par défaut, ou cliquez sur la flèche pour sélectionner un autre format d'exportation.



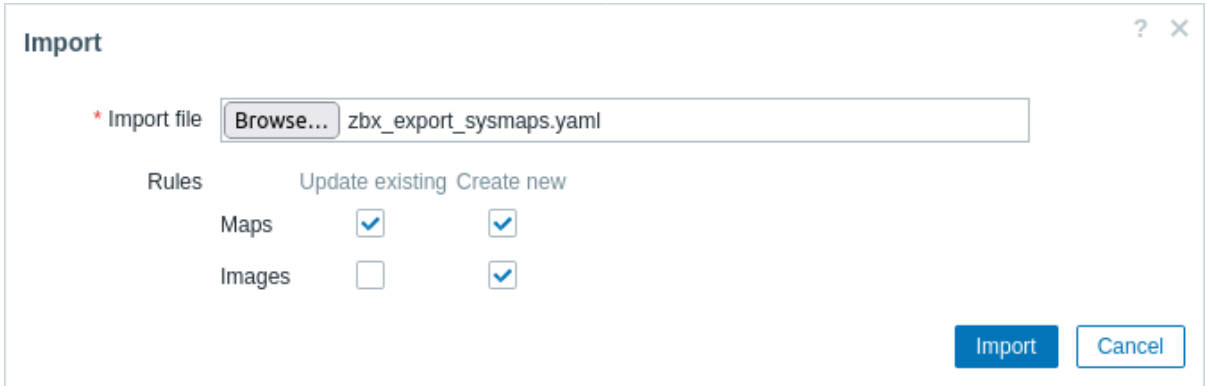
Selon le format sélectionné, les cartes sont exportées vers un fichier local avec un nom par défaut :

- `zabbix_export_maps.yaml` - en export YAML ;
- `zabbix_export_maps.xml` - en export XML ;
- `zabbix_export_maps.json` - en export JSON.

Importation

Pour importer des cartes réseau, procédez comme suit :

1. Accédez à *Monitoring* > *Maps*.
2. Cliquez sur *Import* dans le coin supérieur droit.
3. Sélectionnez le fichier d'importation.
4. Cochez les cases correspondant aux règles d'importation que vous souhaitez appliquer (voir ci-dessous).
5. Cliquez sur *Import* dans le coin inférieur droit du formulaire de configuration.



Règles d'importation :

Rule	Description
Update existing	Les cartes existantes seront mises à jour à l'aide des données du fichier d'importation. Dans le cas contraire, elles ne seront pas mises à jour.
Create new	De nouvelles cartes seront créées à l'aide des données du fichier d'importation. Dans le cas contraire, elles ne seront pas créées.

Si vous décochez les deux options relatives aux cartes et cochez les options correspondantes pour les images, seules les images seront importées. L'importation d'images est uniquement disponible pour les utilisateurs *Super admin*.

Vous ne pouvez pas examiner les modifications apportées aux cartes réseau pendant l'importation. Un message indiquant la réussite ou l'échec de l'importation s'affichera dans l'interface.

Warning:
Si vous remplacez une image existante, cela affectera toutes les cartes qui utilisent cette image.

Format d'export

Export vers YAML :

```yaml
zabbix_export:
  version: '8.0'
  images:
    - name: Zabbix_server_3D_(128)
      imagetype: '1'
      encodedImage: iVBOR...5CYII=
  maps:
    - name: 'Réseau local'
      width: '680'
      height: '200'
      label_type: '0'
      label_location: '0'
      highlight: '1'
      expandproblem: '1'
      markelements: '1'
      show_unack: '0'
      severity_min: '0'
      show_suppressed: '0'
      grid_size: '50'
      grid_show: '1'
      grid_align: '1'
      label_format: '0'
      label_type_host: '2'
      label_type_hostgroup: '2'
      label_type_trigger: '2'
      label_type_map: '2'
      label_type_image: '2'
      label_string_host: ''
      label_string_hostgroup: ''
      label_string_trigger: ''
      label_string_map: ''
      label_string_image: ''
      expand_macros: '1'
      background: {  }
      iconmap: {  }
      urls: {  }
      selements:
        - elementtype: '0'
          elements:
            - host: 'Zabbix server'
          label: |
            {HOST.NAME}
            {HOST.CONN}
          label_location: '0'
          x: '111'
          'y': '61'
          elementsubtype: '0'
          areatype: '0'
          width: '200'
          height: '200'
          viewtype: '0'
          use_iconmap: '0'
          selementid: '1'
          icon_off:
```

```
            name: Zabbix_server_3D_(128)
          icon_on: {  }
          icon_disabled: {  }
          icon_maintenance: {  }
          urls: {  }
          evaltype: '0'
      shapes:
        - type: '0'
          x: '0'
          'y': '0'
          width: '680'
          height: '15'
          text: '{MAP.NAME}'
          font: '9'
          font_size: '11'
          font_color: '000000'
          text_halign: '0'
          text_valign: '0'
          border_type: '0'
          border_width: '0'
          border_color: '000000'
          background_color: ''
          zindex: '0'
      lines: {  }
      links: {  }
```

Éléments exportés

Les éléments exportés sont expliqués dans le tableau ci-dessous.

Element	Type	Description
images		Élément racine pour les images.
name	string	Nom unique de l'image.
imagetype	integer	Type d'image. Valeurs possibles : 1 - Image; 2 - Arrière-plan.
encodedImage	string	Image encodée en Base64.
maps		Élément racine pour les maps.

Maps

Element	Type	Description
name	string	Nom unique de la carte.
width	integer	Largeur de la carte, en pixels.
height	integer	Hauteur de la carte, en pixels.
label_type	integer	Type d'étiquette de l'élément de carte. Valeurs possibles : 0 - Étiquette; 1 - Adresse IP de l'hôte; 2 - Nom de l'élément; 3 - Statut uniquement; 4 - Rien.
label_location	integer	Emplacement de l'étiquette de l'élément de carte par défaut. Valeurs possibles : 0 - Bas; 1 - Gauche; 2 - Droite; 3 - Haut.

Element	Type	Description
highlight	integer	Activer la mise en évidence des icônes pour les déclencheurs actifs et les statuts des hôtes. Valeurs possibles : 0 - Non; 1 - Oui.
expandproblem	integer	Afficher le déclencheur de problème pour les éléments ayant un seul problème. Valeurs possibles : 0 - Non; 1 - Oui.
markelements	integer	Mettre en évidence les éléments de la carte dont le statut a récemment changé. Valeurs possibles : 0 - Non; 1 - Oui.
show_unack	integer	Affichage des problèmes. Valeurs possibles : 0 - Nombre total de problèmes; 1 - Nombre de problèmes non acquittés; 2 - Nombre de problèmes acquittés et non acquittés séparément.
severity_min	integer	Sévérité minimale du déclencheur à afficher sur la carte par défaut. Valeurs possibles : 0 - Non classé; 1 - Information; 2 - Avertissement; 3 - Moyen; 4 - Élevé; 5 - Désastre.
show_suppressed	integer	Afficher les problèmes qui seraient autrement supprimés (non affichés) en raison de la maintenance de l'hôte. Valeurs possibles : 0 - Non; 1 - Oui.
grid_size	integer	Taille des cellules de la grille de la carte en pixels. Pris en charge si `grid_show` est défini sur 0. Valeurs possibles : 20, 40, 50, 75 ou 100.
grid_show	integer	Afficher une grille dans la configuration de la carte. Valeurs possibles : 0 - Oui; 1 - Non.
grid_align	integer	Aligner automatiquement les icônes dans la configuration de la carte. Valeurs possibles : 0 - Oui; 1 - Non.
label_format	integer	Utiliser une configuration avancée des étiquettes. Valeurs possibles : 0 - Non; 1 - Oui.
label_type_host	integer	Afficher l'étiquette comme étiquette d'hôte. Pris en charge si `label_format` est défini sur 1. Valeurs possibles : 0 - Étiquette; 1 - Adresse IP de l'hôte; 2 - Nom de l'élément; 3 - Statut uniquement; 4 - Rien; 5 - Étiquette personnalisée.

Element	Type	Description
label_type_hostgroup	integer	Afficher l'étiquette comme étiquette de groupe d'hôtes. Pris en charge si `label_format` est défini sur 1. Valeurs possibles : 0 - Étiquette; 2 - Nom de l'élément; 3 - Statut uniquement; 4 - Rien; 5 - Étiquette personnalisée.
label_type_trigger	integer	Afficher l'étiquette comme étiquette de déclencheur. Pris en charge si `label_format` est défini sur 1. Valeurs possibles : 0 - Étiquette; 2 - Nom de l'élément; 3 - Statut uniquement; 4 - Rien; 5 - Étiquette personnalisée.
label_type_map	integer	Afficher l'étiquette comme étiquette de carte. Pris en charge si `label_format` est défini sur 1. Valeurs possibles : 0 - Étiquette; 2 - Nom de l'élément; 3 - Statut uniquement; 4 - Rien; 5 - Étiquette personnalisée.
label_type_image	integer	Afficher l'étiquette comme étiquette d'image. Pris en charge si `label_format` est défini sur 1. Valeurs possibles : 0 - Étiquette; 2 - Nom de l'élément; 4 - Rien; 5 - Étiquette personnalisée.
label_string_host	string	Étiquette personnalisée pour les éléments d'hôte. Pris en charge si `label_type_host` est défini sur 5.
label_string_hostgroup	string	Étiquette personnalisée pour les éléments de groupe d'hôtes. Pris en charge si `label_type_hostgroup` est défini sur 5.
label_string_trigger	string	Étiquette personnalisée pour les éléments de déclencheur. Pris en charge si `label_type_trigger` est défini sur 5.
label_string_map	string	Étiquette personnalisée pour les éléments de carte. Pris en charge si `label_type_map` est défini sur 5.
label_string_image	string	Étiquette personnalisée pour les éléments d'image. Pris en charge si `label_type_image` est défini sur 5.
expand_macros	integer	Développer les macros dans les étiquettes de la configuration de la carte. Valeurs possibles : 0 - Non; 1 - Oui.
background		Élément racine pour l'image d'arrière-plan (le cas échéant). Pris en charge si `imagetype` est défini sur 2.
name	string	Nom de l'image d'arrière-plan.
iconmap		Élément racine pour le mappage des icônes (le cas échéant).
name	string	Nom du mappage des icônes.
urls		Élément racine pour les URL utilisées par les cartes ou par chaque élément de carte.
name	string	Nom du lien.
url	string	URL du lien.
elementtype	integer	Type d'élément de carte auquel le lien appartient. Valeurs possibles : 0 - Hôte; 1 - Carte; 2 - Déclencheur; 3 - Groupe d'hôtes; 4 - Image.
selements		Élément racine pour les selements de carte.

Element	Type	Description
shapes		Élément racine pour les formes de carte.
type	integer	Type de forme. Valeurs possibles : 0 - Rectangle; 1 - Ellipse.
x	integer	Coordonnée X de la forme en pixels.
y	integer	Coordonnée Y de la forme en pixels.
width	integer	Largeur de la forme.
height	integer	Hauteur de la forme.
text	string	Texte à l’intérieur de la forme.
font	integer	Style de police du texte. Valeurs possibles : 0 - Georgia, serif; 1 - ”Palatino Linotype”, ”Book Antiqua”, Palatino, serif; 2 - ”Times New Roman”, Times, serif; 3 - Arial, Helvetica, sans-serif; 4 - ”Arial Black”, Gadget, sans-serif; 5 - ”Comic Sans MS”, cursive, sans-serif; 6 - Impact, Charcoal, sans-serif; 7 - ”Lucida Sans Unicode”, ”Lucida Grande”, sans-serif; 8 - Tahoma, Geneva, sans-serif; 9 - ”Trebuchet MS”, Helvetica, sans-serif; 10 - Verdana, Geneva, sans-serif; 11 - ”Courier New”, Courier, monospace; 12 - ”Lucida Console”, Monaco, monospace.
font_size	integer	Taille de police en pixels.
font_color	string	Couleur de police représentée en code hexadécimal.
text_halign	integer	Alignement horizontal du texte. Valeurs possibles : 0 - Centre; 1 - Gauche; 2 - Droite.
text_valign	integer	Alignement vertical du texte. Valeurs possibles : 0 - Milieu; 1 - Haut; 2 - Bas.
border_type	integer	Type de bordure de la forme. Valeurs possibles : 0 - Aucune; 1 - Ligne épaisse; 2 - Ligne pointillée; 3 - Ligne en tirets.
border_width	integer	Largeur de la bordure en pixels.
border_color	string	Couleur de la bordure représentée en code hexadécimal.
background_color	string	Couleur d’arrière-plan (remplissage) représentée en code hexadécimal.
zindex	integer	Valeur pour l’ordre de toutes les formes et lignes (z-index).
lines		Élément racine pour les lignes de carte.
x1	integer	Coordonnée X du point 1 de la ligne en pixels.
y1	integer	Coordonnée Y du point 1 de la ligne en pixels.
x2	integer	Coordonnée X du point 2 de la ligne en pixels.
y2	integer	Coordonnée Y du point 2 de la ligne en pixels.
line_type	integer	Type de ligne. Valeurs possibles : 0 - Aucune; 1 - Ligne épaisse; 2 - Ligne pointillée; 3 - Ligne en tirets.
line_width	integer	Largeur de la ligne en pixels.
line_color	string	Couleur de la ligne représentée en code hexadécimal.
zindex	integer	Valeur pour l’ordre de toutes les formes et lignes (z-index).

Element	Type	Description
links		Élément racine pour les liens entre les éléments de carte.
drawtype	integer	Style du lien. Valeurs possibles : 0 - Ligne; 2 - Ligne épaisse; 3 - Ligne pointillée; 4 - Ligne en tirets.
color	string	Couleur du lien (6 caractères, hexadécimal).
label	string	Étiquette du lien.
selementid1	id	ID d'un élément à connecter.
selementid2	id	ID de l'autre élément à connecter.
linktriggers		Élément racine pour les indicateurs d'état des liens.

Note:
Voir aussi : Objet Map (se référer à la propriété correspondante portant le même nom).

Éléments de carte

Element	Type	Description
elementtype	integer	Type d'élément de carte. Valeurs possibles : 0 - Hôte; 1 - Carte; 2 - Déclencheur; 3 - Groupe d'hôtes; 4 - Image.
elements		Élément racine pour les entités Zabbix (hôte, groupe d'hôtes, carte, etc.) représentées sur la carte. Toutes les entités ont leur propre élément (`host`, etc. ; voir, par exemple, Format d'export).
label	string	Libellé de l'icône.
label_location	integer	Emplacement du libellé. Valeurs possibles : -1 - Utiliser la valeur par défaut de la carte; 0 - Bas; 1 - Gauche; 2 - Droite; 3 - Haut.
x	integer	Emplacement sur l'axe X.
y	integer	Emplacement sur l'axe Y.
elementsubtype	integer	Sous-type d'élément. Pris en charge si `elementtype` est défini sur 3. Valeurs possibles : 0 - Groupe d'hôtes unique; 1 - Tous les groupes d'hôtes.
areatype	integer	Taille de la zone. Pris en charge si `elementtype` est défini sur 1. Valeurs possibles : 0 - Identique à l'ensemble de la carte; 1 - Taille personnalisée.
width	integer	Largeur de la zone. Pris en charge si `areatype` est défini sur 1.
height	integer	Hauteur de la zone. Pris en charge si `areatype` est défini sur 1.
viewtype	integer	Algorithme de placement de la zone. Pris en charge si `elementsubtype` est défini sur 1. Valeurs possibles : 0 - Placer uniformément dans la zone.

Element	Type	Description
use_iconmap	integer	Utiliser le mappage d'icônes pour cet élément. Pertinent uniquement si le mappage d'icônes est activé au niveau de la carte. Valeurs possibles : 0 - Non; 1 - Oui.
selementid	id	ID d'enregistrement unique de l'élément.
icon_off		Élément racine pour l'image à utiliser lorsque l'élément est dans l'état `OK`.
name	string	Nom unique de l'image.
icon_on		Élément racine pour l'image à utiliser lorsque l'élément est dans l'état `Problem`.
name	string	Nom unique de l'image.
icon_disabled		Élément racine pour l'image à utiliser lorsque l'élément est désactivé.
name	string	Nom unique de l'image.
icon_maintenance		Élément racine pour l'image à utiliser lorsque l'élément est en maintenance.
name	string	Nom unique de l'image.
urls		Élément racine pour les URL utilisées par les cartes ou chaque élément de carte.
name	string	Nom du lien.
url	string	URL du lien.
evaltype	integer	Type d'évaluation pour les balises.
tags		Élément racine pour les balises de problème (pour les éléments hôte et groupe d'hôtes). Si des balises sont fournies, seuls les problèmes portant ces balises seront affichés sur la carte.
tag	string	Nom de la balise.
value	string	Valeur de la balise.
operator	integer	Opérateur.

Note:
Voir aussi : Objet d'élément de carte (reportez-vous à la propriété correspondante portant le même nom).

Indicateurs d'état des liens de carte

Element	Type	Description
drawtype	integer	Style du lien lorsque le déclencheur est dans l'état 'Problem'. Valeurs possibles : 0 - Ligne; 2 - Ligne en gras; 3 - Ligne pointillée; 4 - Ligne en tirets.
color	string	Couleur du lien (6 symboles, hexadécimal) lorsque le déclencheur est dans l'état 'Problem'.
trigger		Élément racine du déclencheur utilisé pour indiquer l'état du lien.
description	string	Nom du déclencheur.
expression	string	Expression du déclencheur.
recovery_expression	string	Expression de rétablissement du déclencheur.

Note:
Voir aussi : Objet de déclencheur de lien de carte (se référer à la propriété correspondante portant le même nom).

Types de média

Vue d'ensemble

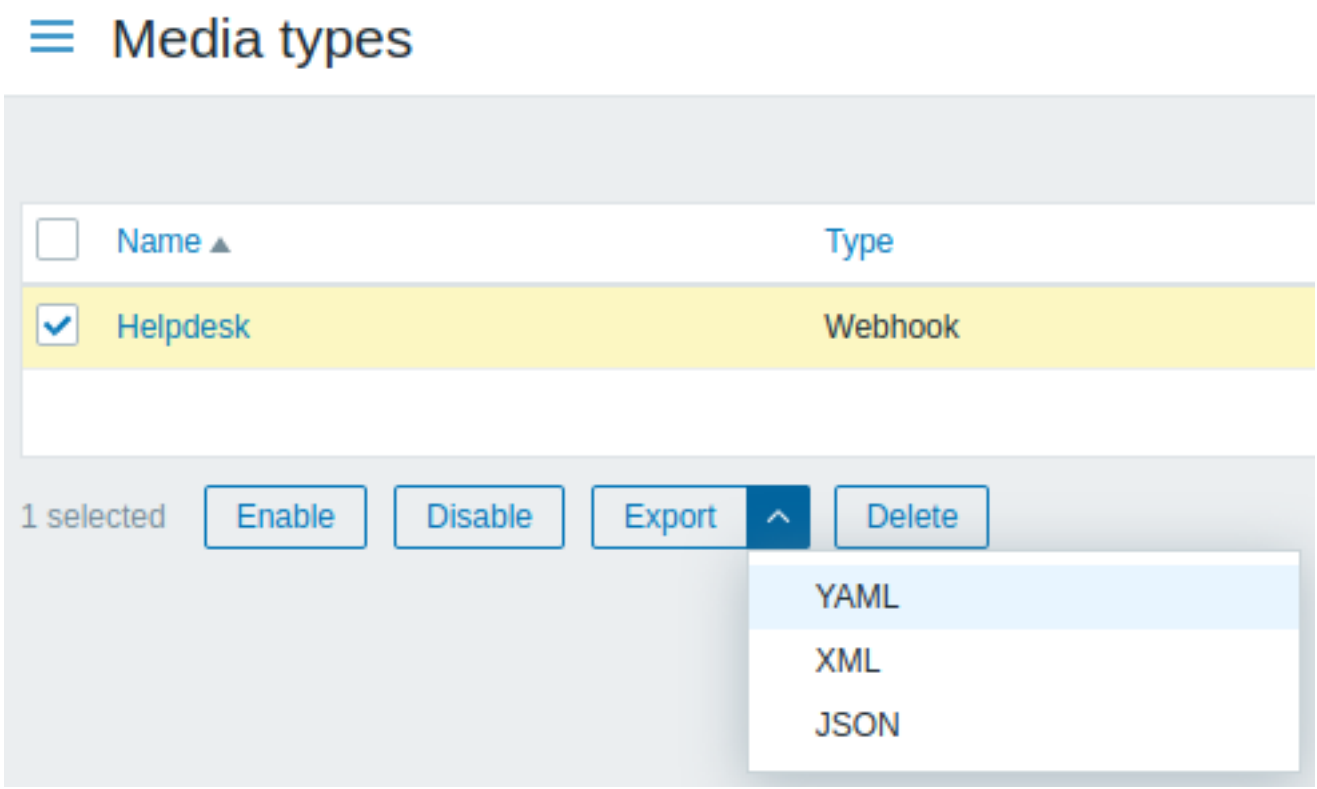
Les types de média sont exportés avec tous les objets associés et les relations entre objets.

Exportation

Pour exporter des types de média, procédez comme suit :

1. Accédez à *Alerts* > *Media types*.
2. Cochez les cases des types de média à exporter.

3. Cliquez sur *Export* sous la liste pour exporter au format YAML par défaut, ou cliquez sur la flèche pour sélectionner un autre format d'exportation.



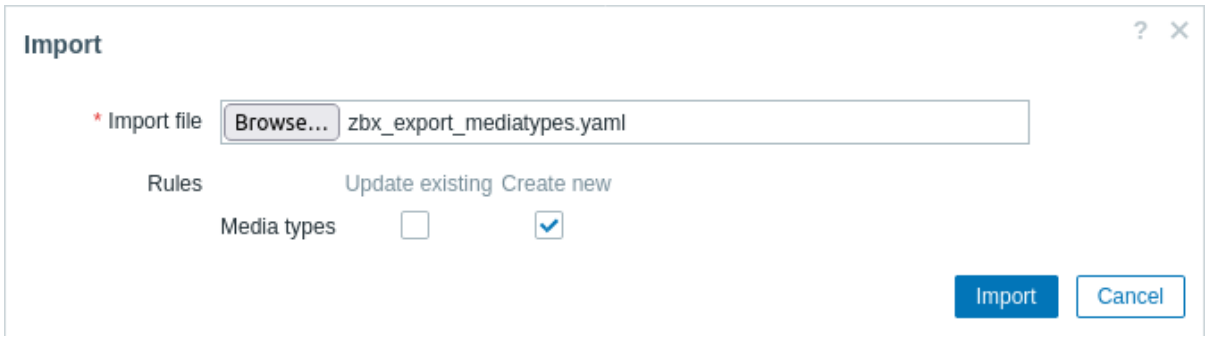
Selon le format sélectionné, les types de média sont exportés vers un fichier local avec un nom par défaut :

- `zabbix_export_mediatypes.yaml` - en export YAML ;
- `zabbix_export_mediatypes.xml` - en export XML ;
- `zabbix_export_mediatypes.json` - en export JSON.

Importation

Pour importer des types de média, procédez comme suit :

1. Accédez à *Alertes* > *Types de média*.
2. Cliquez sur *Importer* dans le coin supérieur droit.
3. Sélectionnez le fichier d'importation.
4. Cochez les cases correspondant aux règles d'importation que vous souhaitez appliquer (voir ci-dessous).
5. Cliquez sur *Importer* dans le coin inférieur droit du formulaire de configuration.



Règles d'importation :

Règle	Description
Mettre à jour les éléments existants	Cochez cette case pour mettre à jour les types de média existants à l'aide des données du fichier d'importation.
Créer de nouveaux éléments	Cochez cette case pour créer de nouveaux types de média à l'aide des données du fichier d'importation. Si le fichier d'importation contient un type de média qui existe déjà (par exemple, *E-mail*), un nouveau type ne sera pas créé et le type de média existant ne sera pas mis à jour.

Vous ne pouvez pas examiner les modifications apportées aux types de média pendant l'importation. Un message indiquant la réussite ou l'échec de l'importation s'affiche dans l'interface.

Format d'export

Export au format YAML :

```yaml
zabbix_export:
  version: '8.0'
  media_types:
    - name: Pushover
      type: WEBHOOK
      parameters:
        - name: endpoint
          value: 'https://api.pushover.net/1/messages.json'
        - name: eventid
          value: '{EVENT.ID}'
        - name: event_nseverity
          value: '{EVENT.NSEVERITY}'
        - name: event_source
          value: '{EVENT.SOURCE}'
        - name: event_value
          value: '{EVENT.VALUE}'
        - name: expire
          value: '1200'
        - name: message
          value: '{ALERT.MESSAGE}'
        - name: priority_average
          value: '0'
        - name: priority_default
          value: '0'
        - name: priority_disaster
          value: '0'
        - name: priority_high
          value: '0'
        - name: priority_information
          value: '0'
        - name: priority_not_classified
          value: '0'
        - name: priority_warning
          value: '0'
        - name: retry
          value: '60'
        - name: title
          value: '{ALERT.SUBJECT}'
        - name: token
          value: '<PUSHOVER TOKEN HERE>'
        - name: triggerid
          value: '{TRIGGER.ID}'
        - name: url
          value: '{$ZABBIX.URL}'
        - name: url_title
          value: Zabbix
        - name: user
          value: '{ALERT.SENDTO}'
      status: DISABLED
      max_sessions: '0'
      script: |
        try {
            var params = JSON.parse(value),
                request = new HttpRequest(),
                data,
                response,
                severities = [
```

```
            {name: 'not_classified', color: '#97AAB3'},
            {name: 'information', color: '#7499FF'},
            {name: 'warning', color: '#FFC859'},
            {name: 'average', color: '#FFA059'},
            {name: 'high', color: '#E97659'},
            {name: 'disaster', color: '#E45959'},
            {name: 'resolved', color: '#009900'},
            {name: 'default', color: '#000000'}
        ],
        priority;

    if (typeof params.HTTPProxy === 'string' && params.HTTPProxy.trim() !== '') {
        request.setProxy(params.HTTPProxy);
    }

    if ([0, 1, 2, 3].indexOf(parseInt(params.event_source)) === -1) {
        throw 'Paramètre "event_source" incorrect fourni: "' + params.event_source + '".\nDoit êtr
    }

    if (params.event_value !== '0' && params.event_value !== '1'
        && (params.event_source === '0' || params.event_source === '3')) {
        throw 'Paramètre "event_value" incorrect fourni: ' + params.event_value + '\nDoit être 0 o
    }

    if ([0, 1, 2, 3, 4, 5].indexOf(parseInt(params.event_nseverity)) === -1) {
        params.event_nseverity = '7';
    }

    if (params.event_value === '0') {
        params.event_nseverity = '6';
    }

    priority = params['priority_' + severities[params.event_nseverity].name] || params.priority_de

    if (isNaN(priority) || priority < -2 || priority > 2) {
        throw '"priority" doit être compris entre -2 et 2';
    }

    if (params.event_source === '0' && isNaN(params.triggerid)) {
        throw 'le champ "triggerid" n\'est pas un nombre';
    }

    if (isNaN(params.eventid)) {
        throw 'le champ "eventid" n\'est pas un nombre';
    }

    if (typeof params.message !== 'string' || params.message.trim() === '') {
        throw 'le champ "message" ne peut pas être vide';
    }

    data = {
        token: params.token,
        user: params.user,
        title: params.title,
        message: params.message,
        url: (params.event_source === '0')
            ? params.url + '/tr_events.php?triggerid=' + params.triggerid + '&eventid=' + params.e
            : params.url,
        url_title: params.url_title,
        priority: priority
    };
```

```
        if (priority == 2) {
            if (isNaN(params.retry) || params.retry < 30) {
                throw 'le champ "retry" doit être un nombre d\'une valeur d\'au moins 30 si "priority"
            }

            if (isNaN(params.expire) || params.expire > 10800) {
                throw 'le champ "expire" doit être un nombre d\'une valeur d\'au plus 10800 si "priori
            }

            data.retry = params.retry;
            data.expire = params.expire;
        }

        data = JSON.stringify(data);
        Zabbix.log(4, '[ Pushover Webhook ] Envoi de la requête : ' + params.endpoint + '\n' + data);

        request.addHeader('Content-Type: application/json');
        response = request.post(params.endpoint, data);

        Zabbix.log(4, '[ Pushover Webhook ] Réponse reçue avec le code de statut ' + request.getStatus

        if (response !== null) {
            try {
                response = JSON.parse(response);
            }
            catch (error) {
                Zabbix.log(4, '[ Pushover Webhook ] Échec de l\'analyse de la réponse reçue de Pushove
                response = null;
            }
        }

        if (request.getStatus() != 200 || response === null || typeof response !== 'object' || respons
            if (response !== null && typeof response === 'object' && typeof response.errors === 'objec
                    && typeof response.errors[0] === 'string') {
                throw response.errors[0];
            }
            else {
                throw 'Erreur inconnue. Consultez le journal de débogage pour plus d\'informations.';
            }
        }

        return 'OK';
    }
    catch (error) {
        Zabbix.log(4, '[ Pushover Webhook ] Échec de la notification Pushover : ' + error);
        throw 'Échec de la notification Pushover : ' + error;
    }
description: |
    Veuillez consulter le guide de configuration ici : https://git.zabbix.com/projects/ZBX/repos/zabbi

    Définissez le paramètre token avec la clé de votre application Pushover.
    Lors de l'attribution du média Pushover à l'utilisateur Zabbix, ajoutez la clé utilisateur dans le
message_templates:
    - event_source: TRIGGERS
      operation_mode: PROBLEM
      subject: 'Problème: {EVENT.NAME}'
      message: |
        Le problème a commencé à {EVENT.TIME} le {EVENT.DATE}
        Nom du problème: {EVENT.NAME}
        Hôte: {HOST.NAME}
        Gravité: {EVENT.SEVERITY}
        Données opérationnelles: {EVENT.OPDATA}
```

```
      ID du problème d'origine: {EVENT.ID}
      {TRIGGER.URL}
  - event_source: TRIGGERS
    operation_mode: RECOVERY
    subject: 'Résolu en {EVENT.DURATION}: {EVENT.NAME}'
    message: |
      Le problème a été résolu à {EVENT.RECOVERY.TIME} le {EVENT.RECOVERY.DATE}
      Nom du problème: {EVENT.NAME}
      Durée du problème: {EVENT.DURATION}
      Hôte: {HOST.NAME}
      Gravité: {EVENT.SEVERITY}
      ID du problème d'origine: {EVENT.ID}
      {TRIGGER.URL}
  - event_source: TRIGGERS
    operation_mode: UPDATE
    subject: 'Problème mis à jour en {EVENT.AGE}: {EVENT.NAME}'
    message: |
      {USER.FULLNAME} a {EVENT.UPDATE.ACTION} le problème le {EVENT.UPDATE.DATE} à {EVENT.UPDATE.TIM
      {EVENT.UPDATE.MESSAGE}

      L'état actuel du problème est {EVENT.STATUS}, son ancienneté est de {EVENT.AGE}, accusé de réc
  - event_source: DISCOVERY
    operation_mode: PROBLEM
    subject: 'Découverte: {DISCOVERY.DEVICE.STATUS} {DISCOVERY.DEVICE.IPADDRESS}'
    message: |
      Règle de découverte: {DISCOVERY.RULE.NAME}

      IP de l'appareil: {DISCOVERY.DEVICE.IPADDRESS}
      DNS de l'appareil: {DISCOVERY.DEVICE.DNS}
      État de l'appareil: {DISCOVERY.DEVICE.STATUS}
      Temps de fonctionnement de l'appareil: {DISCOVERY.DEVICE.UPTIME}

      Nom du service de l'appareil: {DISCOVERY.SERVICE.NAME}
      Port du service de l'appareil: {DISCOVERY.SERVICE.PORT}
      État du service de l'appareil: {DISCOVERY.SERVICE.STATUS}
      Temps de fonctionnement du service de l'appareil: {DISCOVERY.SERVICE.UPTIME}
  - event_source: AUTOREGISTRATION
    operation_mode: PROBLEM
    subject: 'Auto-enregistrement: {HOST.HOST}'
    message: |
      Nom de l'hôte: {HOST.HOST}
      IP de l'hôte: {HOST.IP}
      Port de l'agent: {HOST.PORT}
```

Éléments exportés

Les éléments exportés sont expliqués dans le tableau ci-dessous.

Element	Type	Description
name	string	(obligatoire) Nom du type de média.
type	string	(obligatoire) Transport utilisé par le type de média. Valeurs possibles:[1] EMAIL (0), SMS (1), SCRIPT (2), WEBHOOK (4).
status	string	Indique si le type de média est activé. Valeurs possibles:[1] ENABLED (0, par défaut), DISABLED (1)
max_sessions	integer	Nombre maximal d'alertes pouvant être traitées en parallèle. Valeurs possibles pour SMS:[1] 1 (par défaut). Valeurs possibles pour les autres types de média:[1] 0-100 (où 0 - illimité).
attempts	integer	Nombre maximal de tentatives d'envoi d'une alerte. Valeurs possibles:[1] 1-10 (par défaut: 3).
attempt_interval	string	Intervalle (en utilisant des secondes ou un suffixe de temps) entre les tentatives de réessai. Valeurs possibles:[1] 0-60s (par défaut: 10s).
description	string	Description du type de média.

Element	Type	Description
message_templates		Élément racine pour les modèles de message du type de média.
event_source	string	(obligatoire) Source de l'événement. Valeurs possibles:[1] TRIGGERS (0), DISCOVERY (1), AUTOREGISTRATION (2), INTERNAL (3), SERVICE (4).
operation_mode	string	Mode de fonctionnement. Valeurs possibles:[1] PROBLEM (0), RECOVERY (1), UPDATE (2).
subject	string	Objet du message.
message	string	Corps du message.

Note:
Voir aussi: Objet Media type (reportez-vous à la propriété correspondante portant le même nom).

Email

Les éléments supplémentaires suivants sont exportés uniquement pour le type de média *Email*.

Element	Type	Description
provider	string	Fournisseur de messagerie.
smtp_server	string	Serveur SMTP.
smtp_port	integer	Port du serveur SMTP auquel se connecter. Par défaut: 25.
smtp_helo	string	HELO SMTP.
smtp_email	string	Adresse e-mail à partir de laquelle les notifications seront envoyées.
smtp_security	string	Niveau de sécurité de la connexion SMTP à utiliser. Valeurs possibles:[1] NONE (0, par défaut), STARTTLS (1), SSL_OR_TLS (2).
smtp_verify_host	string	Vérification SSL de l'hôte pour SMTP. Valeurs possibles:[1] NO (0, par défaut), YES (1).
smtp_verify_peer	string	Vérification SSL du pair pour SMTP. Valeurs possibles:[1] NO (0, par défaut), YES (1).
smtp_authentication	string	Méthode d'authentification SMTP à utiliser. Valeurs possibles:[1] NONE (0, par défaut), PASSWORD (1), OAUTH (2).
username	string	Nom d'utilisateur.
password	string	Mot de passe d'authentification.
redirection_url	string	URL de l'interface Zabbix vers laquelle rediriger l'autorisation OAuth.
client_id	string	Identifiant du client enregistré auprès du serveur d'autorisation OAuth.
authorization_url	string	URL OAuth, avec paramètres, pour obtenir des jetons d'accès et d'actualisation.
token_url	string	URL OAuth pour échanger le jeton d'autorisation contre des jetons d'accès et d'actualisation.
message_format	string	Format du message. Valeurs possibles:[1] TEXT (0), HTML (1, par défaut).

Note:
Voir aussi: Objet Media type (reportez-vous à la propriété correspondante portant le même nom).

SMS

Les éléments supplémentaires suivants sont exportés uniquement pour le type de média *SMS*.

Element	Type	Description
gsm_modem	string	(obligatoire) Nom du périphérique série du modem GSM.

Note:
Voir aussi : Media type object (reportez-vous à la propriété correspondante portant le même nom).

Script

Les éléments supplémentaires suivants sont exportés uniquement pour le type de média *Script*.

Element	Type	Description
script name	string	(required) Nom du script.
parameters		Élément racine pour les paramètres du script.
sortorder	string	(required) Ordre des paramètres du script transmis au script en tant qu'arguments de ligne de commande.
value	string	Valeur du paramètre du script.

Note:
Voir aussi: Objet type de média (se référer à la propriété correspondante portant le même nom).

Webhook

Les éléments supplémentaires suivants sont exportés uniquement pour le type de média *Webhook*.

Element	Type	Description
script	string	Script.
timeout	string	Intervalle de délai d'attente de la requête HTTP du script JavaScript. Valeurs possibles :[1] 1-60s (par défaut : 30s).
process_tags	string	Indique s'il faut traiter les tags renvoyés. Valeurs possibles :[1] NO (0, par défaut), YES (1).
show_event_menu	string	Indique la présence d'une entrée dans le menu d'événement si la macro {EVENT.TAGS.*} a été résolue avec succès dans les champs `event_menu_url` et `event_menu_name`. Valeurs possibles :[1] NO (0, par défaut), YES (1).
event_menu_url	string	URL de l'entrée du menu d'événement. Prend en charge la macro {EVENT.TAGS.*}.
event_menu_name	string	Nom de l'entrée du menu d'événement. Prend en charge la macro {EVENT.TAGS.*}.
parameters		Élément racine pour les paramètres du type de média webhook.
name	string	(obligatoire) Nom du paramètre webhook.
value	string	Valeur du paramètre webhook.

Note:
Voir aussi : Media type object (reportez-vous à la propriété correspondante portant le même nom).

Notes de bas de page

[1] Les valeurs entières de l'API entre parenthèses, par exemple, ENABLED (0), sont mentionnées uniquement à titre de référence. Pour plus d'informations, consultez la page de l'objet API liée dans l'entrée du tableau ou à la fin de chaque section.

Tableaux de bord

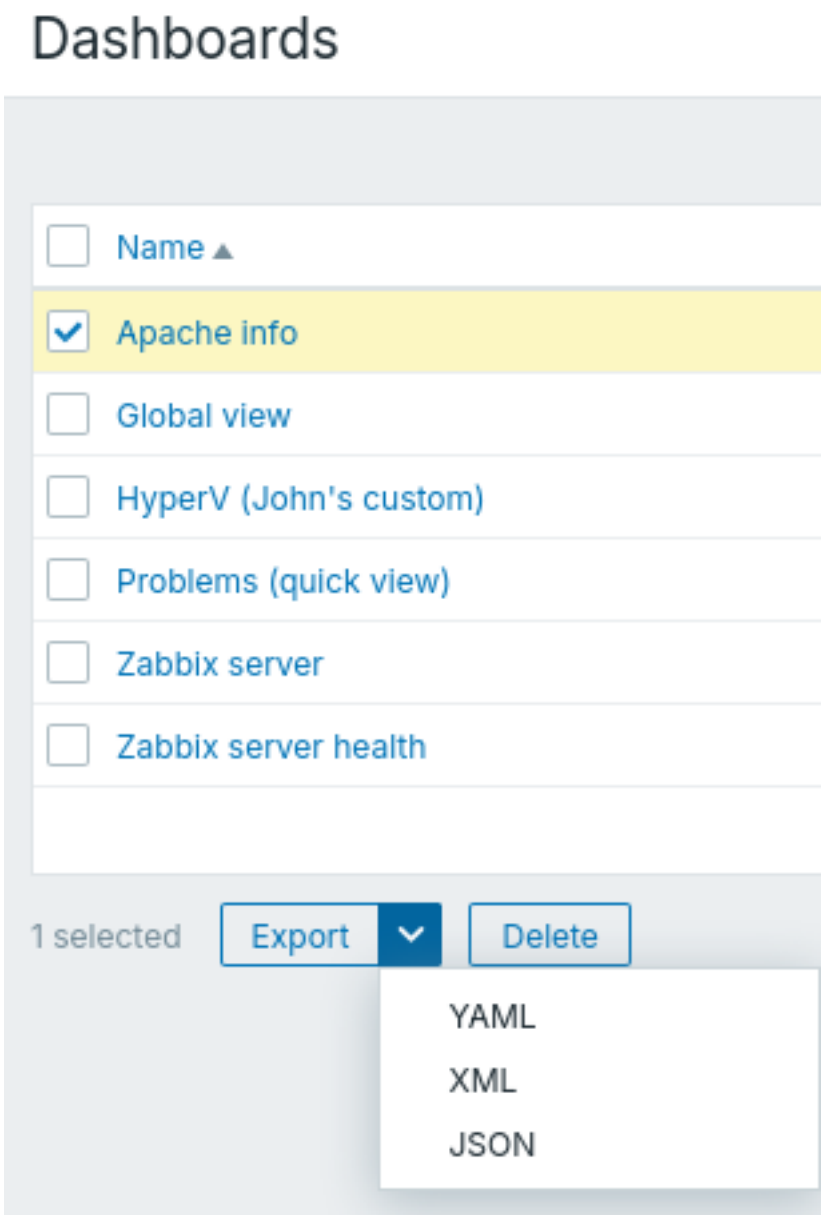
Vue d'ensemble

Les tableaux de bord sont exportés avec tous les objets associés et les relations entre objets.

Exportation

Pour exporter des tableaux de bord, procédez comme suit :

1. Accédez à *Tableaux de bord* > *Tous les tableaux de bord*.
2. Cochez les cases des tableaux de bord à exporter.
3. Cliquez sur *Exporter* sous la liste pour exporter au format YAML par défaut, ou cliquez sur la flèche pour sélectionner un autre format d'exportation.



Selon le format sélectionné, les tableaux de bord sont exportés vers un fichier local avec un nom par défaut :

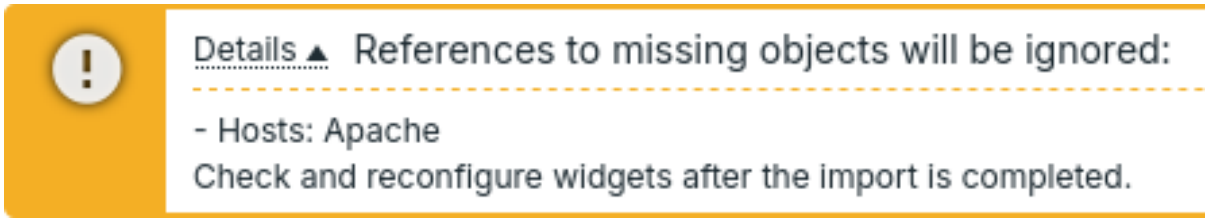
- `zbx_export_dashboards.yaml` - en export YAML ;
- `zbx_export_dashboards.xml` - en export XML ;
- `zbx_export_dashboards.json` - en export JSON.

Importation

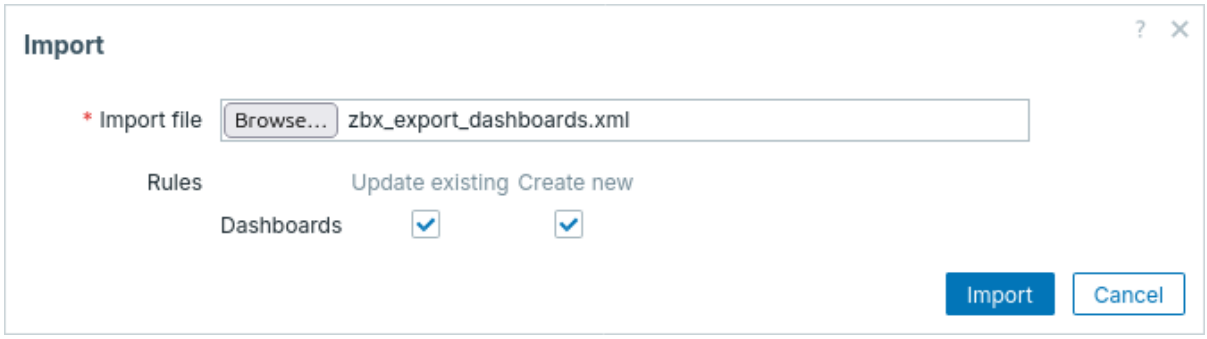
Pour importer des tableaux de bord, procédez comme suit :

1. Accédez à *Tableaux de bord* > *Tous les tableaux de bord*.
2. Cliquez sur *Importer* dans le coin supérieur droit. Cette option est désactivée si l'utilisateur ne dispose pas de l'autorisation *Créer et modifier les tableaux de bord*.
3. Sélectionnez le fichier d'importation.
4. Cochez les cases correspondant aux règles d'importation que vous souhaitez appliquer (voir ci-dessous).
5. Cliquez sur *Importer* dans le coin inférieur droit du formulaire de configuration.

Si le fichier d'importation fait référence à des objets qui n'existent pas sur le système cible, un message d'alerte s'affiche. Cela est attendu pour les widgets qui font référence à des hôtes, éléments ou autres objets spécifiques. Dans ce cas, Zabbix autorise l'importation du tableau de bord, supprime les références manquantes et marque le widget comme n'étant pas entièrement configuré lorsque les données sont obligatoires, ou supprime les critères de filtrage lorsque les champs sont utilisés uniquement pour le filtrage.



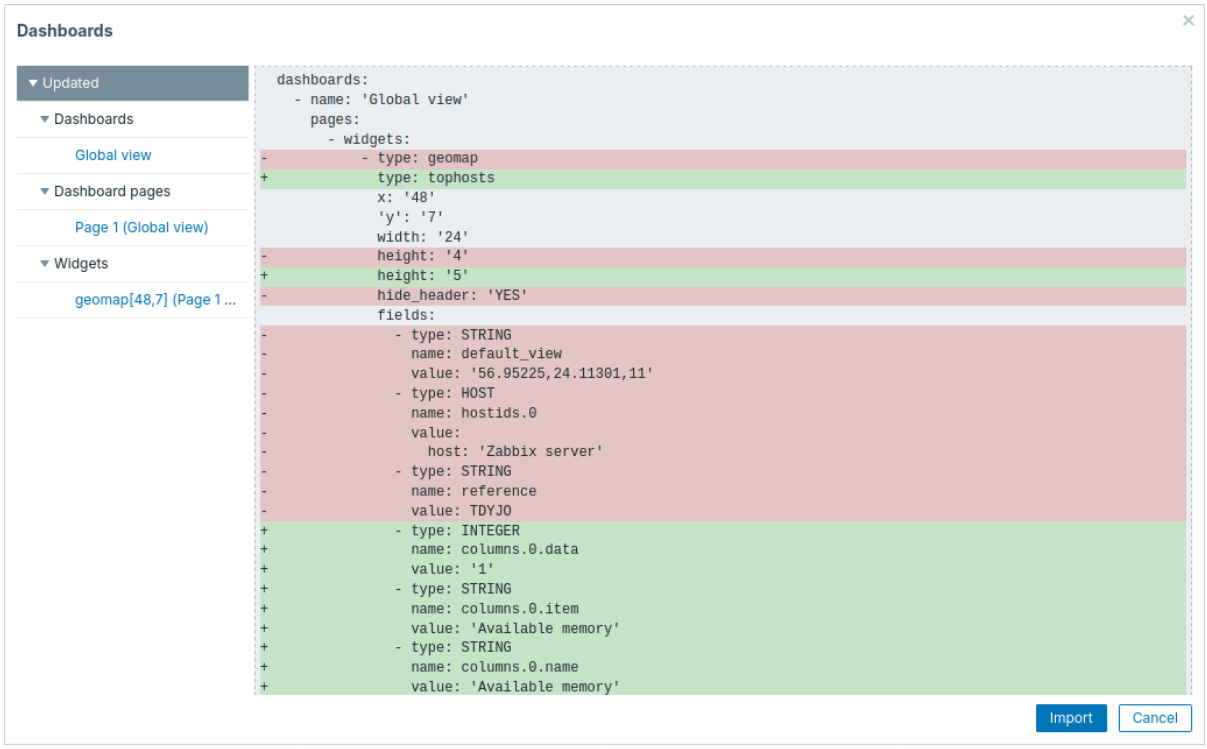
Cliquer à nouveau sur le bouton *Importer* ajoutera et/ou mettra à jour le tableau de bord.



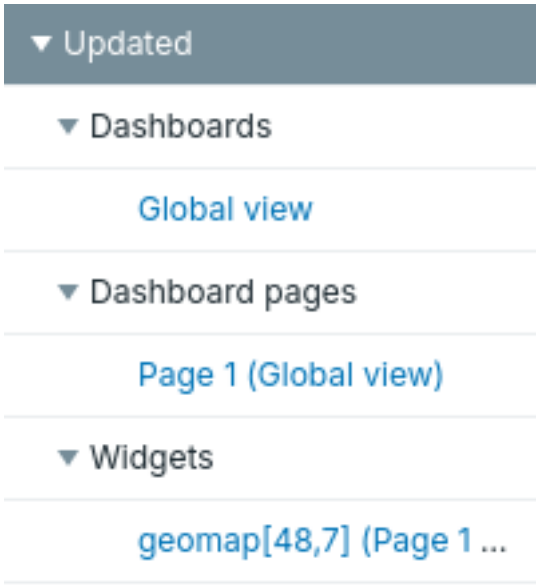
Règles d'importation :

Règle	Description
Mettre à jour les éléments existants	Les éléments existants seront mis à jour à l'aide des données du fichier d'importation. Sinon, ils ne seront pas mis à jour.
Créer de nouveaux éléments	Les nouveaux éléments seront créés à l'aide des données du fichier d'importation. Sinon, ils ne seront pas créés.

Sur l'écran suivant, vous pourrez consulter le contenu des tableaux de bord importés. S'il s'agit d'un nouveau tableau de bord, tous les éléments seront affichés en vert. Lors de la mise à jour d'un tableau de bord existant, les nouveaux éléments du tableau de bord seront mis en évidence en vert ; les éléments supprimés seront mis en évidence en rouge ; les éléments qui n'ont pas changé seront affichés sur un arrière-plan gris.



Le menu situé à gauche permet de parcourir la liste des modifications. La section *Mis à jour* met en évidence toutes les modifications apportées aux éléments existants du tableau de bord. La section *Ajoutés* répertorie les nouveaux éléments du tableau de bord. Les éléments de chaque section sont regroupés par type d'élément ; cliquez sur la flèche grise pour développer ou réduire le groupe d'éléments.



Examinez les modifications apportées au tableau de bord, puis cliquez sur *Importer* pour effectuer l'importation. Un message indiquant la réussite ou l'échec de l'importation s'affichera dans l'interface.

Format d'export

Export vers YAML :

```yaml
zabbix_export:
  version: '8.0'
  dashboards:
    - name: 'Global view'
      pages:
        - widgets:
            - type: tophosts
              name: 'Top hosts by CPU utilization'
              width: '25'
              height: '4'
              fields:
                - type: INTEGER
                  name: column
                  value: '1'
                - type: INTEGER
                  name: columns.0.data
                  value: '2'
                - type: STRING
                  name: columns.0.item
                  value: 'System name'
                - type: STRING
                  name: columns.0.name
                  value: 'Host name'
                - type: STRING
                  name: columns.1.base_color
                  value: 4CAF50
                - type: INTEGER
                  name: columns.1.data
                  value: '1'
                - type: INTEGER
                  name: columns.1.display
                  value: '3'
                - type: STRING
                  name: columns.1.item
                  value: 'CPU utilization'
```

```
            - type: STRING
              name: columns.1.max
              value: '100'
            - type: STRING
              name: columns.1.min
              value: '0'
            - type: STRING
              name: columns.1.name
              value: Utilization
            - type: INTEGER
              name: columns.2.data
              value: '1'
            - type: STRING
              name: columns.2.item
              value: 'Load average (1m avg)'
            - type: STRING
              name: columns.2.name
              value: '1m avg'
            - type: INTEGER
              name: columns.3.data
              value: '1'
            - type: STRING
              name: columns.3.item
              value: 'Load average (5m avg)'
            - type: STRING
              name: columns.3.name
              value: '5m avg'
            - type: INTEGER
              name: columns.4.data
              value: '1'
            - type: STRING
              name: columns.4.item
              value: 'Load average (15m avg)'
            - type: STRING
              name: columns.4.name
              value: '15m avg'
            - type: INTEGER
              name: columns.5.data
              value: '1'
            - type: INTEGER
              name: columns.5.decimal_places
              value: '0'
            - type: STRING
              name: columns.5.item
              value: 'Number of processes'
            - type: STRING
              name: columns.5.name
              value: Processes
            - type: STRING
              name: columnsthresholds.1.color.0
              value: FFFF00
            - type: STRING
              name: columnsthresholds.1.color.1
              value: FF8000
            - type: STRING
              name: columnsthresholds.1.color.2
              value: FF4000
            - type: STRING
              name: columnsthresholds.1.threshold.0
              value: '50'
            - type: STRING
              name: columnsthresholds.1.threshold.1
```

```
            value: '80'
          - type: STRING
            name: columnsthresholds.1.threshold.2
            value: '90'
          - type: STRING
            name: reference
            value: NLVMO
        - type: hostavail
          'y': '4'
          width: '25'
          fields:
            - type: INTEGER
              name: interface_type.0
              value: '1'
        - type: problems
          name: 'Current problems'
          'y': '6'
          width: '48'
          height: '5'
          fields:
            - type: STRING
              name: reference
              value: BRUQE
            - type: INTEGER
              name: show
              value: '3'
            - type: INTEGER
              name: show_tags
              value: '3'
        - type: systeminfo
          x: '25'
          width: '23'
          height: '4'
        - type: problemsbysv
          name: 'Problems by severity'
          x: '25'
          'y': '4'
          width: '23'
          fields:
            - type: STRING
              name: reference
              value: LBSVB
            - type: INTEGER
              name: show_type
              value: '1'
        - type: gauge
          name: 'Zabbix server'
          x: '48'
          width: '12'
          height: '4'
          hide_header: 'YES'
          fields:
            - type: INTEGER
              name: angle
              value: '270'
            - type: INTEGER
              name: desc_size
              value: '8'
            - type: ITEM
              name: itemid.0
              value:
                host: 'Zabbix server'
```

```
      key: vm.memory.utilization
  - type: STRING
    name: max
    value: '100'
  - type: STRING
    name: min
    value: '0'
  - type: INTEGER
    name: scale_size
    value: '11'
  - type: INTEGER
    name: show.0
    value: '1'
  - type: INTEGER
    name: show.1
    value: '2'
  - type: INTEGER
    name: show.2
    value: '3'
  - type: INTEGER
    name: show.3
    value: '4'
  - type: INTEGER
    name: show.4
    value: '5'
  - type: STRING
    name: thresholds.0.color
    value: 42A5F5
  - type: STRING
    name: thresholds.0.threshold
    value: '0'
  - type: STRING
    name: thresholds.1.color
    value: 9CCC65
  - type: STRING
    name: thresholds.1.threshold
    value: '20'
  - type: STRING
    name: thresholds.2.color
    value: FFEE58
  - type: STRING
    name: thresholds.2.threshold
    value: '40'
  - type: STRING
    name: thresholds.3.color
    value: FFA726
  - type: STRING
    name: thresholds.3.threshold
    value: '60'
  - type: STRING
    name: thresholds.4.color
    value: FF7043
  - type: STRING
    name: thresholds.4.threshold
    value: '80'
  - type: INTEGER
    name: th_arc_size
    value: '9'
  - type: INTEGER
    name: th_show_arc
    value: '1'
  - type: INTEGER
```

```
            name: th_show_labels
            value: '1'
          - type: INTEGER
            name: units_bold
            value: '1'
          - type: INTEGER
            name: units_size
            value: '24'
          - type: INTEGER
            name: value_arc_size
            value: '30'
          - type: INTEGER
            name: value_bold
            value: '1'
          - type: INTEGER
            name: value_size
            value: '24'
      - type: svggraph
        x: '48'
        'y': '4'
        width: '24'
        height: '3'
        hide_header: 'YES'
        fields:
          - type: INTEGER
            name: axisx
            value: '0'
          - type: STRING
            name: ds.0.color
            value: AB47BC
          - type: INTEGER
            name: ds.0.fill
            value: '0'
          - type: STRING
            name: ds.0.hosts.0
            value: 'Zabbix server'
          - type: STRING
            name: ds.0.items.0
            value: 'Available memory'
          - type: INTEGER
            name: ds.0.transparency
            value: '2'
          - type: INTEGER
            name: ds.0.type
            value: '2'
          - type: INTEGER
            name: ds.0.width
            value: '2'
          - type: INTEGER
            name: ds.1.axisy
            value: '1'
          - type: STRING
            name: ds.1.color
            value: 42A5F5
          - type: INTEGER
            name: ds.1.fill
            value: '0'
          - type: STRING
            name: ds.1.hosts.0
            value: 'Zabbix server'
          - type: STRING
            name: ds.1.items.0
```

```
            value: 'CPU utilization'
          - type: INTEGER
            name: ds.1.transparency
            value: '0'
          - type: INTEGER
            name: ds.1.type
            value: '2'
          - type: INTEGER
            name: ds.1.width
            value: '2'
          - type: STRING
            name: reference
            value: TOYGS
        - type: geomap
          x: '48'
          'y': '7'
          width: '24'
          height: '4'
          hide_header: 'YES'
          fields:
            - type: STRING
              name: default_view
              value: '56.95225,24.11301,11'
            - type: HOST
              name: hostids.0
              value:
                host: 'Zabbix server'
            - type: STRING
              name: reference
              value: TDYJO
        - type: clock
          name: 'Heure locale'
          x: '60'
          width: '12'
          hide_header: 'YES'
          fields:
            - type: INTEGER
              name: clock_type
              value: '1'
            - type: INTEGER
              name: show.0
              value: '1'
            - type: INTEGER
              name: show.1
              value: '2'
            - type: INTEGER
              name: show.2
              value: '3'
            - type: INTEGER
              name: time_sec
              value: '0'
            - type: INTEGER
              name: tzone_format
              value: '1'
        - type: item
          name: 'Performance du serveur'
          x: '60'
          'y': '2'
          width: '12'
          hide_header: 'YES'
          fields:
            - type: INTEGER
```

```
      name: aggregate_function
      value: '2'
    - type: INTEGER
      name: decimal_size
      value: '32'
    - type: STRING
      name: description
      value: 'Values per second'
    - type: INTEGER
      name: desc_h_pos
      value: '0'
    - type: INTEGER
      name: desc_v_pos
      value: '0'
    - type: ITEM
      name: itemid.0
      value:
        host: 'Zabbix server'
        key: 'zabbix[wcache,values]'
    - type: INTEGER
      name: show.0
      value: '1'
    - type: INTEGER
      name: show.1
      value: '2'
    - type: INTEGER
      name: show.2
      value: '5'
    - type: INTEGER
      name: sparkline.fill
      value: '2'
    - type: INTEGER
      name: sparkline.width
      value: '2'
    - type: STRING
      name: thresholds.0.color
      value: DCEDC8
    - type: STRING
      name: thresholds.0.threshold
      value: '20'
    - type: STRING
      name: thresholds.1.color
      value: FFF9C4
    - type: STRING
      name: thresholds.1.threshold
      value: '40'
    - type: STRING
      name: thresholds.2.color
      value: FFCCBC
    - type: STRING
      name: thresholds.2.threshold
      value: '60'
    - type: INTEGER
      name: units_bold
      value: '0'
    - type: INTEGER
      name: units_show
      value: '0'
    - type: INTEGER
      name: value_h_pos
      value: '0'
    - type: INTEGER
```

```
                name: value_size
                value: '32'
```

Éléments exportés

Les éléments exportés sont expliqués dans le tableau ci-dessous.

Element	Type	Description
name	string	(obligatoire) Nom du tableau de bord.
pages		Élément racine pour les pages du tableau de bord
widgets		Élément racine pour les pages du tableau de bord

Note:
Voir aussi: Objet Dashboard (reportez-vous à la propriété correspondante portant le même nom).

Widget du tableau de bord

Element	Type	Description
type	string	(obligatoire) Type du widget du tableau de bord. Valeurs possibles : actionlog - Journal des actions; clock - Horloge; discovery - État de la découverte; favgraphs - Graphiques favoris; favmaps - Cartes favorites; gauge - Jauge; geomap - Carte géographique; graph - Graphique (classique); graphprototype - Prototype de graphique; honeycomb - Honeycomb; hostavail - Disponibilité de l'hôte; hostcard - Carte de l'hôte; hostnavigator - Navigateur d'hôtes; itemcard - Carte de l'élément; itemhistory - Historique de l'élément; itemnavigator - Navigateur d'éléments; item - Valeur de l'élément; map - Carte; navtree - Arborescence de navigation de la carte; piechart - Diagramme circulaire; problemhosts - Hôtes en problème; problems - Problèmes; problemsbysv - Problèmes par gravité; scatterplot - Nuage de points; slareport - Rapport SLA; svggraph - Graphique; systeminfo - Informations système; tophosts - Principaux hôtes; topitems - Principaux éléments; toptriggers - Principaux déclencheurs; trigover - Vue d'ensemble des déclencheurs; url - URL; web - Supervision web.
x	integer	Une position horizontale à partir du côté gauche du tableau de bord. Valeurs possibles :[1] 0-71.
y	integer	Une position verticale à partir du haut du tableau de bord. Valeurs possibles :[1] 0-63.
name	string	Nom personnalisé du widget.
width	integer	La largeur du widget. Valeurs possibles :[1] 1-72.

Element	Type	Description
height	integer	La hauteur du widget. Valeurs possibles :[1] 1-64.
hide_header	string	Masquer l'en-tête du widget. Valeurs possibles :[1] NO (0, par défaut), YES (1).
fields		Élément racine pour les champs du widget du tableau de bord.

Note:
Voir aussi : Widget du tableau de bord (reportez-vous à la propriété correspondante portant le même nom).

Champ de widget du tableau de bord

Element		Type	Description
type		string	(obligatoire) Type de champ du widget. Valeurs possibles :[1] INTEGER (0), STRING (1), HOST_GROUP (2), HOST (3), ITEM (4), ITEM_PROTOTYPE (5), GRAPH (6), GRAPH_PROTOTYPE (7), MAP (8), SERVICE (9), SLA (10), USER (11), ACTION (12), MEDIA_TYPE (13).
name		string	(obligatoire) Nom du champ du widget.
value		mixed	(obligatoire) Valeur du champ du widget, selon le type de champ.
	host	string	Hôte de l'élément.
	key	string	Clé de l'élément.

Note:
Voir aussi : Champ de widget du tableau de bord (se référer à la propriété correspondante portant le même nom).

Notes de bas de page

[1] Les valeurs entières de l'API entre parenthèses, par exemple ENABLED (0), sont mentionnées uniquement à titre de référence. Pour plus d'informations, consultez la page de l'objet API liée dans l'entrée du tableau ou à la fin de chaque section.

Expressions régulières

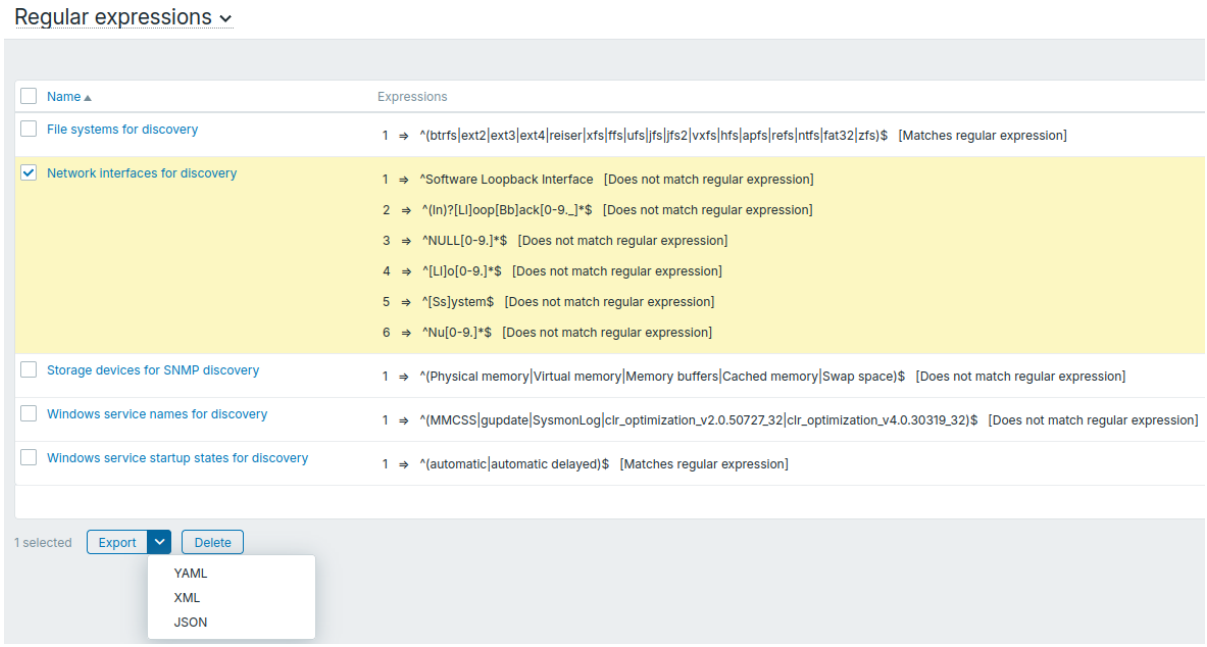
Vue d'ensemble

Les expressions régulières sont exportées avec tous les objets associés et les relations entre objets.

Exportation

Pour exporter des expressions régulières, procédez comme suit :

1. Accédez à *Administration* > *Général* > *Expressions régulières*.
2. Cochez les cases des expressions régulières à exporter.
3. Cliquez sur *Exporter* sous la liste pour exporter au format YAML par défaut, ou cliquez sur la flèche pour sélectionner un autre format d'exportation.



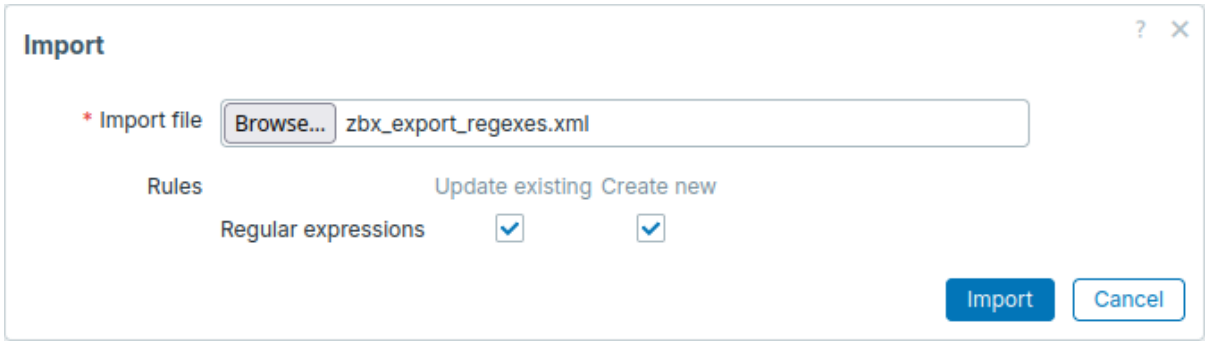
Selon le format sélectionné, les expressions régulières sont exportées vers un fichier local avec un nom par défaut :

- `zbx_export_regexes.yaml` - en export YAML ;
- `zbx_export_regexes.xml` - en export XML ;
- `zbx_export_regexes.json` - en export JSON.

Importation

Pour importer des expressions régulières, procédez comme suit :

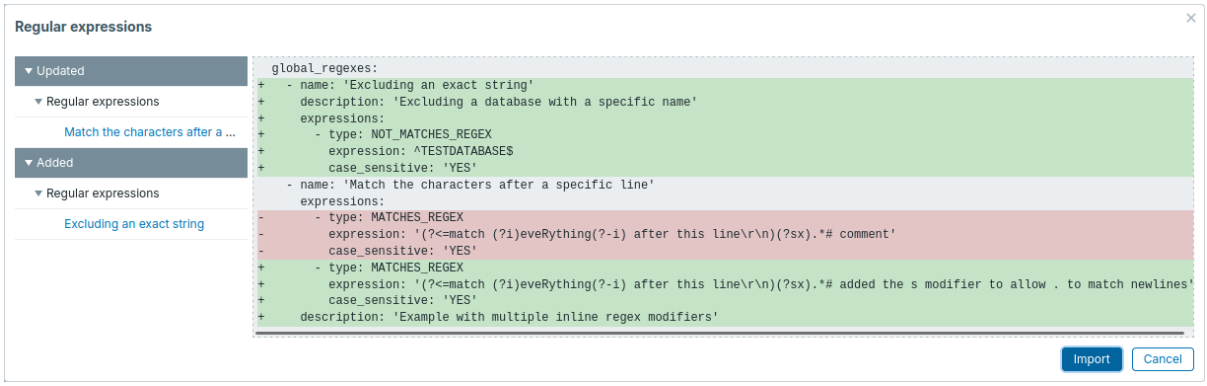
1. Accédez à *Administration* > *Général* > *Expressions régulières*.
2. Cliquez sur *Importer* dans le coin supérieur droit.
3. Sélectionnez le fichier d'importation.
4. Cochez les cases correspondant aux règles d'importation que vous souhaitez appliquer (voir ci-dessous).
5. Cliquez sur *Importer* dans le coin inférieur droit du formulaire de configuration.



Règles d'importation :

Règle	Description
Mettre à jour les éléments existants	Les expressions régulières existantes seront mises à jour à l'aide des données du fichier d'importation, ce qui remplacera entièrement les expressions existantes portant le même nom. Sinon, elles ne seront pas mises à jour.
Créer de nouveaux éléments	De nouvelles expressions régulières seront créées à l'aide des données du fichier d'importation. Sinon, elles ne seront pas créées.

Sur l'écran suivant, vous pourrez consulter le contenu des expressions régulières importées. S'il s'agit d'une nouvelle expression régulière, toutes les données seront affichées en vert. Lors de la mise à jour d'une expression régulière existante, les nouvelles données de l'expression régulière seront mises en évidence en vert ; les données supprimées seront mises en évidence en rouge ; les données inchangées seront affichées sur un arrière-plan gris.



Le menu situé à gauche permet de parcourir la liste des modifications. La section *Mises à jour* met en évidence toutes les modifications apportées aux expressions régulières existantes. La section *Ajouts* répertorie les nouvelles expressions régulières. Cliquez sur la flèche pour développer ou réduire un groupe d'expressions régulières.

Examinez les modifications apportées aux expressions régulières, puis cliquez sur *Importer* pour effectuer l'importation des expressions régulières. Un message indiquant la réussite ou l'échec de l'importation s'affichera dans l'interface.

Format d'export

Exporter au format YAML :

```
zabbix_export:
  version: '8.0'
  global_regexes:
    - name: 'File systems for discovery'
      description: 'Expression régulière globale utilisée pour découvrir et identifier les systèmes de fic
      expressions:
        - type: MATCHES_REGEX
          expression: ^(btrfs|ext2|ext3|ext4|reiser|xfs|ffs|ufs|jfs|jfs2|vxfs|hfs|apfs|refs|ntfs|fat32|zfs
    - name: 'Storage devices for SNMP discovery'
      expressions:
        - type: NOT_MATCHES_REGEX
          expression: '^(Physical memory|Virtual memory|Memory buffers|Cached memory|Swap space)$'
          case_sensitive: 'YES'
```

Éléments exportés

Éléments exportés des expressions régulières :

Element	Type	Description
global_regexes	object	Élément racine pour les expressions régulières.
name	string	Nom de l'expression régulière (obligatoire).
description	text	Description de l'expression régulière.
expressions	object	Élément racine pour les expressions régulières.
type	string	Type de l'expression régulière.
expression	string	Expression régulière.
exp_delimiter	string	Délimiteur de l'expression régulière ; présent lorsque le type est *Contient n'importe quelle sous-chaîne de la liste*.
case_sensitive	string	Option de sensibilité à la casse pour l'expression régulière.

Note:
Voir aussi : Objet expression régulière (reportez-vous à la propriété correspondante portant le même nom).

Expressions régulières

Vue d'ensemble Perl Compatible Regular Expressions (PCRE, PCRE2) sont prises en charge dans Zabbix.

Il existe deux façons d'utiliser les expressions régulières dans Zabbix :

- Saisir manuellement une expression régulière
- Utiliser une expression régulière globale créée dans Zabbix

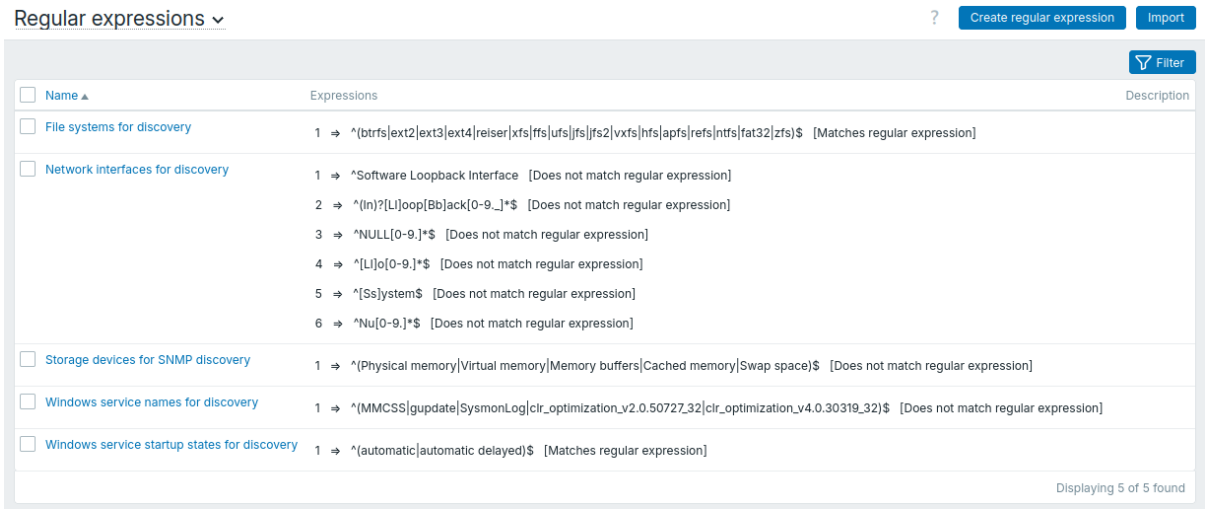
Expressions régulières Vous pouvez saisir manuellement une expression régulière dans les emplacements pris en charge.
 Notez que l'expression ne doit pas commencer par @, car ce symbole est utilisé dans Zabbix pour faire référence aux expressions régulières globales.

> **Attention:**
> Il est possible d'épuiser la mémoire de la pile lors de l'utilisation d'expressions régulières. Consultez PCRE PERFORMANCE pour plus d'informations.

Notez qu'en correspondance multilignes, les ancres ^ et $ correspondent respectivement au début et à la fin de chaque ligne, au lieu du début et de la fin de l'ensemble de la chaîne.

Consultez également des exemples de bon échappement dans divers contextes.

Expressions régulières globales Zabbix fournit un éditeur avancé dans l'interface pour créer et tester des expressions régulières complexes.
 Pour y accéder, allez dans *Administration* > *Général* > *Expressions régulières*.
 Zabbix inclut un certain nombre d'expressions régulières globales prédéfinies dans son jeu de données par défaut.

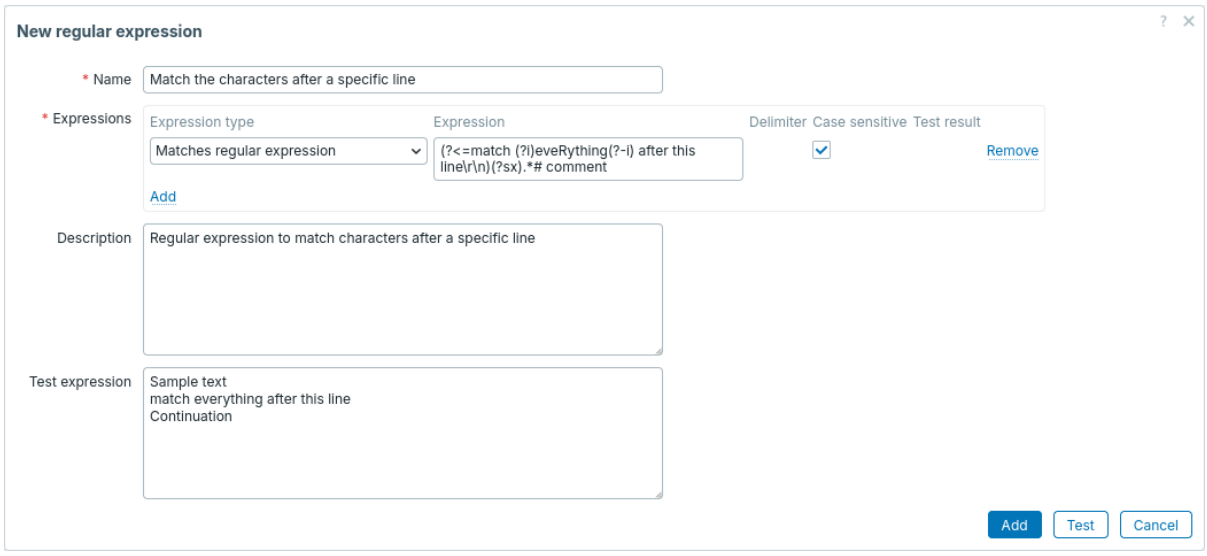


Vous pouvez filtrer la liste des expressions régulières par nom et par description afin de trouver les expressions dont vous avez besoin.

Configuration

Pour créer une expression régulière globale :

1. Cliquez sur *Create regular expression*.
2. Saisissez les paramètres de l'expression régulière dans le formulaire.



Tous les champs obligatoires sont marqués d'un astérisque rouge.

Parameter	Description
Name	Définissez le nom de l'expression régulière. Tous les caractères Unicode sont autorisés.
Expressions	Cliquez sur *Add* dans la section *Expressions* pour ajouter une nouvelle sous-expression.
Expression type	Sélectionnez le type d'expression : **Contains string** - recherche la sous-chaîne. **Contains any substring from list** - recherche toute sous-chaîne d'une liste délimitée. La liste délimitée inclut une virgule `,`, un point `.` ou une barre oblique `/`. **Does not contain string** - recherche toute chaîne à l'exception de la sous-chaîne. **Matches regular expression** - recherche l'expression régulière. **Does not match regular expression** - ne recherche pas l'expression régulière.
Expression	Saisissez la sous-chaîne ou l'expression régulière. Longueur maximale : 2048 caractères.
Delimiter	Une virgule `,`, un point `.` ou une barre oblique `/` pour séparer les chaînes de texte dans une expression régulière. Ce paramètre est actif uniquement lorsque le type d'expression *Contains any substring from list* est sélectionné.
Case sensitive	Case à cocher permettant de spécifier si une expression régulière est sensible à la casse.
Description	Saisissez une description de l'expression régulière.
Test expression	Valeur à utiliser pour tester l'expression régulière.

Une barre oblique / dans l'expression est traitée littéralement et non comme un délimiteur. Cela permet d'enregistrer sans erreur des expressions contenant une barre oblique.

Attention:
Un nom d'expression régulière personnalisée dans Zabbix peut contenir des virgules, des espaces, etc. Lorsque cela peut entraîner une mauvaise interprétation lors de la référence (par exemple, une virgule dans le paramètre d'une clé d'élément), la référence entière peut être placée entre guillemets, comme ceci : *"@My custom regexp for purpose1, purpose2"*.
 Les noms d'expressions régulières ne doivent pas être placés entre guillemets dans les autres emplacements (par exemple, dans les propriétés des règles LLD).

Une fois qu'une expression régulière a été créée, elle peut être utilisée à plusieurs endroits dans l'interface en faisant référence à son nom précédé de `@`, par exemple `@mycustomregexp`.

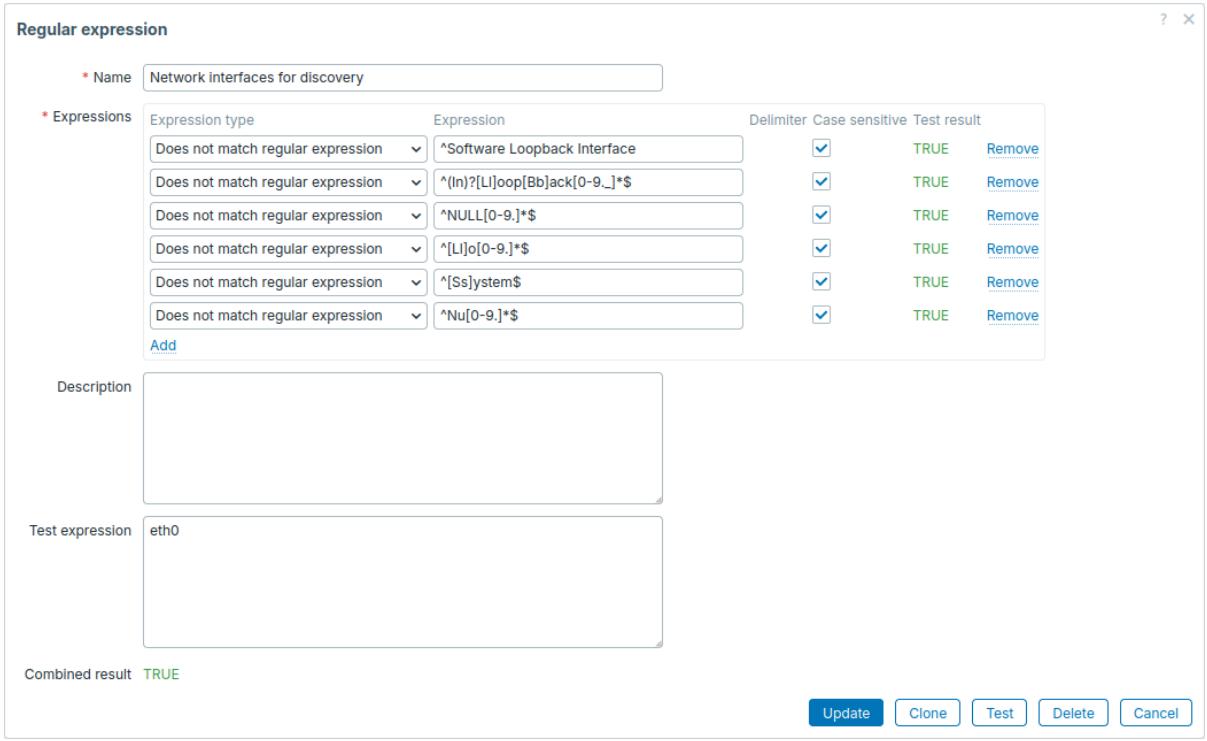
Boutons du formulaire

Les boutons situés en bas du formulaire vous permettent d'effectuer diverses opérations :

Button	Description
Add	Ajouter une expression régulière. Ce bouton n'est disponible que pour les nouvelles expressions régulières.
Update	Mettre à jour les propriétés d'une expression régulière.
Clone	Créer une autre expression régulière basée sur les propriétés de l'expression régulière actuelle.
Test	Tester une expression régulière et ses sous-expressions par rapport à une valeur fournie.
Delete	Supprimer l'expression régulière.
Cancel	Annuler la modification des propriétés de l'expression régulière.

Test d'une expression régulière

Vous pouvez tester l'expression régulière et ses sous-expressions en saisissant une valeur dans le champ *Test expression* et en cliquant sur le bouton *Test*.



Dans la colonne *Test result*, l'état de chaque sous-expression est affiché. Le champ *Combined result* affiche l'état global de l'expression régulière. Si plusieurs sous-expressions sont définies, Zabbix utilise l'opérateur logique AND pour calculer *Combined result* ; c'est-à-dire que si au moins un résultat est `False`, *Combined result* est également `False`.

Expressions régulières globales par défaut

L'ensemble de données par défaut inclut les expressions régulières globales prédéfinies suivantes :

Name	Expression	Matches
File systems for discovery	`^(btrfs|ext2|ext3|ext4|reiser|xfs|ffs|ufs|jfs|jfs2|vxfs|hfs|apfs|refs|ntfs|fat32|zfs)$`	"btrfs" ou "ext2" ou "ext3" ou "ext4" ou "reiser" ou "xfs" ou "ffs" ou "ufs" ou "jfs" ou "jfs2" ou "vxfs" ou "hfs" ou "apfs" ou "refs" ou "ntfs" ou "fat32" ou "zfs".
Network interfaces for discovery	`^Software Loopback Interface`	Chaînes commençant par "Software Loopback Interface".
	`^lo$`	"lo".
	`^(In)?[Ll]oop[Bb]ack[0-9._]*$`	Chaînes commençant éventuellement par "In", suivies de "L" ou "l", puis de "oop", puis de "B" ou "b", puis de "ack", et éventuellement suivies d'un nombre quelconque de chiffres, de points ou de traits de soulignement.
	`^NULL[0-9.]*$`	Chaînes commençant par "NULL" et éventuellement suivies d'un nombre quelconque de chiffres ou de points.
	`^[Ll]o[0-9.]*$`	Chaînes commençant par "Lo" ou "lo" et éventuellement suivies d'un nombre quelconque de chiffres ou de points.
	`^[Ss]ystem$`	"System" ou "system".
	`^Nu[0-9.]*$`	Chaînes commençant par "Nu" et éventuellement suivies d'un nombre quelconque de chiffres ou de points.
Storage devices for SNMP discovery	`^(Physical memory|Virtual memory|Memory buffers|Cached memory|Swap space)$`	"Physical memory" ou "Virtual memory" ou "Memory buffers" ou "Cached memory" ou "Swap space".

Name	Expression	Matches
Windows service names for discovery	`^(MMCSS|gupdate|SysmonLog|clr_optimization_v2.0.50727_32|clr_optimization_v4.0.30319_32)$`	"MMCSS" ou "gupdate" ou "SysmonLog" ou chaînes correspondant à "clr_optimization_v2.0.50727_32" ou "clr_optimization_v4.0.30319_32", où chaque point peut être remplacé par n'importe quel caractère sauf un saut de ligne.
Windows service startup states for discovery	`^(automatic|automatic delayed)$`	"automatic" ou "automatic delayed".

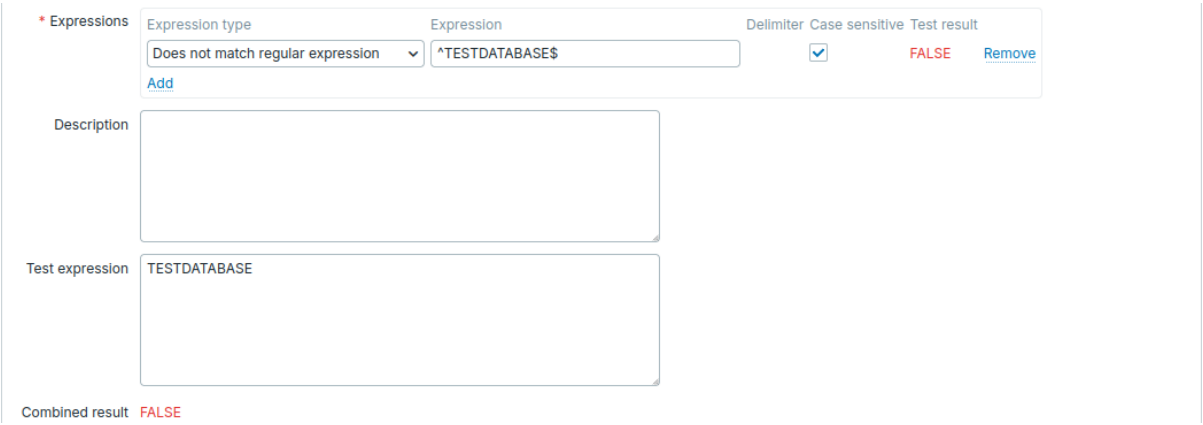
Importation/exportation d'une expression régulière

Vous pouvez importer ou exporter des expressions régulières globales comme décrit sur la page de configuration d'export/import des expressions régulières.

Exemples Exemple d'exclusion d'une chaîne exacte

Utilisez l'expression suivante dans la découverte de bas niveau pour découvrir des bases de données, à l'exception d'une base de données portant un nom spécifique :

```
^TESTDATABASE$
```

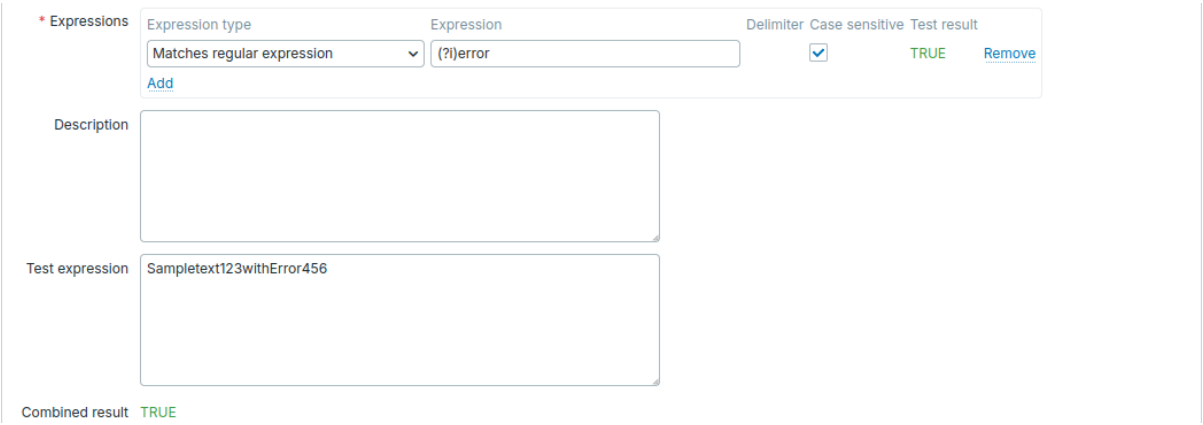


Les champs *Test result* et *Combined result* affichent tous deux `FALSE` : le nom correspond à "TESTDATABASE", donc la condition *Does not match regular expression* n'est pas satisfaite.

Exemple avec un modificateur regex en ligne pour une correspondance insensible à la casse

Utilisez l'expression régulière suivante, qui inclut le modificateur en ligne `(?i)` pour effectuer une correspondance insensible à la casse des caractères "error" :

```
(?i)error
```



Les champs *Test result* et *Combined result* affichent tous deux `TRUE` : les caractères "error" sont correspondus même avec *Case sensitive* sélectionné, car le modificateur en ligne `(?i)` rend la correspondance insensible à la casse.

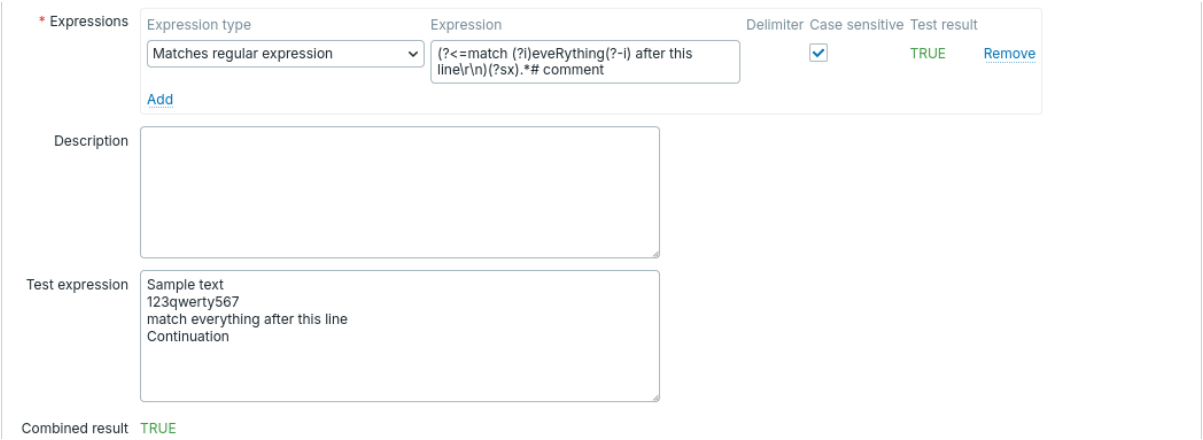
Exemple avec plusieurs modificateurs regex en ligne

Utilisez l'expression régulière suivante, qui inclut plusieurs modificateurs en ligne, pour faire correspondre les caractères après une ligne spécifique :

```
(?<=match (?i)eveRything(?-i) after this line\r\n)(?sx).*# comment
```

où :

- `(?i)` - active la correspondance insensible à la casse ;
- `(?-i)` - désactive la correspondance insensible à la casse ;
- `(?s)` - active le mode dotall, permettant à `.` de correspondre aux caractères de nouvelle ligne ;
- `(?x)` - active le mode étendu, permettant d'ignorer les espaces et les commentaires dans l'expression régulière.



Les champs *Test result* et *Combined result* affichent tous deux TRUE : les caractères suivant la ligne spécifique sont correspondus.

Attention:
Le modificateur g ne peut pas être spécifié en ligne. La liste des modificateurs disponibles se trouve dans la page de manuel pcresyntax. Pour plus d'informations sur la syntaxe PCRE, veuillez consulter la documentation HTML PCRE.

Prise en charge des expressions régulières par emplacement

Emplacement		Expression régulière	Expression régulière globale	Correspondance multi-lignes	Commentaires
Éléments agent					
	eventlog[]	Oui	Oui	Oui	Paramètres `regexp`, `severity`, `source`, `eventid`
	eventlog.count[]				Paramètres `regexp`, `severity`, `source`, `eventid`
	log[]				Paramètre `regexp`
	log.count[]				
	logrt[]		Oui/Non		Le paramètre `regexp` prend en charge les deux, le paramètre `file_regexp` prend en charge uniquement les expressions non globales
	logrt.count[]				
	proc.cpu.util[]		Non	Non	Paramètre `cmdline`
	proc.get[]				
	proc.mem[]				
	proc.num[]				
	sensor[]				Paramètres `device` et `sensor` sous Linux 2.4
	system.hw.macaddr[]				Paramètre `interface`
	system.sw.packages[]				Paramètre `regexp`
	system.sw.packages.get[]				Paramètre `regexp`
	vfs.dir.count[]				Paramètres `regex_incl`, `regex_excl`, `regex_excl_dir`

Emplacement		Expression régulière	Expression régulière globale	Correspondance multi-lignes	Commentaires
	vfs.dir.get[]				Paramètres `regex_incl`, `regex_excl`, `regex_excl_dir`
	vfs.dir.size[]				Paramètres `regex_incl`, `regex_excl`, `regex_excl_dir`
	vfs.file.regexp[] vfs.file.regmatch[] web.page.regexp[]			Oui	Paramètre `regexp`
Trappes SNMP					
	snmptrap[]	Oui	Oui	Non	Paramètre `regexp`
Prétraitement de la valeur de l'élément		Oui	Non	Non	Champ `pattern`
Fonctions pour les dé-clencheurs/éléments calculés					
	count()	Oui	Oui	Oui	Paramètre `pattern` si le paramètre `operator` est *regexp* ou *iregexp*
	countunique()	Oui	Oui		
	find()	Oui	Oui		
	logeventid()	Oui	Oui	Non	Paramètre `pattern`
	logsource()				
Découverte de bas niveau					
	Filtres	Oui	Oui	Non	Champ *Expression régulière*
	Remplacements	Oui	Non		Dans les options *correspond à*, *ne correspond pas à* pour les conditions *Opération*
Conditions d'action		Oui	Non	Non	Dans les options *correspond à*, *ne correspond pas à* pour les conditions d'autodécouverte *Nom d'hôte* et *Métadonnées d'hôte*
Scripts		Oui	Oui	Non	Champ *Règle de validation de l'entrée*
Supervision web		Oui	Non	Oui	*Variables* avec un préfixe **regex:** Champ *Chaîne requise*
Contexte de macro utilisa-teur		Oui	Non	Non	Dans le contexte de macro avec un préfixe **regex:**
Fonctions de macro					
	regsub() iregsub()	Oui	Non	Non	Paramètre `pattern`
Indicateurs de lien dans les cartes		Oui	Non	Non	Champ *Pattern* (pour les éléments textuels)
Mappage d'icônes		Oui	Oui	Non	Champ *Expression*

Emplacement		Expression régulière	Expression régulière globale	Correspondance multi-lignes	Commentaires
Mappage des valeurs Paramètres de configu-ration de l'agent		Oui	Non	Non	Champ *Valeur* si le type de mappage est `regexp`
	AllowKeyRegexp DenyKeyRegexp	Oui	Non	Non	Motif

Événements et résolution

Vue d'ensemble Il existe plusieurs types d'événements générés dans Zabbix :

- événements de déclencheur - chaque fois qu'un déclencheur change d'état (*OK→PROBLÈME→OK*)
- événements de service - chaque fois qu'un service change d'état (*OK→PROBLÈME→OK*)
- événements de découverte - lorsque des hôtes ou des services sont détectés
- événements d'autoréenregistrement - lorsque des agents actifs sont auto-enregistrés par le serveur
- événements internes - lorsqu'un élément/règle de découverte de bas niveau devient non pris en charge ou qu'un déclencheur passe dans un état inconnu

Les événements sont horodatés et peuvent servir de base à des actions telles que l'envoi d'e-mails de notification, la création de tickets ou l'exécution de commandes à distance.

Pour afficher les détails d'un événement dans l'interface, accédez à *Monitoring* > *Problems*. Vous pouvez y cliquer sur la date et l'heure de l'événement pour ouvrir la page des détails de l'événement, où vous pouvez voir des informations supplémentaires sur l'événement, notamment s'il a été supprimé en raison d'une maintenance ou supprimé manuellement par un utilisateur.

Résolution des problèmes Les événements de problème créés par les déclencheurs sont résolus automatiquement si la condition à l'origine du problème n'existe plus.

Pour ajouter de la flexibilité et de la simplicité à la gestion des événements de problème, les options suivantes existent :

- Corrélation des événements
- Accusé de réception et suppression des événements
- Fermeture manuelle des problèmes
- Périodes de maintenance - définir des périodes pendant lesquelles les problèmes sont supprimés
- Étiquetage - étiqueter les problèmes selon une caractéristique commune

Événements de déclencheur Le changement d'état d'un déclencheur est la source d'événements la plus fréquente et la plus importante. Chaque fois que le déclencheur change d'état, un événement est généré. L'événement contient des détails sur le changement d'état du déclencheur - quand il s'est produit et quel est le nouvel état.

Deux types d'événements sont créés par les déclencheurs - `Problem` et `OK`.

Événements de problème

Un événement de problème est créé :

- lorsqu'une expression de déclencheur est évaluée à TRUE si le déclencheur est dans l'état OK ;
- chaque fois qu'une expression de déclencheur est évaluée à TRUE si la génération de plusieurs événements de problème est activée pour le déclencheur.

Événements OK

Un événement OK clôt l(s) événement(s) de problème associé(s) et peut être créé par 3 composants :

- déclencheurs - selon les paramètres « OK event generation » et « OK event closes » ;
- corrélation d'événements
- gestionnaire de tâches - lorsqu'un événement est clos manuellement

Les déclencheurs disposent d'un paramètre « OK event generation » qui contrôle la manière dont les événements OK sont générés :

- *Expression* - un événement OK est généré pour un déclencheur en état de problème lorsque son expression est évaluée à FALSE. Il s'agit du paramètre le plus simple, activé par défaut.
- *Recovery expression* - un événement OK est généré pour un déclencheur en état de problème lorsque son expression est évaluée à FALSE et que l'expression de récupération est évaluée à TRUE. Cela peut être utilisé si le critère de récupération du déclencheur est différent du critère de problème.
- *None* - aucun événement OK n'est jamais généré. Cela peut être utilisé conjointement avec la génération de plusieurs événements de problème afin d'envoyer simplement une notification lorsqu'un événement se produit.

En outre, les déclencheurs disposent d'un paramètre « OK event closes » qui contrôle quels événements de problème sont fermés :

- *All problems* - un événement OK fermera tous les problèmes ouverts créés par le déclencheur
- *All problems if tag values match* - un événement OK fermera les problèmes ouverts créés par le déclencheur et ayant au moins une valeur de tag correspondante. Le tag est défini par le paramètre de déclencheur « Tag for matching ». S'il n'y a aucun événement de problème à fermer, alors l'événement OK n'est pas généré. Cela est souvent appelé corrélation d'événements au niveau du déclencheur.

Corrélation des événements

La corrélation des événements (également appelée corrélation globale des événements) permet de configurer des règles personnalisées de clôture des événements (entraînant la génération d'un événement OK).

Ces règles définissent comment les nouveaux événements de problème sont associés aux événements de problème existants, afin de permettre la clôture du nouvel événement ou des événements correspondants par la génération des événements OK correspondants.

Cependant, la corrélation des événements doit être configurée avec beaucoup de précaution, car elle peut affecter négativement les performances du traitement des événements ou, si elle est mal configurée, clôturer davantage d'événements que prévu (dans le pire des cas, tous les événements de problème pourraient être clôturés). Voici quelques conseils de configuration :

1. réduisez toujours la portée de la corrélation en définissant une étiquette unique pour l'événement de contrôle (l'événement associé aux anciens événements) et utilisez la condition de corrélation « étiquette du nouvel événement »
2. n'oubliez pas d'ajouter une condition basée sur l'ancien événement lorsque vous utilisez l'opération « clôturer l'ancien événement », sinon tous les problèmes existants pourraient être clôturés
3. évitez d'utiliser des noms d'étiquettes communs employés par différentes configurations de corrélation

Gestionnaire de tâches

Si le paramètre « Allow manual close » est activé pour le déclencheur, il est alors possible de fermer manuellement les événements de problème générés par le déclencheur. Cela se fait dans l'interface lors de la mise à jour d'un problème. L'événement n'est pas fermé directement - à la place, une tâche « close event » est créée, puis prise en charge par le gestionnaire de tâches peu après. Le gestionnaire de tâches générera un événement OK correspondant et l'événement de problème sera fermé.

Événements de service Les événements de service sont générés uniquement si les actions de service pour ces événements sont activées. Dans ce cas, chaque changement d'état du service crée un nouvel événement :

- Événement de problème - lorsque l'état du service passe de `OK` à `Problem`
- Événement OK - lorsque l'état du service passe de `Problem` à `OK`

L'événement contient des détails sur le changement d'état du service - quand il s'est produit et quel est le nouvel état.

Événements de découverte Zabbix analyse périodiquement les plages d'adresses IP définies dans les règles de découverte réseau. La fréquence de la vérification est configurable individuellement pour chaque règle. Une fois qu'un hôte ou un service est découvert, un événement de découverte (ou plusieurs événements) est généré.

Zabbix génère les événements suivants :

Event	When generated
Service Up	Chaque fois que Zabbix détecte un service actif.
Service Down	Chaque fois que Zabbix ne peut pas détecter le service.
Host Up	Si au moins un des services est UP pour l'adresse IP.
Host Down	Si tous les services ne répondent pas.
Service Discovered	Si le service est de nouveau disponible après une indisponibilité ou découvert pour la première fois.
Service Lost	Si le service est perdu après avoir été UP.
Host Discovered	Si l'hôte est de nouveau disponible après une indisponibilité ou découvert pour la première fois.

Event	When generated
Host Lost	Si l'hôte est perdu après avoir été UP.

Événements d'autoréenregistrement de l'agent actif L'autoréenregistrement de l'agent actif crée des événements dans Zabbix.

Si elle est configurée, un événement d'autoréenregistrement de l'agent actif est créé lorsqu'un agent actif auparavant inconnu demande des vérifications ou si les métadonnées de l'hôte ont changé. Le serveur ajoute un nouvel hôte auto-enregistré, en utilisant l'adresse IP et le port de l'agent reçus.

Pour plus d'informations, consultez la page d'autoréenregistrement de l'agent actif.

Événements internes Les événements internes se produisent lorsque :

- un élément change d'état de 'normal' à 'unsupported'
- un élément change d'état de 'unsupported' à 'normal'
- une règle de découverte à bas niveau change d'état de 'normal' à 'unsupported'
- une règle de découverte à bas niveau change d'état de 'unsupported' à 'normal'
- un déclencheur change d'état de 'normal' à 'unknown'
- un déclencheur change d'état de 'unknown' à 'normal'

L'objectif de l'introduction des événements internes est de permettre aux utilisateurs d'être avertis lorsqu'un événement interne se produit, par exemple lorsqu'un élément devient non pris en charge et cesse de collecter des données.

Les événements internes ne sont créés que lorsque les actions internes pour ces événements sont activées. Pour arrêter la génération d'événements internes (par exemple, pour des éléments devenant non pris en charge), désactivez toutes les actions pour les événements internes dans *Alerts* > *Actions* > *Internal actions*.

> **Note:**
> Si les actions internes sont désactivées, lorsqu'un objet est dans l'état 'unsupported', un événement de rétablissement pour cet objet sera tout de même créé.
> Si les actions internes sont activées, lorsqu'un objet est dans l'état 'unsupported', un événement de rétablissement pour cet objet sera créé, même si aucun 'problem event' n'a été créé pour l'objet.

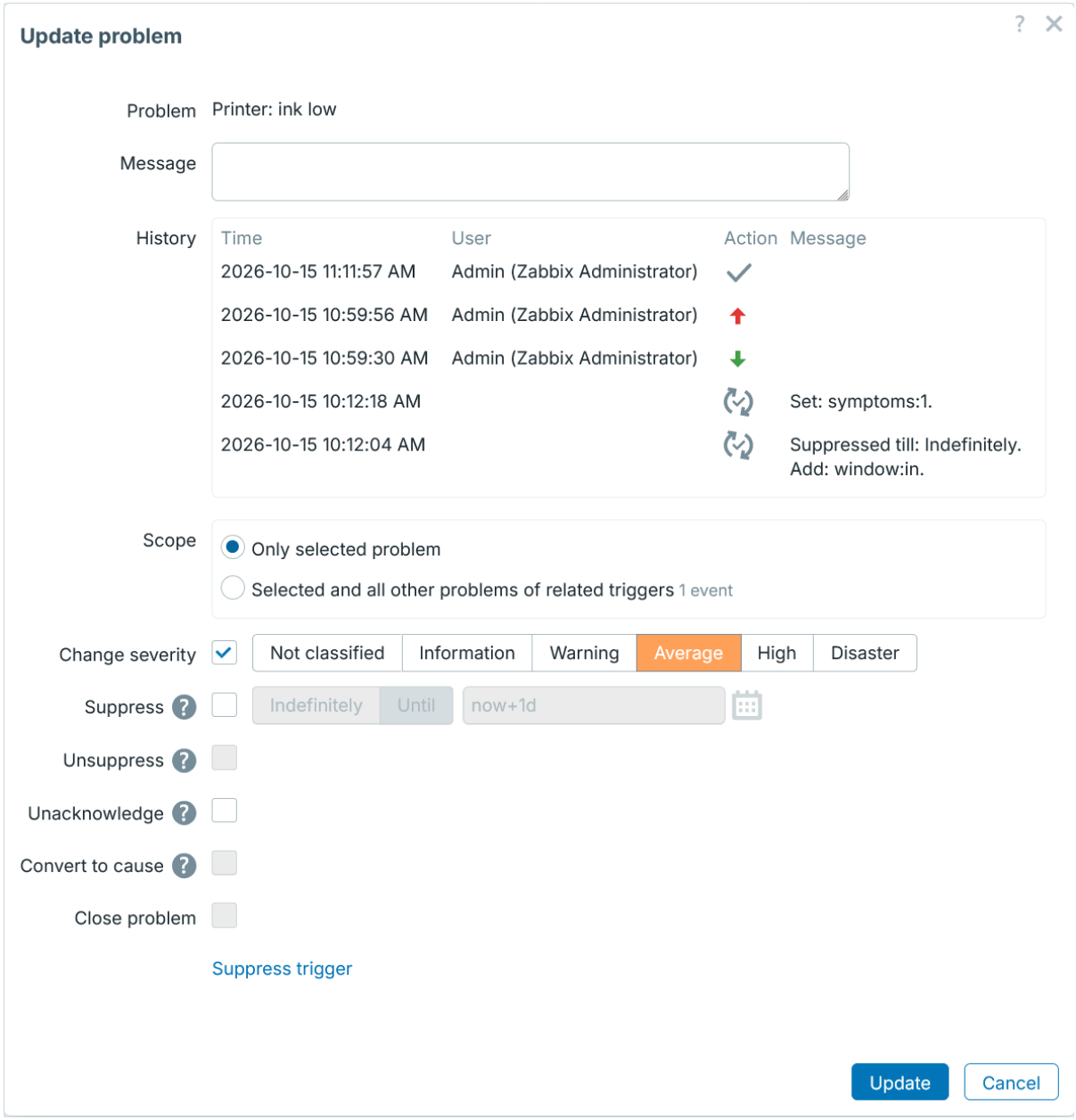
Voir aussi : Recevoir une notification sur les éléments non pris en charge

Accusé de réception et suppression des problèmes

Les événements de problème dans Zabbix peuvent être accusés de réception ou supprimés par les utilisateurs.

Mise à jour des problèmes La fenêtre contextuelle de mise à jour du problème vous permet de :

- Ajouter un commentaire au problème.
- Consulter les commentaires et les actions effectuées jusqu'à présent.
- Modifier la gravité du problème.
- Supprimer/rétablir la suppression du problème.
- Acquitter/annuler l'acquittement du problème.
- Convertir un problème symptôme en problème cause.
- Fermer manuellement le problème.
- Créer une maintenance ad hoc pour le déclencheur.



Tous les champs obligatoires sont signalés par un astérisque rouge.

Parameter	Description
Problème	Si un seul problème est sélectionné, le nom du problème est affiché. Si plusieurs problèmes sont sélectionnés, *N problèmes sélectionnés* est affiché.
Message	Saisissez le texte du commentaire à ajouter au problème (2048 caractères maximum).
Historique	Les activités et commentaires précédents concernant le problème sont répertoriés, avec l'heure et les informations sur l'utilisateur. Pour connaître la signification des icônes utilisées pour indiquer les actions des utilisateurs, consultez la page détails de l'événement. Notez que l'historique est affiché uniquement si un seul problème est sélectionné pour la mise à jour.
Portée	Définissez la portée des actions telles que la modification de la gravité, l'acquittement ou la fermeture manuelle des problèmes : **Problème sélectionné uniquement** - affectera uniquement cet événement **Problème sélectionné et tous les autres problèmes des déclencheurs associés** - dans le cas de l'acquittement ou de la fermeture d'un problème, affectera cet événement ainsi que tous les autres problèmes qui n'ont pas encore été acquittés ou fermés. Si la portée contient des problèmes déjà acquittés ou fermés, ceux-ci ne seront pas acquittés ou fermés à nouveau. En revanche, le nombre d'opérations de commentaire et de modification de la gravité n'est pas limité.

Parameter	Description
Modifier la gravité	Cochez la case, puis cliquez sur le bouton de gravité pour mettre à jour la gravité du problème. La case permettant de modifier la gravité est disponible si des permissions de lecture-écriture existent pour au moins un des problèmes sélectionnés. Seuls les problèmes pour lesquels l'accès en lecture-écriture est autorisé seront mis à jour lorsque vous cliquerez sur *Mettre à jour*. Si aucun des déclencheurs sélectionnés ne dispose de permissions de lecture-écriture, la case est désactivée.
Supprimer	Cochez la case pour supprimer le problème : **Indéfiniment** - supprimer indéfiniment **Jusqu'à** - supprimer jusqu'à une heure donnée. Les formats horaires absolu et relatif sont pris en charge, par exemple : `now+1d` - pendant un jour à partir de maintenant (valeur par défaut) `now/w` - jusqu'à la fin de la semaine en cours `2022-05-28 12:00:00` - jusqu'à une date/heure absolue Notez qu'une période simple (par ex. `1d`, `1w`) n'est pas prise en charge. La disponibilité de cette option dépend des paramètres du rôle utilisateur « Supprimer les problèmes ». Voir également : Suppression des problèmes
Rétablir la suppression	Cochez la case pour rétablir la suppression du problème. Cette case est active uniquement si au moins un des problèmes sélectionnés est supprimé. La disponibilité de cette option dépend des paramètres du rôle utilisateur « Supprimer les problèmes ».
Acquitter	Cochez la case pour acquitter le problème. Cette case est disponible si au moins un problème non acquitté figure parmi les problèmes sélectionnés. Il n'est pas possible d'ajouter un autre acquittement à un problème déjà acquitté (il est toutefois possible d'ajouter un autre commentaire).
Annuler l'acquittement	Cochez la case pour annuler l'acquittement du problème. Cette case est disponible si au moins un problème acquitté figure parmi les problèmes sélectionnés.
Convertir en cause	Cochez la case pour convertir le ou les problèmes symptômes en problèmes causes.
Fermer le problème	Cochez la case pour fermer manuellement le ou les problèmes sélectionnés. La case permettant de fermer un problème est disponible si l'option *Autoriser la fermeture manuelle* est cochée dans la configuration du déclencheur pour au moins un des problèmes sélectionnés. Seuls les problèmes pouvant être fermés seront fermés lorsque vous cliquerez sur *Mettre à jour*. Si aucun problème ne peut être fermé manuellement, la case est désactivée. Les problèmes déjà fermés ne seront pas fermés à nouveau.
Supprimer le déclencheur	Cliquez pour démarrer une maintenance ad hoc pour le déclencheur. Un formulaire de configuration de période de maintenance s'ouvrira. Une période de maintenance ponctuelle d'une heure (configurable dans les paramètres et le profil de l'utilisateur), commençant maintenant, y sera présélectionnée. Le déclencheur correspondant sera également présélectionné. Le nom de la maintenance sera prérempli au format `Ad-hoc: <event name>`.

Accusé de réception d'un problème Si un utilisateur est notifié d'un événement de problème, il peut se rendre dans l'interface Zabbix, ouvrir la fenêtre contextuelle de mise à jour du problème correspondant en utilisant l'une des méthodes □□□□ées ci-dessous et accuser réception du problème. Lors de l'accusé de réception, il peut saisir un commentaire, indiquant qu'il est en train de le traiter ou tout autre message qu'il souhaite ajouter à son sujet.

Ainsi, si un autre utilisateur du système constate le même problème, il voit immédiatement s'il a déjà été accusé de réception ainsi que les commentaires saisis jusqu'à présent.

De cette manière, le flux de travail de résolution des problèmes impliquant plusieurs utilisateurs du système peut se dérouler de façon coordonnée.

L'état d'accusé de réception est également utilisé lors de la définition des opérations d'action. Vous pouvez définir, par exemple, qu'une notification soit envoyée à un responsable de niveau supérieur uniquement si un événement n'a pas été accusé de réception pendant un certain temps.

Pour accuser réception d'événements et les commenter, un utilisateur doit disposer au minimum des autorisations de lecture sur les déclencheurs correspondants. Pour modifier la gravité d'un problème ou fermer un problème, un utilisateur doit disposer des autorisations de lecture-écriture sur les déclencheurs correspondants.

Il existe **plusieurs** façons d'accéder à la fenêtre contextuelle de mise à jour du problème, qui permet d'accuser réception d'un problème.

- Vous pouvez sélectionner des problèmes dans *Monitoring* > *Problems*, puis cliquer sur *Mass update* sous la liste
- Vous pouvez cliquer sur *Update* dans la colonne *Update* d'un problème dans :
  - *Dashboards* (widgets *Problems* et *Problems by severity*)
  - *Monitoring* > *Problems*
  - *Monitoring* > *Problems* > *Event details*
- Vous pouvez cliquer sur une cellule de problème non résolu dans :
  - *Dashboards* (widget *Trigger overview*)

Le menu contextuel contient une option *Update* qui vous mènera à la fenêtre de mise à jour du problème.

Selon les informations d'acquittement, il est possible de configurer la manière dont le nombre de problèmes est affiché dans le tableau de bord ou les cartes. Pour ce faire, vous devez effectuer des sélections dans l'option *Affichage des problèmes*, disponible à la fois dans la configuration de la carte et dans le widget de tableau de bord *Problèmes par gravité*.

Il est possible d'afficher le nombre total de problèmes, le nombre de problèmes non acquittés séparément du total, ou uniquement le nombre de problèmes non acquittés.

Selon les informations de mise à jour des problèmes (acquittement, etc.), il est possible de configurer des opérations de mise à jour : envoyer un message ou exécuter des commandes à distance.

Suppression des problèmes La suppression des problèmes permet de masquer temporairement un problème qui pourra être traité ultérieurement. Cette fonctionnalité est utile pour nettoyer la liste des problèmes afin d'accorder la priorité la plus élevée aux problèmes les plus urgents. Par exemple, un problème peut parfois survenir pendant le week-end sans être suffisamment urgent pour être traité immédiatement ; il peut alors être « mis en veille » jusqu'au lundi matin.

La suppression des problèmes permet de masquer un *seul* problème, contrairement à la suppression effectuée via la maintenance d'un hôte, qui masque tous les problèmes de l'hôte en maintenance.

Les opérations des actions de déclencheur sont suspendues pour les problèmes supprimés, de la même manière que pour la maintenance d'un hôte.

Un problème peut être supprimé via la fenêtre **mise à jour du problème**, où la suppression fait partie des options de mise à jour du problème, avec l'ajout de commentaires, la modification de la gravité, la prise en compte, etc.

Un problème peut également être réactivé via la même fenêtre de mise à jour du problème.

Affichage

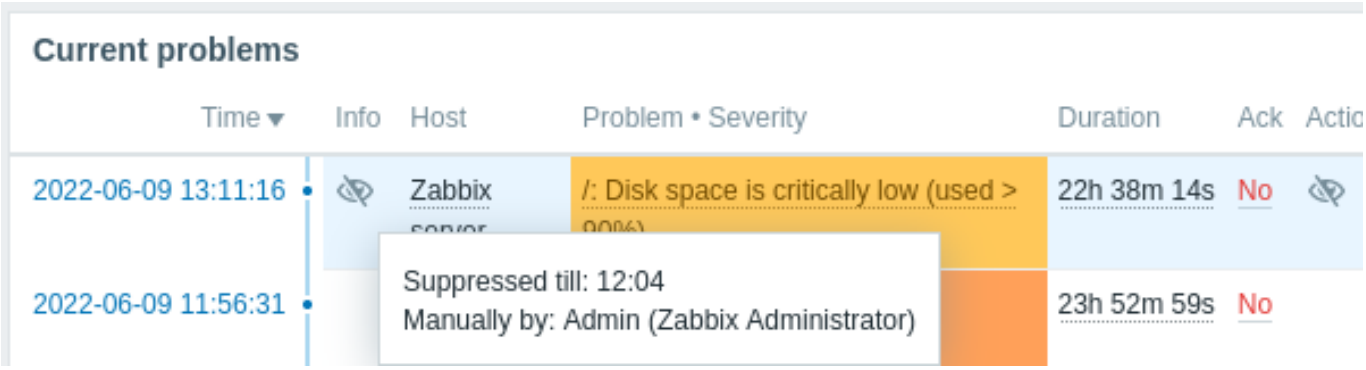
Une fois supprimé, le problème est signalé par une icône de suppression clignotante dans la colonne *Info*, avant d'être masqué.

Si la suppression est due à une maintenance de l'hôte, l'icône de suppression dans la colonne *Info* s'affiche en orange. Si la maintenance et la suppression manuelle s'appliquent toutes deux, la suppression manuelle est prioritaire pour la couleur de l'icône.

L'icône de suppression clignote tant que la tâche de suppression se trouve dans la liste d'attente. Une fois que le gestionnaire de tâches a supprimé le problème, l'icône cesse de clignoter. Si l'icône de suppression continue de clignoter pendant une longue période, cela peut indiquer un problème de serveur, par exemple si le serveur est arrêté et que le gestionnaire de tâches ne peut pas terminer la tâche. La même logique s'applique à la réactivation. Dans le court laps de temps qui suit l'envoi de la tâche et avant que le serveur ne l'ait terminée, l'icône de réactivation clignote.

Un problème supprimé peut être à la fois masqué ou affiché, selon les paramètres du filtre ou du widget de problèmes.

Lorsqu'il est affiché dans la liste des problèmes, un problème supprimé est signalé par l'icône de suppression et les détails de la suppression sont affichés au survol de la souris :



Les détails de la suppression sont également affichés dans une fenêtre contextuelle lorsque vous placez la souris sur l'icône de suppression dans la colonne *Actions*.

Fermeture manuelle des problèmes

Aperçu Bien que, en général, les événements de problème soient résolus automatiquement lorsque l'état du déclencheur passe de `Problem` à `OK`, il peut arriver qu'il soit difficile de déterminer si un problème a été résolu au moyen d'une expression de déclencheur. Dans ce cas, le problème doit être résolu manuellement.

Par exemple, *syslog* peut indiquer que certains paramètres du noyau doivent être ajustés pour obtenir des performances optimales. Dans ce cas, le problème est signalé aux administrateurs Linux, qui le corrigent puis ferment le problème manuellement.

Les problèmes ne peuvent être fermés manuellement que pour les déclencheurs pour lesquels l'option *Allow manual close* est activée.

Lorsqu'un problème est « fermé manuellement », Zabbix génère une nouvelle tâche interne pour le serveur Zabbix. Ensuite, le processus *task manager* exécute cette tâche et génère un événement OK, ce qui ferme l'événement de problème.

L'événement *OK* généré inclut l'ensemble complet des balises d'événement résolues pour cet événement (y compris les balises héritées des modèles, des hôtes et des déclencheurs). Ces balises sont disponibles dans les notifications et dans des macros telles que `{EVENT.RECOVERY.TAGS}` et `{EVENT.RECOVERY.TAGSJSON}`.

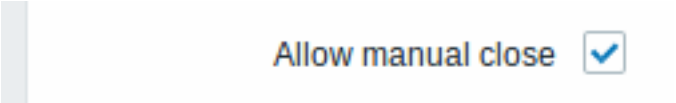
Le fait qu'un problème soit fermé manuellement ne signifie pas que le déclencheur sous-jacent ne repassera jamais à l'état *Problem*. L'expression du déclencheur est réévaluée et peut à nouveau produire un problème :

- Lorsque de nouvelles données arrivent pour tout élément inclus dans l'expression du déclencheur (notez que les valeurs ignorées par une étape de prétraitement de limitation de débit ne sont pas considérées comme reçues et ne provoqueront pas de réévaluation de l'expression du déclencheur) ;
- Lorsque les fonctions date and time et/ou nodata() sont utilisées dans l'expression.

Configuration Deux étapes sont nécessaires pour fermer manuellement un problème.

Configuration du déclencheur

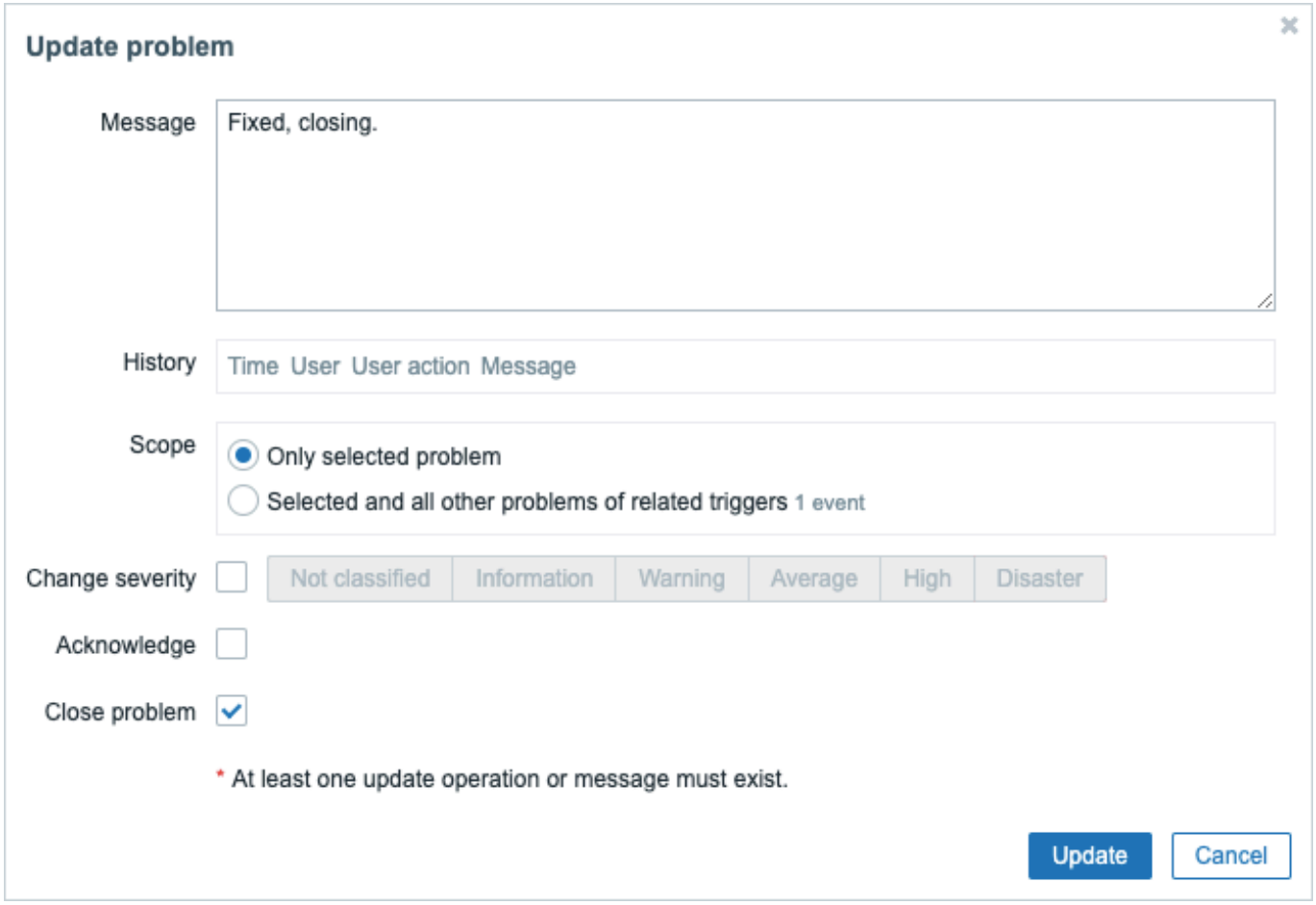
Dans la configuration du déclencheur, activez l'option *Autoriser la fermeture manuelle*.



Fenêtre de mise à jour du problème

Si un problème survient pour un déclencheur avec l'indicateur *Fermeture manuelle*, vous pouvez ouvrir la fenêtre contextuelle mise à jour du problème de ce problème et le fermer manuellement.

Pour fermer le problème, cochez l'option *Fermer le problème* dans le formulaire, puis cliquez sur *Mettre à jour*.



Tous les champs obligatoires sont marqués d'un astérisque rouge.

La requête est traitée par le serveur Zabbix. En général, il faut quelques secondes pour fermer le problème. Pendant ce processus, *CLOSING* s'affiche dans *Monitoring* > *Problems* comme statut du problème.

Vérification Il est possible de vérifier qu'un problème a été fermé manuellement :

- dans les détails de l'événement, accessibles via *Monitoring* > *Problems* ;
- en utilisant la macro `{EVENT.UPDATE.HISTORY}` dans les messages de notification, qui fournira cette information.

Périodes de maintenance

Vue d'ensemble

La maintenance permet de supprimer les problèmes pendant des périodes prédéfinies.

Vous pouvez définir des périodes de maintenance pour les entités suivantes :

- Groupes d'hôtes
- Hôtes
- Déclencheurs
- Noms d'événements
- Étiquettes

Si la maintenance est définie pour des étiquettes ou des noms d'événements spécifiques, elle sera activée uniquement pour les déclencheurs correspondants ; tous les autres déclencheurs de l'hôte ou du groupe d'hôtes ne seront pas en maintenance.

Il existe deux types de maintenance : *Avec collecte de données* et *Sans collecte de données*.

Pendant une maintenance *Avec collecte de données*, les déclencheurs sont traités normalement et les événements sont créés lorsque cela est nécessaire. Cependant, si l'option *Suspendre les opérations pour les problèmes supprimés* est cochée dans la configuration de l'action, les escalades de problèmes sont suspendues pour les entités en maintenance. Dans ce cas, les étapes d'escalade pouvant inclure l'envoi de notifications ou l'exécution de commandes à distance sont ignorées pendant toute la durée de la période de maintenance. Notez que les opérations de récupération et de mise à jour ne sont pas supprimées pendant la

maintenance, seules les escalades le sont. Les notifications de récupération ne sont pas envoyées si le problème a commencé pendant la maintenance.

Par exemple, si des étapes d'escalade sont planifiées à 0, 30 et 60 minutes après le début d'un problème, et qu'une maintenance d'une demi-heure a lieu de 10 à 40 minutes après l'apparition d'un problème réel, les étapes deux et trois seront exécutées une demi-heure plus tard, c'est-à-dire à 60 et 90 minutes (à condition que le problème existe toujours). De même, si un problème apparaît pendant la maintenance, l'escalade commencera après la maintenance.

Pour recevoir normalement les notifications de problèmes pendant la maintenance (sans délai), vous devez décocher l'option *Suspendre les opérations pour les problèmes supprimés* dans la configuration de l'action.

> **Note:**
> Si au moins un hôte (utilisé dans l'expression du déclencheur) n'est pas en mode maintenance, Zabbix enverra une notification de problème.

Le serveur Zabbix doit être en fonctionnement pendant la maintenance. Les maintenances sont recalculées chaque minute ou dès que le cache de configuration est rechargé si des modifications ont été apportées à la période de maintenance.

Les processus de temporisation vérifient si l'état de l'hôte doit passer en maintenance ou en sortir à 0 seconde de chaque minute. De plus, chaque seconde, le processus de temporisation vérifie si des maintenances doivent démarrer ou s'arrêter en fonction des modifications apportées aux périodes de maintenance après la mise à jour de la configuration. Ainsi, la vitesse de démarrage et d'arrêt des périodes de maintenance dépend de l'intervalle de mise à jour de la configuration (10 secondes par défaut). Notez que les modifications des périodes de maintenance n'incluent pas les paramètres *Actif depuis/Actif jusqu'à*. De même, si une entité (groupe d'hôtes/hôte/déclencheur/etc.) est ajoutée à une période de maintenance active existante, les modifications ne seront activées par le processus de temporisation qu'au début de la minute suivante.

Notez que lorsqu'un hôte entre en maintenance, les processus de temporisation du serveur Zabbix lisent tous les problèmes ouverts afin de vérifier s'il est nécessaire de les supprimer. Cela peut avoir un impact sur les performances s'il existe de nombreux problèmes ouverts. Le serveur Zabbix lit également tous les problèmes ouverts au démarrage, même si aucune maintenance n'est configurée à ce moment-là.

Notez que le serveur Zabbix (ou le proxy) collecte toujours les données, quel que soit le type de maintenance (y compris une maintenance *Sans collecte de données*). Les données sont ensuite ignorées par le serveur si l'option *Sans collecte de données* est définie.

Lorsqu'une maintenance *Sans collecte de données* se termine, les déclencheurs utilisant la fonction `nodata()` ne se déclencheront pas avant la prochaine vérification au cours de la période qu'ils contrôlent.

Si un élément de journal est ajouté alors qu'un hôte est en maintenance et que la maintenance se termine, seules les nouvelles entrées du fichier journal depuis la fin de la maintenance seront collectées.

Si une valeur horodatée est envoyée pour un hôte qui est en maintenance *Sans collecte de données* (par exemple, à l'aide de Zabbix sender), cette valeur sera ignorée ; toutefois, il est possible d'envoyer une valeur horodatée pour une période de maintenance expirée, et celle-ci sera acceptée.

Si la période de maintenance ou l'option de maintenance (groupe d'hôtes, hôte, déclencheur, etc.) est modifiée par l'utilisateur, les modifications ne prendront effet qu'après la synchronisation du cache de configuration.

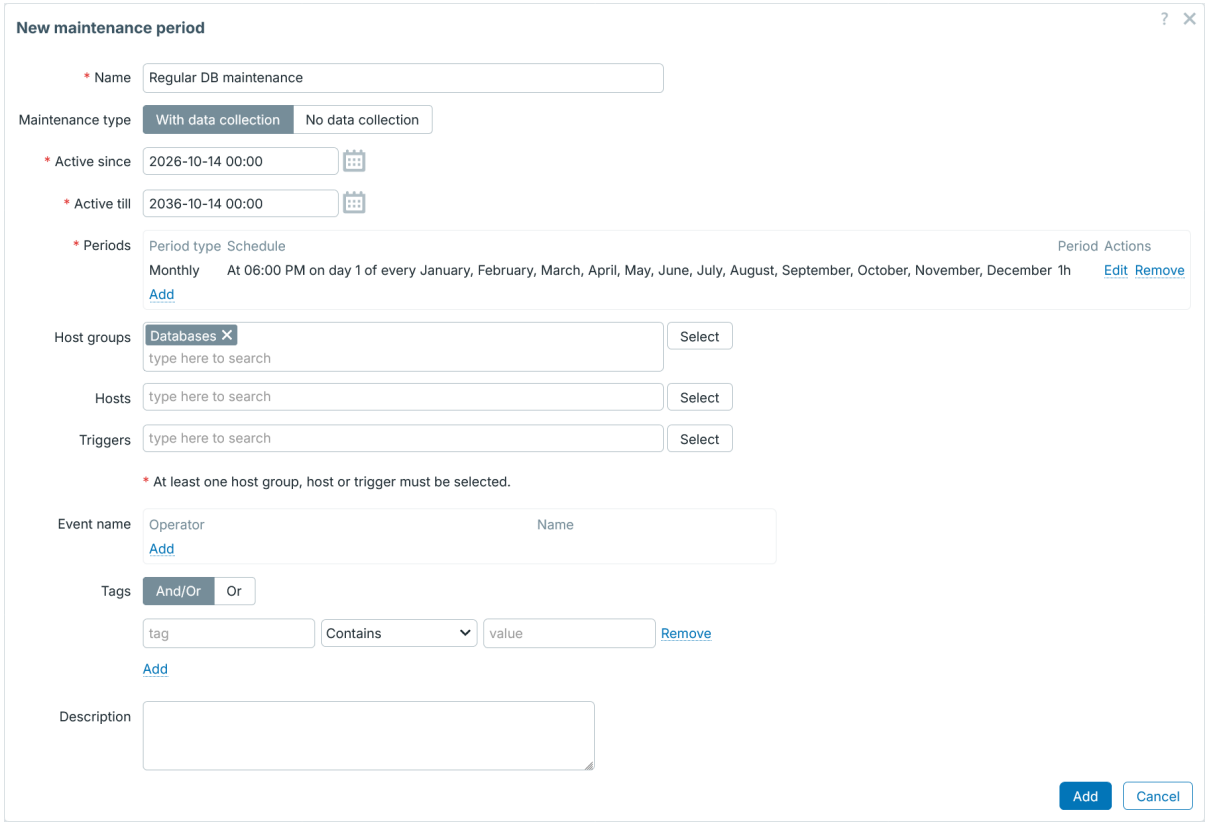
Configuration

Pour démarrer immédiatement une période de maintenance :

1. Accédez à *Monitoring* > *Problems*.
2. Cliquez sur le nom du problème.
3. Sélectionnez l'option de maintenance dans le menu de l'événement (hôte, déclencheur, nom de l'événement, étiquettes).
4. Modifiez les paramètres de maintenance dans le formulaire.

Pour configurer une période de maintenance planifiée :

1. Accédez à *Data collection* > *Maintenance*.
2. Cliquez sur *Create maintenance period* (ou sur le nom d'une période de maintenance existante).
3. Saisissez les paramètres de maintenance dans le formulaire.



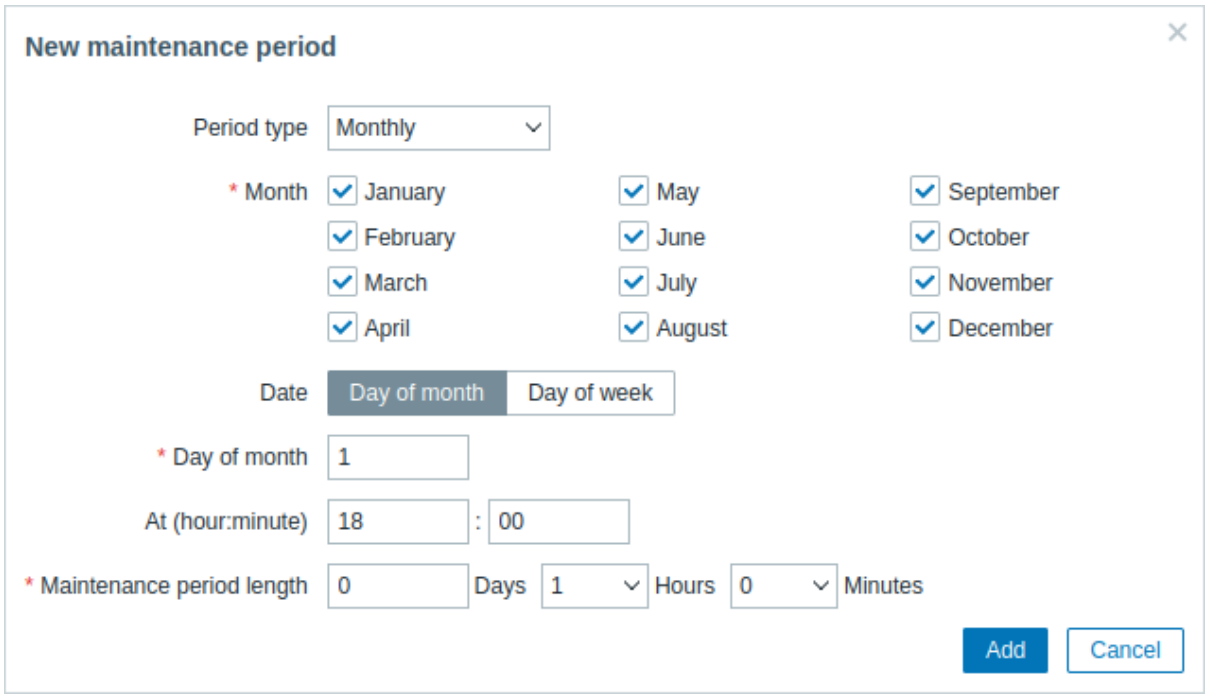
Tous les champs obligatoires sont signalés par un astérisque rouge.

Parameter	Description
Name	Nom de la période de maintenance.
Maintenance type	Deux types de maintenance peuvent être définis : **With data collection** - les données sont collectées par le serveur pendant la maintenance et les déclencheurs sont traités ; **No data collection** - les données peuvent toujours être collectées, mais elles ne sont pas enregistrées dans la base de données pendant la maintenance et les déclencheurs (y compris la fonction `nodata()`) ne se déclenchent pas. Consultez Effet des périodes de maintenance pour savoir comment chaque type affecte le rapport de disponibilité.
Active since	Date et heure à partir desquelles l'exécution des périodes de maintenance devient active. *Remarque :* le fait de définir cette heure seule n'active pas une période de maintenance ; les périodes de maintenance doivent être configurées dans *Periods* (voir ci-dessous).
Active till	Date et heure jusqu'auxquelles l'exécution des périodes de maintenance reste active.
Periods	Ce bloc vous permet de définir les jours et les heures exacts auxquels la maintenance a lieu. Cliquez sur Add pour ouvrir une fenêtre contextuelle contenant un formulaire flexible de *Maintenance period*, dans lequel vous pouvez définir le calendrier de maintenance. Consultez Périodes de maintenance pour obtenir une description détaillée.
Host groups	Sélectionnez les groupes d'hôtes pour lesquels la maintenance sera activée. La maintenance sera activée pour tous les hôtes des groupes d'hôtes spécifiés. Ce champ propose la saisie semi-automatique : commencez à saisir du texte pour afficher une liste déroulante de tous les groupes d'hôtes disponibles. La spécification d'un groupe d'hôtes parent sélectionne implicitement tous les groupes d'hôtes imbriqués. La maintenance sera donc également activée sur les hôtes des groupes imbriqués.
Hosts	Sélectionnez les hôtes pour lesquels la maintenance sera activée. Ce champ propose la saisie semi-automatique : commencez à saisir du texte pour afficher une liste déroulante de tous les hôtes disponibles.
Triggers	Sélectionnez les déclencheurs pour lesquels la maintenance sera activée. Ce champ propose la saisie semi-automatique : commencez à saisir du texte pour afficher une liste déroulante de tous les déclencheurs disponibles.

Parameter	Description
Event name	Spécifiez les noms d'événements pour lesquels la maintenance sera activée. Plusieurs conditions peuvent être définies. Deux opérateurs sont disponibles pour chaque condition : **Contains** - inclure les noms d'étiquettes spécifiés lorsque les valeurs des étiquettes contiennent la chaîne saisie (correspondance de sous-chaîne, insensible à la casse) **Does not contain** - exclure les noms d'étiquettes spécifiés lorsque les valeurs des étiquettes contiennent la chaîne saisie (correspondance de sous-chaîne, insensible à la casse)
Tags	Spécifiez les étiquettes pour lesquelles la maintenance sera activée. Plusieurs conditions peuvent être définies. La correspondance des noms d'étiquettes est toujours sensible à la casse. Plusieurs opérateurs sont disponibles pour chaque condition : **Contains** - inclure les noms d'étiquettes spécifiés lorsque les valeurs des étiquettes contiennent la chaîne saisie (correspondance de sous-chaîne, sensible à la casse) ; **Equals** - inclure les noms et valeurs d'étiquettes spécifiés (sensible à la casse). **Does not equal** - exclure les noms et valeurs d'étiquettes spécifiés (sensible à la casse) **Does not contain** - exclure les noms d'étiquettes spécifiés lorsque les valeurs des étiquettes contiennent la chaîne saisie (correspondance de sous-chaîne, insensible à la casse) Deux types de calcul sont disponibles pour les conditions : **And/Or** - toutes les conditions doivent être remplies ; les conditions ayant le même nom d'étiquette sont regroupées avec la condition Or ; **Or** - une seule condition remplie suffit. Les étiquettes peuvent être spécifiées uniquement si le type de maintenance *With data collection* est sélectionné.
Description	Description de la période de maintenance.

Périodes de maintenance

La fenêtre de période de maintenance sert à planifier le temps d'une maintenance récurrente ou ponctuelle. Le formulaire est dynamique, les champs disponibles changeant en fonction du *Type de période* sélectionné.



Type de période	Description
Une seule fois	Configurez une période de maintenance ponctuelle : *Date* - date et heure de la période de maintenance ; *Durée de la période de maintenance* - durée pendant laquelle la maintenance sera active.

Type de période	Description
Quotidienne	Configurez une période de maintenance quotidienne : *Tous les jour(s)* - fréquence de la maintenance (1 - *(par défaut)* tous les jours, 2 - tous les deux jours, etc.) ; *À (heure:minute)* - heure de la journée à laquelle la maintenance commence ; *Durée de la période de maintenance* - durée pendant laquelle la maintenance sera active. Lorsque le paramètre *Tous les jour(s)* est supérieur à "1", le jour de départ est celui auquel correspond l'heure *Actif depuis*. Exemples : - si *Actif depuis* est défini sur "2021-01-01 12:00", *Tous les jour(s)* est défini sur "2" et *À (heure:minute)* est défini sur "23:00", alors la première période de maintenance commencera le 1er janvier à 23:00, tandis que la deuxième période de maintenance commencera le 3 janvier à 23:00 ; - si *Actif depuis* est défini sur "2021-01-01 12:00", *Tous les jour(s)* est défini sur "2" et *À (heure:minute)* est défini sur "01:00", alors la première période de maintenance commencera le 3 janvier à 01:00, tandis que la deuxième période de maintenance commencera le 5 janvier à 01:00.
Hebdomadaire	Configurez une période de maintenance hebdomadaire : *Toutes les semaine(s)* - fréquence de la maintenance (1 - *(par défaut)* chaque semaine, 2 - toutes les deux semaines, etc.) ; *Jour de la semaine* - jour auquel la maintenance doit avoir lieu ; *À (heure:minute)* - heure de la journée à laquelle la maintenance commence ; *Durée de la période de maintenance* - durée pendant laquelle la maintenance sera active. Lorsque le paramètre *Toutes les semaine(s)* est supérieur à "1", la semaine de départ est celle à laquelle correspond l'heure *Actif depuis*. Pour des exemples, voir la description du paramètre *Quotidienne* ci-dessus.
Mensuelle	Configurez une période de maintenance mensuelle : *Mois* - sélectionnez tous les mois pendant lesquels la maintenance régulière est effectuée ; *Date:* ***Jour du mois*** - sélectionnez cette option si la maintenance doit avoir lieu à la même date chaque mois (par exemple, chaque 1er jour du mois), puis sélectionnez le jour requis dans le champ *Jour du mois* qui apparaît ; *Date:* ***Jour de la semaine*** - sélectionnez cette option si la maintenance doit avoir lieu uniquement certains jours (par exemple, chaque premier lundi du mois), puis sélectionnez (dans la liste déroulante) la semaine du mois requise (première, deuxième, troisième, quatrième ou dernière), puis cochez les cases correspondant au(x) jour(s) de maintenance ; *À (heure:minute)* - heure de la journée à laquelle la maintenance commence ; *Durée de la période de maintenance* - durée pendant laquelle la maintenance sera active.

Attention:
Lors de la création d'une période de maintenance, le fuseau horaire de l'utilisateur qui la crée est utilisé. Cependant, lorsque des périodes de maintenance récurrentes (*Daily*, *Weekly*, *Monthly*) sont planifiées, le fuseau horaire du serveur Zabbix est utilisé. Pour garantir un comportement prévisible des périodes de maintenance récurrentes, il est nécessaire d'utiliser un fuseau horaire commun pour toutes les parties de Zabbix.

Lorsque vous avez terminé, cliquez sur *Add* pour ajouter la période de maintenance à la section *Periods*.

Notez que les changements d'heure d'été (DST) n'affectent pas la durée de la maintenance. Par exemple, supposons qu'une maintenance de deux heures soit configurée et qu'elle commence généralement à 01:00 et se termine à 03:00 :

- si, après une heure de maintenance (à 02:00), un changement DST se produit et que l'heure actuelle passe de 02:00 à 03:00, la maintenance se poursuivra pendant une heure supplémentaire (jusqu'à 04:00) ;
- si, après deux heures de maintenance (à 03:00), un changement DST se produit et que l'heure actuelle passe de 03:00 à 02:00, la maintenance s'arrêtera, car deux heures se sont écoulées ;
- si une période de maintenance commence pendant l'heure supprimée par un changement DST, alors la maintenance ne démarrera pas.

Si une période de maintenance est définie sur "1 day" (la période réelle de la maintenance est de 24 heures, car Zabbix calcule les jours en heures), commence à 00:00 et se termine à 00:00 le jour suivant :

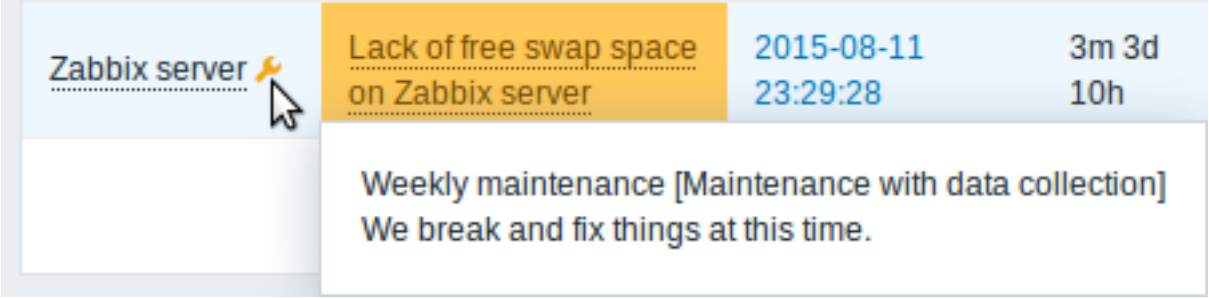
- la maintenance s'arrêtera à 01:00 le jour suivant si l'heure actuelle avance d'une heure ;
- la maintenance s'arrêtera à 23:00 ce jour-là si l'heure actuelle recule d'une heure.

Affichage

Affichage des hôtes en maintenance

Une icône de clé orange à côté du nom de l'hôte indique que cet hôte est en maintenance dans :

- *Dashboards*
- *Monitoring* > *Problèmes*
- *Inventory* > *Hosts* > *Host inventory details*
- *Data collection* > *Hosts* (voir la colonne *Status*)



Les détails de la maintenance s'affichent lorsque le pointeur de la souris est placé sur l'icône.

De plus, les hôtes en maintenance ont un arrière-plan orange dans *Monitoring* > *Maps*.

Affichage des problèmes masqués

Normalement, les problèmes des hôtes en maintenance sont masqués, c'est-à-dire qu'ils ne sont pas affichés dans l'interface.

Les problèmes masqués peuvent également être cachés parce qu'ils ont été masqués manuellement.

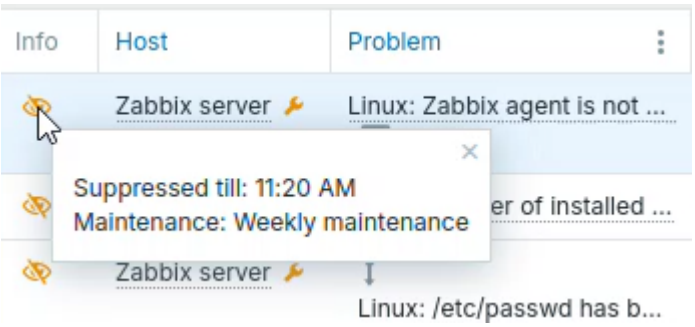
Il est possible d'afficher les problèmes masqués en sélectionnant l'option *Afficher les problèmes masqués* aux emplacements suivants :

- *Tableaux de bord* (dans la configuration des widgets *Hôtes avec problèmes*, *Problèmes*, *Problèmes par gravité*, *Vue d'ensemble des déclencheurs*)
- *Surveillance* > *Problèmes* (dans le filtre)
- *Surveillance* > *Cartes* (dans la configuration de la carte)
- notifications globales (dans la configuration du profil utilisateur)

Lorsque les problèmes masqués sont affichés, l'icône suivante est visible : . Si le masquage est dû à la maintenance de l'hôte, les icônes suivantes sont affichées :

- - le problème a été masqué en raison de la maintenance de l'hôte.
- - le problème a été réaffiché après la maintenance de l'hôte.

Le survol de l'icône affiche plus de détails. Dans les détails de l'événement, vous pouvez vérifier si l'événement a été masqué en raison de la maintenance ou masqué manuellement. Si la maintenance et le masquage manuel s'appliquent tous deux, le masquage manuel est prioritaire dans la couleur de l'icône.



Calcul du calcul des files d'attente pendant la maintenance

Les files d'attente affichées dans l'interface Zabbix (Administration > Queue) sont calculées par le serveur Zabbix. Elles n'incluent pas les éléments en maintenance *No data collection*, et la longueur de la file d'attente est toujours nulle pour ces éléments, même lorsque leurs valeurs sont en retard. Les éléments en retard en maintenance *With data collection* sont toujours comptabilisés dans la file d'attente.

Le proxy Zabbix n'a pas connaissance des périodes de maintenance, car il n'existe aucune synchronisation de la configuration de maintenance entre le serveur Zabbix et le proxy. Les vérifications internes calculées sur les proxies Zabbix (par exemple,

`zabbix[queue,,]` et `zabbix[stats,,,queue,,]`) signalent les éléments en retard indépendamment de l'état de maintenance sur le serveur Zabbix.

Par conséquent, des longueurs de file d'attente différentes peuvent être signalées pour les mêmes éléments en maintenance *No data collection* par l'interface Zabbix et par les vérifications internes sur les proxies Zabbix.

Étiquetage

Les tags ont plusieurs usages, notamment pour marquer des événements. Lorsque des entités sont taguées, tout nouvel événement lié à une entité taguée héritera de ses tags. Par exemple :

- avec des modèles tagués - tout problème d'hôte (créé par des déclencheurs du modèle) héritera des tags du modèle.
- avec des hôtes tagués - tout problème d'hôte héritera des tags de l'hôte.
- avec des éléments/scénarios web tagués - tout problème d'élément/scénario web héritera des tags de l'élément/scénario web.
- avec des déclencheurs tagués - tout problème créé par le déclencheur héritera des tags du déclencheur.

Un événement de problème hérite de tous les tags de l'ensemble de la chaîne d'entités - modèles, hôtes, éléments/scénarios web, déclencheurs. Les combinaisons `tag:value` identiques (après résolution des macros) sont fusionnées en une seule, évitant ainsi les doublons.

Les événements de récupération générés par une fermeture manuelle incluent également les tags d'événement résolus hérités des modèles, hôtes, éléments/scénarios web et déclencheurs. Ces tags sont disponibles dans les notifications et via des macros telles que `{EVENT.RECOVERY.TAGS}` et `{EVENT.RECOVERY.TAGSJSON}`.

Les tags d'événement personnalisés offrent davantage de flexibilité. Par exemple :

- la corrélation d'événements peut être configurée en fonction des tags d'événement.
- les conditions d'action peuvent être configurées en fonction des tags d'événement.
- les problèmes d'élément peuvent être regroupés en fonction des tags d'événement.
- les tags de problème peuvent être utilisés pour associer des problèmes à des services.

Les entités peuvent être taguées avec le même nom de tag mais des valeurs de tag différentes (par exemple, `component:memory` et `component:storage`). De même, une entité peut avoir un tag sans valeur et le même tag avec une valeur (par exemple, `database` et `database:postgresql`). De tels tags ne sont pas considérés comme des doublons.

Les tags se composent d'un nom de tag et d'une valeur de tag. Lors du taggage des entités, vous pouvez utiliser uniquement le nom ou l'associer à une valeur (par exemple, `mysql`, `jira`, `target:mysql`, `service:jira`, etc.).

Les tags peuvent être définis pour diverses entités :

- Modèles
- Hôtes
- Éléments
- Scénarios web
- Déclencheurs
- Services
- Éléments et déclencheurs de modèle
- Prototypes d'hôte, d'élément et de déclencheur

Dans les listes *Modèles*, *Hôtes*, *Éléments*, *Déclencheurs*, *Scénarios web* et leurs prototypes, la colonne *Tags* affiche à la fois les tags normaux et hérités. Au survol ou au clic d'un tag *hérité*, une info-bulle JavaScript apparaît avec le texte "Inherited tag". Si un tag hérité existe également en tant que tag normal, un texte d'info-bulle différent est affiché selon la liste où il apparaît (par exemple, "Inherited and template tag" dans la liste modèles ou "Inherited and host tag" dans la liste hôtes). Dans les autres parties de l'interface, les tags normaux et hérités sont affichés ensemble (les doublons sont supprimés), sans icônes et sans texte d'info-bulle supplémentaire.

> **Note:**
> Reportez-vous aux recommandations officielles de Zabbix pour les recommandations générales sur la définition des tags, ainsi qu'aux consignes spécifiques pour les modèles, les éléments, les déclencheurs et les règles de découverte de bas niveau.

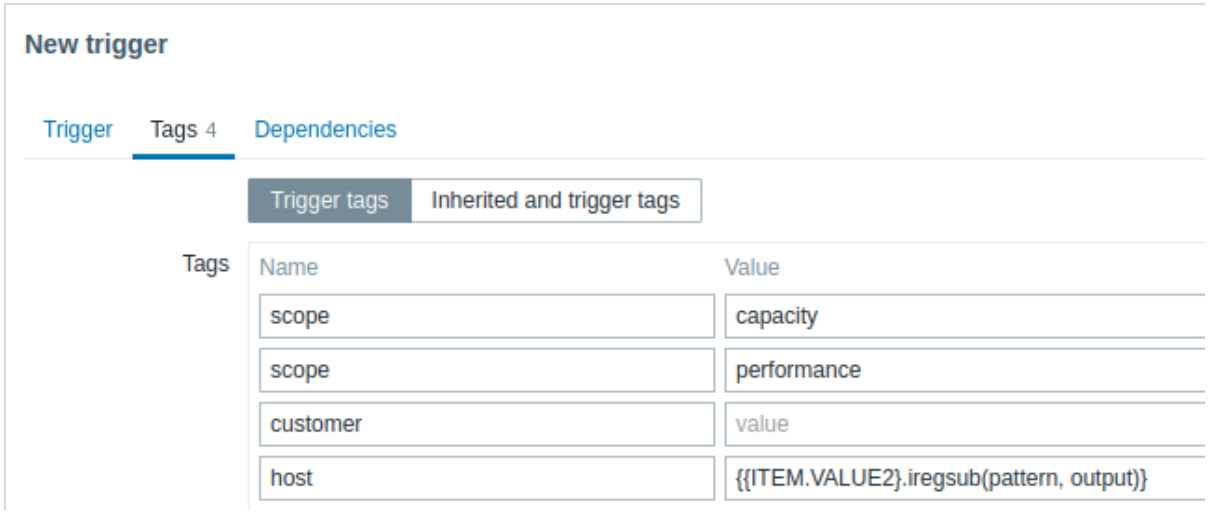
Cas d'utilisation Voici quelques cas d'utilisation courants du marquage :

1. Marquer les événements de déclencheur :
   - Définissez une balise de déclencheur (par exemple, `scope:performance`).

- Les problèmes créés par ce déclencheur auront la balise de déclencheur.

2. Marquer les problèmes hérités du modèle :
   - Définissez une balise de modèle (par exemple, `target:mysql`).
   - Les problèmes créés par des déclencheurs issus de ce modèle auront la balise de modèle.
3. Marquer les problèmes de l'hôte :
   - Définissez une balise d'hôte (par exemple, `service:jira`).
   - Les problèmes créés par des déclencheurs de cet hôte auront la balise d'hôte.
4. Filtrer les éléments associés :
   - Définissez une balise d'élément (par exemple, `component:cpu`).
   - Dans *Monitoring* > *Latest data*, les éléments peuvent être filtrés par la balise `component:cpu`.
5. Utiliser comme valeur de balise les informations extraites de la valeur de l'élément :
   - Définissez une balise avec une macro comme valeur de balise (par exemple, `tag-name:{{ITEM.VALUE<N>}.regsub()}`).
   - Dans *Monitoring* > *Problems*, les problèmes auront une valeur de balise résolue à partir des données extraites de la valeur de l'élément.
6. Identifier les problèmes dans un fichier journal et les fermer séparément :
   - Définissez une balise de déclencheur pour le déclencheur de l'élément de surveillance des journaux qui extraira des valeurs de la valeur de l'élément à l'aide d'une macro (par exemple, `service:{{ITEM.VALUE<N>}.regsub()}`).
   - Dans la configuration du déclencheur, configurez la corrélation d'événements :
     - définissez le *mode de génération des événements PROBLEM* sur "Multiple" ;
     - définissez *OK event closes* sur "All problems if tag values match" ;
     - définissez la balise pour la correspondance.
   - Les problèmes créés par le déclencheur de l'élément de journal auront la balise de déclencheur et seront fermés individuellement.
7. Filtrer les notifications :
   - Définissez des balises de déclencheur (par exemple, `scope:security` pour trigger1 et `scope:availability` pour trigger2).
   - Utilisez le filtrage par balises dans les conditions d'action pour recevoir des notifications uniquement sur les événements correspondant aux données de balise.
8. Identifier les problèmes dans les notifications :
   - Définissez des balises de déclencheur.
   - Utilisez la macro {EVENT.TAGS} dans la notification de problème.
   - La notification de problème contiendra les balises de déclencheur, ce qui facilitera l'identification de l'application ou du service auquel la notification appartient.
9. Simplifier les tâches de configuration en utilisant des balises de modèle :
   - Définissez une balise de déclencheur de modèle.
   - Les déclencheurs créés à partir de ce déclencheur de modèle auront sa balise.
10. Créer des déclencheurs avec des balises issues de la découverte de bas niveau (LLD) :
    - Définissez une balise de prototype de déclencheur avec une macro LLD dans le nom ou la valeur de la balise (par exemple, `scope:{#FSNAME}`).
    - Les déclencheurs créés à partir du prototype de déclencheur auront sa balise.
11. Faire correspondre des services à l'aide de balises de service :
    - Définissez des balises de service.
    - Configurez des actions de service pour les services ayant des balises correspondantes.
    - Utilisez également les balises de service pour lier un service à un SLA pour les calculs de SLA.
12. Lier des services aux problèmes à l'aide de balises de problème de service :
    - Définissez une balise de problème dans la configuration du service (par exemple, `target:mysql`).
    - Les problèmes ayant une balise correspondante seront automatiquement corrélés au service, et l'état du service changera en fonction des règles de calcul de l'état du service configurées.
13. Masquer les problèmes lorsqu'un hôte est en mode maintenance :
    - Définissez des balises dans la configuration de la période de maintenance.
    - Les problèmes ayant les balises définies seront masqués.
14. Accorder l'accès à des groupes d'utilisateurs :
    - Définissez des balises dans la configuration du groupe d'utilisateurs.
    - Les utilisateurs du groupe d'utilisateurs pourront voir uniquement les problèmes ayant les balises définies.

Configuration Les tags peuvent être définis dans un onglet dédié, par exemple, dans la configuration du déclencheur :



Prise en charge des macros Les macros intégrées et les macros utilisateur dans les tags sont résolues au moment de l'événement. Tant que l'événement ne s'est pas produit, ces macros seront affichées dans l'interface Zabbix sans être résolues.

Les macros de découverte de bas niveau sont résolues pendant le processus de découverte.

Les macros suivantes peuvent être utilisées dans les noms et valeurs des tags de déclencheur :

- {ITEM.VALUE}, {ITEM.VALUE.AGE}, {ITEM.VALUE.DATE}, {ITEM.VALUE.TIME}, {ITEM.VALUE.TIMESTAMP}, {ITEM.LASTVALUE}, {ITEM.LASTVALUE.AGE}, {ITEM.LASTVALUE.DATE}, {ITEM.LASTVALUE.TIME}, {ITEM.LASTVALUE.TIMESTAMP}, {HOST.HOST}, {HOST.NAME}, {HOST.CONN}, {HOST.DNS}, {HOST.IP}, {HOST.PORT} et {HOST.ID} macros intégrées
- {INVENTORY.*} macros intégrées (pour référencer les valeurs d'inventaire de l'hôte à partir d'un ou plusieurs hôtes dans une expression de déclencheur)
- Macros utilisateur et macros utilisateur avec contexte (le contexte peut inclure des macros de découverte de bas niveau)
- Macros de découverte de bas niveau (uniquement dans les tags de prototype de déclencheur)

Les macros suivantes peuvent être utilisées dans les noms et valeurs des tags de modèle, d'hôte et d'élément/scénario web :

- {HOST.HOST}, {HOST.NAME}, {HOST.CONN}, {HOST.DNS}, {HOST.IP}, {HOST.PORT} et {HOST.ID} macros intégrées
- {INVENTORY.*} macros intégrées
- Macros utilisateur
- Macros de découverte de bas niveau (uniquement dans les tags de prototype d'hôte et d'élément)

Les macros suivantes peuvent être utilisées dans les notifications basées sur des déclencheurs :

- {EVENT.TAGS} et {EVENT.RECOVERY.TAGS} macros intégrées (ces macros seront résolues en une liste de tags d'événement ou de tags d'événement de récupération séparés par des virgules)
- {EVENT.TAGSJSON} et {EVENT.RECOVERY.TAGSJSON} macros intégrées (ces macros seront résolues en un tableau JSON contenant des objets de tag d'événement objects ou des objets de tag d'événement de récupération)

Extraction de sous-chaîne dans les tags de déclencheur

L'extraction de sous-chaîne est prise en charge pour renseigner le nom du tag ou la valeur du tag, à l'aide d'une fonction de macro. La fonction applique une expression régulière à la valeur obtenue par la macro prise en charge. Par exemple :

```
{{ITEM.VALUE}.regsub(pattern, output)}
{{ITEM.VALUE}.iregsub(pattern, output)}

{{#LLDMACRO}.regsub(pattern, output)}
{{#LLDMACRO}.iregsub(pattern, output)}
```

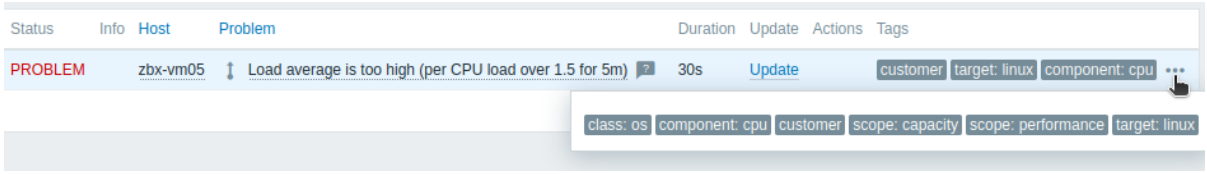
Si le nom ou la valeur du tag dépasse 255 caractères après résolution de la macro, il sera tronqué à 255 caractères.

Voir aussi : Utilisation des fonctions de macro dans les macros de découverte bas niveau pour le marquage des événements.

Affichage des tags d'événement Les tags, s'ils sont définis, peuvent être consultés avec les nouveaux événements dans :

- *Monitoring* > *Problèmes*
- *Monitoring* > *Problèmes* > *Détails de l'événement*
- *Tableaux de bord* > *widget Problèmes*

L'ordre et le nombre de tags affichés sont déterminés par les options de filtrage *Priorité d'affichage des tags* et *Afficher les tags* dans *Monitoring* > *Problèmes* ou dans le widget de tableau de bord *Problèmes*. Notez qu'un maximum de trois tags peut être affiché ; s'il y a plus de tags, le survol des trois points affiche tous les tags dans une fenêtre contextuelle.



Complex event processing

Overview Complex event processing (CEP) performs rule-based processing for Zabbix events.

For example, you might want to:

- group related events and mark first event in a series as cause (the following as symptoms)
- specify a time-to-live period for events
- specify a maximum number for certain events
- be notified about similar events by location
- be notified if some event happens too often (or not at all) within a time period

In addition to single event processing, you may define **time windows** for event analysis. Within a time window, all matching events form a single context and can be analyzed as a whole.

A sliding time window moves forward with the time. Older events leave the window (are "evicted"), while new events may enter.

A tumbling time window does not move forward. When the window expires, a new window with the same duration is started. *Cause and symptom grouping* windows are a tumbling time window.

The time window concept presents logical points for executing operations. For example, an event can be transformed or "enriched" when it enters the window - severity can be raised or lowered, tags can be added or adjusted, suppression can be activated, etc.

Similar operations can be executed when the event leaves the window or when a pattern is found.

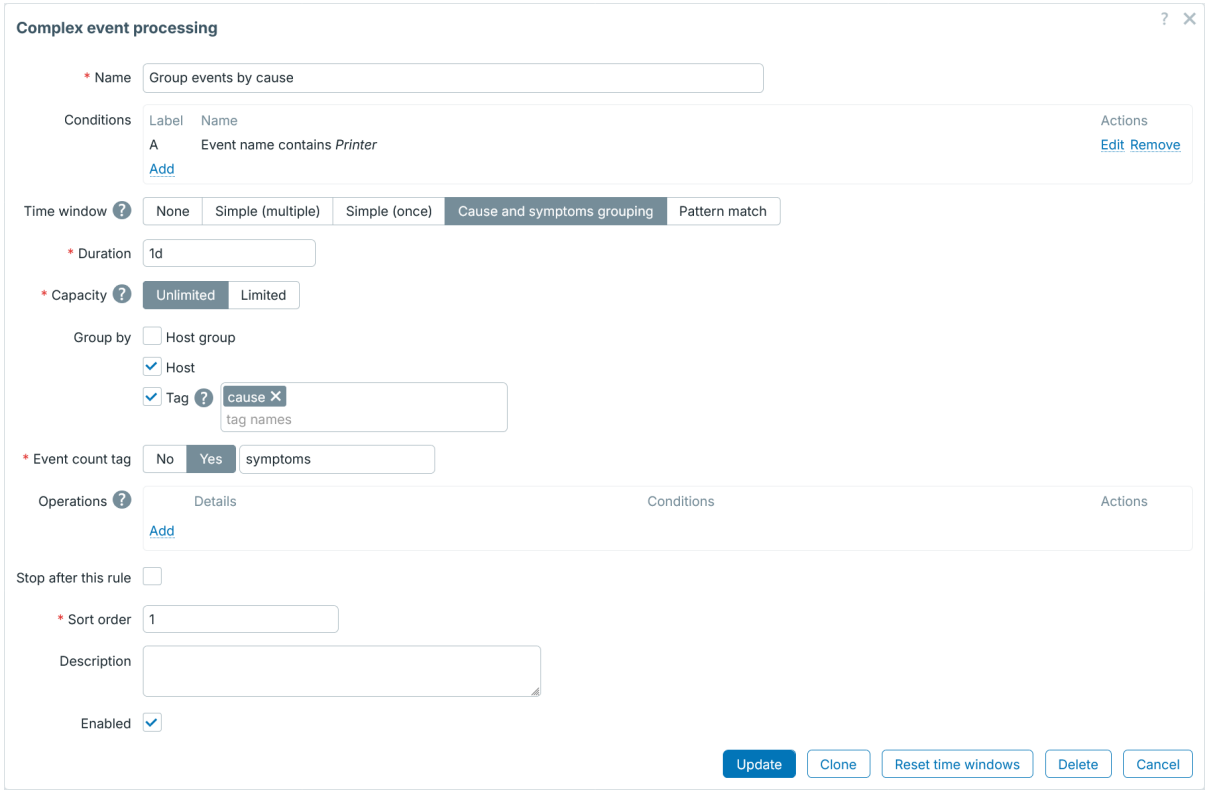
Event stream analysis is defined by CEP rules. A CEP rule consists of a time window (optional), conditions and operations. Operations are tied to an execution point.

Use cases

- Group related events, with the first such event automatically marked as cause (all following as symptoms).
- Enrich event with higher severity/specific tag if too many 401 events happen within a minute (possible bot activity).

Configuration To create a CEP rule in Zabbix frontend, do the following:

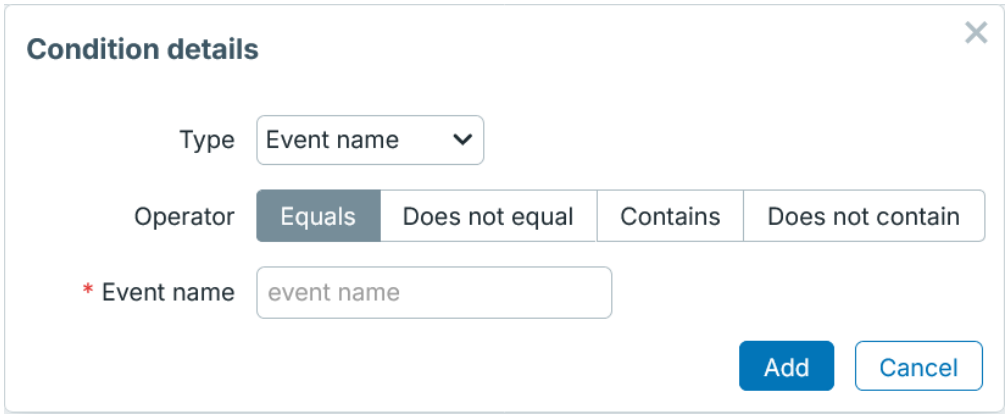
1. Go to *Data collection* > *Event processing*.
2. Click *Create complex event processing* in the upper-right corner of the screen.
3. Enter parameters of the rule in the form.



Parameter	Description
Name	Specify a unique CEP rule name.
Conditions	Specify conditions for associating events with this CEP rule.
Time window	Select the time window type: **None** - process single events with no time window (default) **Simple (multiple)** - process events in a sliding time window (event can be added to multiple windows of this type) **Simple (once)** - process events in a sliding time window (event can be added only to one window of this type) **Cause and symptom grouping** - process first event in time window as cause, consecutive events as symptoms **Pattern match** - analyze sliding time window events based on a JavaScript script. The window is examined every second; events older than the duration are evicted, and the script is then run over the events that remain in it. When the script reports a match, the pattern match operations are executed for every event the window holds; the events stay in the window and are examined again a second later, until they age out. A window can be forcibly closed by the ”Close window” operation.
Duration	Limit the time window duration (in seconds, 365 days max). The window duration is started by the first matching event. In *Simple* and *Pattern match* windows, the window is moving so older events will leave the window (are evicted) after the duration passes. In *Cause and symptom grouping* windows, window duration is a fixed timeframe for setting cause and symptom relationship. The window is started when the first event matches. When the duration expires, the window is closed and a new one is started. No event is evicted when the duration expires; periods run back to back. Time suffixes are supported (`10m`, `1h`, etc). User macros are supported.
Capacity	Specify the event capacity within the time window: **Unlimited** - event count is not limited **Limited** - event count is limited by the specified number Once the limit is surpassed, the latest event is evicted. User macros are supported.
Group by	Group events by host group, host, and/or event tag. This option is available for CEP rules with a time window. If specified, the rule will create multiple windows based on specified criteria; otherwise events will be grouped into a single window group.

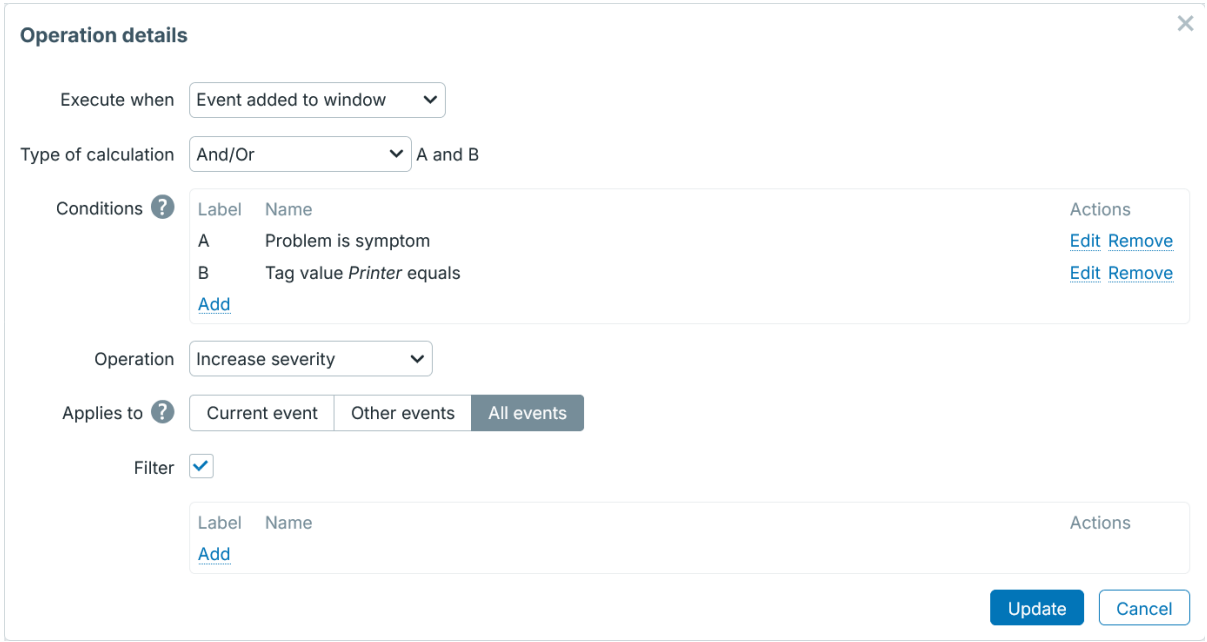
Parameter	Description
Event count tag	Specify name for a tag displaying symptom event count. Symptom event count will be displayed as `<tag name>:<event count>` (for example: `symptoms:2`). This field is only available if *Cause and symptom grouping* is selected as time window type.
Script	Specify the script for identifying event patterns. This field is only available if *Pattern match* is selected as time window type.
Operations	Click on *Add* to specify operations.
Stop after this rule	Mark the checkbox if no further processing is required after this rule. Otherwise, continue with the next CEP rule according to the sort order.
Sort order	Specify position of this CEP rule in the processing order. If the order priority is the same, the rules will be executed alphabetically.
Description	Add the CEP rule description.

CEP rule condition



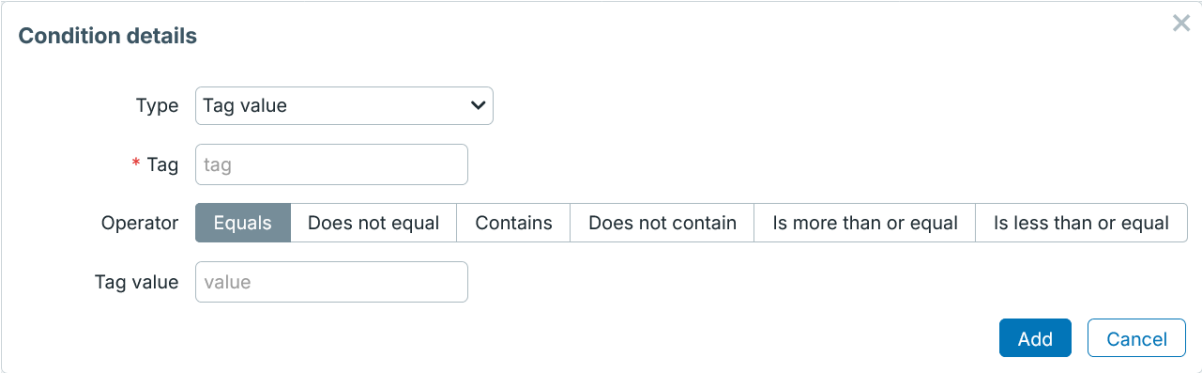
Parameter	Description
Type	Select condition type. Event filtering can be based on event name, tag name, tag value, severity, host group, host (visible name), or time period.

CEP rule operation



Parameter	Description
Execute when	Select execution point: **Event occurred** - when event occurs. This is the only stage where *Discard* operation is supported, since event is not actually created yet and can be discarded. Window-related operations are not available at this point. Changes made to the event in *Event occurred* phase (except closing) are visible to the next CEP rules trying to match this event. For example, if rule A added tag X and rule B has condition ”Tag X exists”, then it will match rule A and B. Of course, rule A should not be configured to stop after matching. **Event added to window** - when event is added to time window. Operations are executed after event is added to matched CEP-rule windows. If window is at its capacity, then *Event evicted* operations are executed instead. **Event evicted** - when event leaves the window. Operations are executed before event is evicted from a window because of duration or capacity. **Event closed** - when event is closed. **Window closed** - when time window is closed. Operations are executed on all window events when window is closed because of *Close window* operation or as expired cause-symptom window. **Pattern matched** - when pattern is matched. Operations are executed on all window events when the pattern match JavaScript returns `true`. Note that the *Event occurred*, *Event added to window* and *Event evicted* have fixed execution order, while *Window closed* and *Pattern matched* operate independently.
Conditions	Select additional conditions for the operation.
Operation	Select operation. **Events**: *Set name* - set event name *Close* - close event *Discard* - discard event. Only available at *Event occurred* stage. *Set severity* - set event severity *Increase severity* - increase event severity *Decrease severity* - decrease event severity *Suppress* - suppress event *Unsuppress* - unsuppress event *Clone first* - clone first event. Useful for producing a summary event based on the first event. Only available at *Event evicted*, *Window closed*, or *Pattern matched* stages. *Clone last* - clone last event. Useful for producing a summary event based on the last event. Only available at *Event evicted*, *Window closed*, or *Pattern matched* stages. For event cloning to work either the source (first/last) event must be resolved or its trigger must have multi-event generation enabled. **Tags**: *Add* - add tag *Set* - set tag and value *Set value* - set tag value *Increase value* - increase tag value *Decrease value* - decrease tag value *Rename* - rename tag *Remove* - remove tag **Window**: *Close (window)* - close event window
Applies to	Select operation scope: **Current event** - apply operation to the current event **Other events** - apply operation to all events except the current one **All events** - apply operation to all events
Filter	Select additional filter for the operation scope. This field is available if *Applies to* is set ”Other events” or ”All events”.

CEP rule operation condition



Parameter	Description
Type	Select condition type. Event filtering can be based on tag name, tag value, problem is opened/closed, problem is symptom/not symptom, first event in time window, last event in time window, problem is suppressed/not suppressed, event is cloned/not cloned.

Corrélation des événements

Vue d'ensemble

La corrélation des événements permet d'associer les événements de problème à leur résolution de manière très précise et flexible.

La corrélation des événements peut être définie :

- Au niveau du déclencheur - un déclencheur peut être utilisé pour associer des problèmes distincts à leur résolution ;
- Globalement - les problèmes peuvent être associés à leur résolution à partir d'un déclencheur ou d'une méthode d'interrogation différente, à l'aide de règles de corrélation globales.

Corrélation d'événements basée sur les déclencheurs

Vue d'ensemble

La corrélation des événements basée sur les déclencheurs permet de mettre en corrélation des problèmes distincts signalés par un même déclencheur.

En général, un événement OK peut clôturer tous les événements de problème créés par un même déclencheur, mais certaines situations nécessitent une approche plus détaillée. Par exemple, lors de la surveillance de fichiers journaux, vous pouvez vouloir détecter certains problèmes dans un fichier journal et les clôturer individuellement plutôt que tous ensemble.

C'est le cas des déclencheurs dont le paramètre *PROBLEM event generation mode* est défini sur *Multiple*. Ces déclencheurs sont généralement utilisés pour la surveillance des journaux, le traitement des traps, etc.

Dans Zabbix, il est possible de mettre en relation les événements de problème en fonction de leur étiquetage. Les étiquettes servent à extraire des valeurs et à créer un identifiant pour les événements de problème. En tirant parti de cette fonctionnalité, les problèmes peuvent également être clôturés individuellement en fonction d'une étiquette correspondante.

En d'autres termes, le même déclencheur peut créer des événements distincts identifiés par l'étiquette de l'événement. Les événements de problème peuvent donc être identifiés un par un et clôturés séparément en fonction de leur identification par l'étiquette de l'événement.

Comment cela fonctionne

Dans la surveillance des journaux, vous pouvez rencontrer des lignes similaires à celles-ci :

```
Ligne1: Service 1 stopped
Ligne2: Service 2 stopped
Ligne3: Service 1 was restarted
Ligne4: Service 2 was restarted
```

L'idée de la corrélation d'événements est de pouvoir associer l'événement de problème de la Ligne1 à la résolution de la Ligne3 et l'événement de problème de la Ligne2 à la résolution de la Ligne4, puis de fermer ces problèmes un par un :

```
Ligne1: Service 1 stopped
Ligne3: Service 1 was restarted #problem from Line 1 closed

Ligne2: Service 2 stopped
Ligne4: Service 2 was restarted #problem from Line 2 closed
```

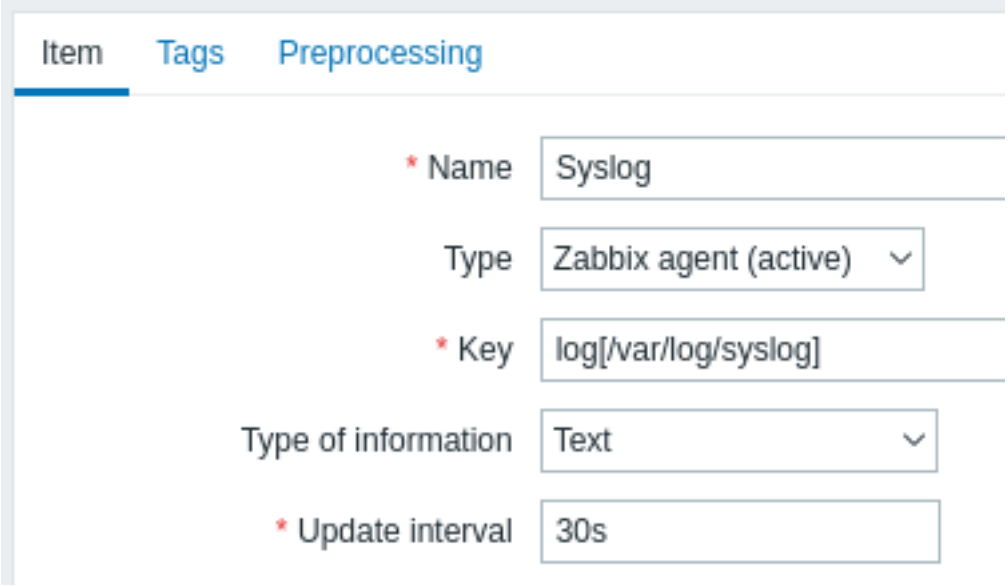
Pour cela, vous devez étiqueter ces événements liés, par exemple, ”Service 1” et ”Service 2”. Cela peut être fait en appliquant une expression régulière à la ligne du journal afin d’extraire la valeur de l’étiquette. Ensuite, lorsque les événements sont créés, ils sont étiquetés respectivement ”Service 1” et ”Service 2”, et le problème peut être associé à la résolution.

Configuration

Élément

Pour commencer, vous pouvez configurer un élément qui surveille un fichier journal, par exemple :

```
log[/var/log/syslog]
```



Une fois l’élément configuré, attendez une minute que les modifications de configuration soient prises en compte, puis allez dans Latest data pour vous assurer que l’élément a commencé à collecter des données.

Déclencheur

Avec l’élément en fonctionnement, vous devez configurer le déclencheur. Il est important de décider quelles entrées du fichier journal méritent votre attention. Par exemple, l’expression de déclencheur suivante recherchera une chaîne comme ’Stopping’ pour signaler des problèmes potentiels :

```
find(/My host/log[/var/log/syslog],,"regexp","Stopping")=1
```

Attention:
Pour vous assurer que chaque ligne contenant la chaîne ”Stopping” soit considérée comme un problème, définissez également le *Mode de génération des événements de problème* dans la configuration du déclencheur sur ’Multiple’.

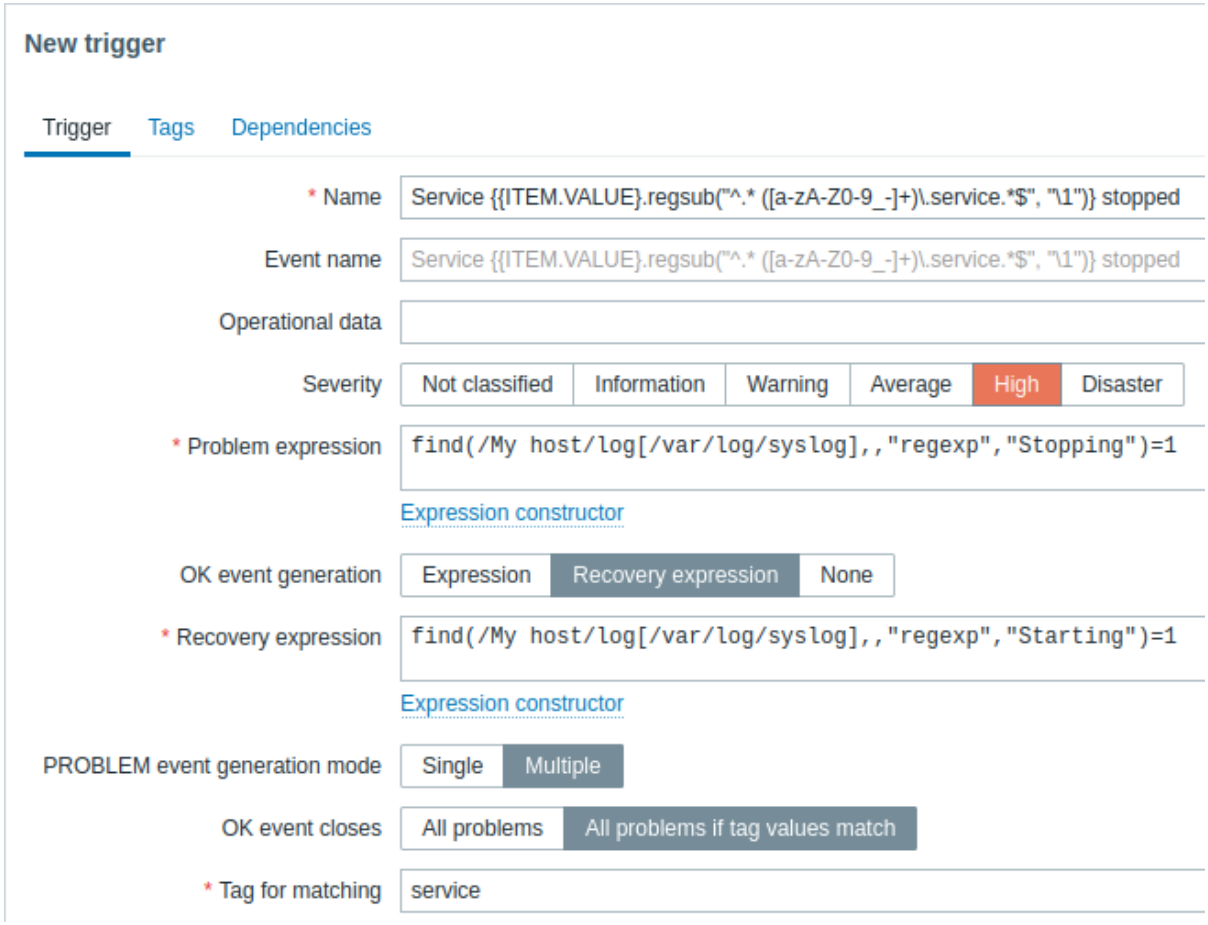
Définissez ensuite une expression de rétablissement. L’expression de rétablissement suivante résoudra tous les problèmes si une ligne du journal contenant la chaîne ”Starting” est trouvée :

```
find(/My host/log[/var/log/syslog],,"regexp","Starting")=1
```

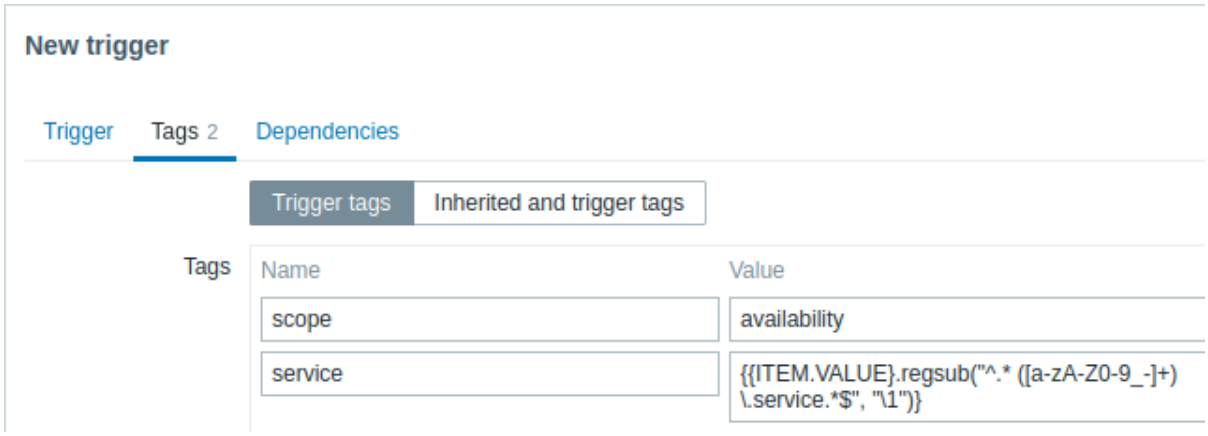
Comme nous ne le voulons pas, il est important de s’assurer d’une manière ou d’une autre que les problèmes racine correspondants soient fermés, et pas seulement tous les problèmes. C’est là que le marquage peut aider.

Les problèmes et les rétablissements peuvent être mis en correspondance en spécifiant un tag dans la configuration du déclencheur. Les paramètres suivants doivent être définis :

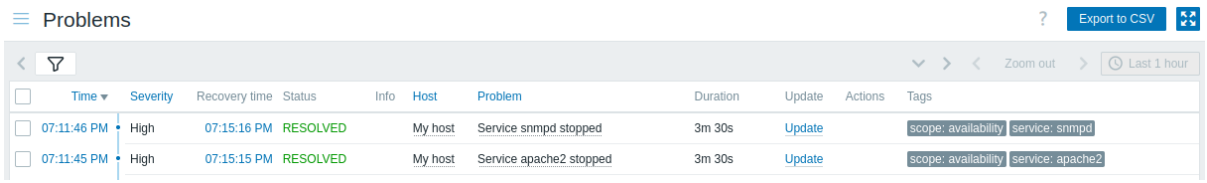
- *Mode de génération des événements de problème* : Multiple
- *L’événement OK ferme* : Tous les problèmes si les valeurs des tags correspondent
- Saisissez le nom du tag pour la correspondance des événements



- configurez les tags pour extraire les valeurs des tags à partir des lignes du journal



Si la configuration est correcte, vous pourrez voir les événements de problème tagués par application et associés à leur rétablissement dans *Monitoring* > *Problèmes*.



Warning:
Comme une mauvaise configuration est possible, et que des tags d'événements similaires peuvent être créés pour des problèmes **sans rapport**, veuillez examiner les cas décrits ci-dessous !

- Les macros indexées font toujours référence au champ *Expression* de la configuration du déclencheur, et non à l'*expression de rétablissement*. Par exemple, dans un événement de rétablissement, **{ITEM.VALUE1}** résoudra la dernière valeur du premier élément **dans l'expression de problème** au moment du rétablissement. Si l'expression de rétablissement est basée sur un élément différent et que la valeur de l'élément de l'expression de problème change d'ici le rétablissement, les

événements auront des tags différents et ne seront pas corrélés.

- Avec deux applications écrivant des messages d'erreur et de rétablissement dans le même fichier journal, un utilisateur peut décider d'utiliser deux tags *service* dans le même déclencheur avec des valeurs de tag différentes en utilisant des expressions régulières distinctes dans les valeurs des tags pour extraire les noms, par exemple, du service A et du service B à partir de la macro {ITEM.VALUE} (par exemple, lorsque les formats des messages diffèrent). Cependant, cela peut ne pas fonctionner comme prévu s'il n'y a pas de correspondance avec les expressions régulières. Les expressions régulières sans correspondance produiront des valeurs de tag vides, et une seule valeur de tag vide dans les événements de problème et OK suffit pour les corréler. Ainsi, un message de rétablissement du service A peut accidentellement fermer un message d'erreur du service B.
- Les tags réels et les valeurs des tags ne deviennent visibles que lorsqu'un déclencheur se déclenche. Si l'expression régulière utilisée est invalide, elle est silencieusement remplacée par une chaîne *UNKNOWN*. Si l'événement de problème initial avec une valeur de tag *UNKNOWN* est manqué, il peut apparaître ensuite des événements OK avec la même valeur de tag *UNKNOWN* qui peuvent fermer des événements de problème qu'ils n'auraient pas dû fermer.
- Si un utilisateur utilise la macro {ITEM.VALUE} sans fonctions de macro comme valeur de tag, la limite de 255 caractères s'applique. Lorsque les messages du journal sont longs et que les 255 premiers caractères ne sont pas spécifiques, cela peut également entraîner des tags d'événements similaires pour des problèmes sans rapport.

Corrélation globale des événements

Présentation

La corrélation globale des événements permet d'étendre la corrélation à toutes les métriques surveillées par Zabbix et de créer des corrélations.

Il est possible de corréler des événements créés par des déclencheurs complètement différents et de leur appliquer à tous les mêmes opérations. En créant des règles de corrélation intelligentes, il est ainsi possible de se débarrasser de milliers de notifications répétitives et de se concentrer sur les causes profondes d'un problème !

La corrélation globale des événements est un mécanisme puissant qui permet de s'affranchir d'une logique de problème et de résolution basée sur un seul déclencheur. Jusqu'à présent, un seul événement de problème était créé par un déclencheur et nous dépendions de ce même déclencheur pour résoudre le problème. Nous ne pouvions pas résoudre un problème créé par un déclencheur avec un autre déclencheur. Mais grâce à la corrélation des événements basée sur les tags d'événement, c'est désormais possible.

Par exemple, un déclencheur de journal peut signaler des problèmes liés à une application, tandis qu'un déclencheur d'interrogation peut signaler que l'application est opérationnelle. En utilisant les tags d'événement, vous pouvez attribuer au déclencheur de journal le tag *status:down* et au déclencheur d'interrogation le tag *status:up*. Ensuite, dans une règle de corrélation globale, vous pouvez associer ces déclencheurs et attribuer une opération appropriée à cette corrélation, comme la fermeture des anciens événements.

Dans un autre cas d'utilisation, la corrélation globale peut identifier des déclencheurs similaires et leur appliquer la même opération. Et s'il était possible d'obtenir un seul rapport de problème par problème de port réseau ? Il ne serait pas nécessaire de tous les signaler. La corrélation globale des événements permet également cela.

La corrélation globale des événements est configurée dans les **règles de corrélation**. Une règle de corrélation définit comment les nouveaux événements de problème sont associés aux événements de problème existants et quelle action effectuer en cas de correspondance (fermer le nouvel événement, fermer les anciens événements correspondants en générant les événements OK correspondants). Si un problème est fermé par la corrélation globale, cela est indiqué dans la colonne *Info* de *Surveillance* > *Problèmes*.

La configuration des règles de corrélation globale est réservée aux utilisateurs de niveau Super administrateur.

> **Attention:**
> La corrélation des événements doit être configurée avec beaucoup de précautions, car elle peut affecter négativement les performances du traitement des événements ou, si elle est mal configurée, fermer plus d'événements que prévu (dans le pire des cas, tous les événements de problème pourraient être fermés).

Pour configurer les règles de corrélation globale **en toute sécurité**, suivez les recommandations importantes suivantes :

- Réduisez la portée de la corrélation. Définissez toujours un tag unique pour le nouvel événement associé aux anciens événements et utilisez la condition de corrélation *Nom du tag du nouvel événement*.
- Ajoutez une condition explicite pour les anciens événements lors de l'utilisation de **Fermer les anciens événements**. Ajoutez toujours au moins une condition *Ancien événement* (par exemple, *Nom du tag de l'ancien événement*, *Valeur du tag de l'ancien événement* ou *Paire de tags d'événement*) lorsque vous sélectionnez **Fermer les anciens événements** ;

sinon, la règle peut correspondre à des événements de problème existants sans rapport et les fermer (dans le pire des cas, tous les problèmes). Privilégiez *Paire de tags d'événement* pour faire correspondre les valeurs d'exécution (hôte:port, ID de session, etc.) et réduisez davantage la correspondance en utilisant l'hôte ou le groupe d'hôtes lorsque cela est possible.

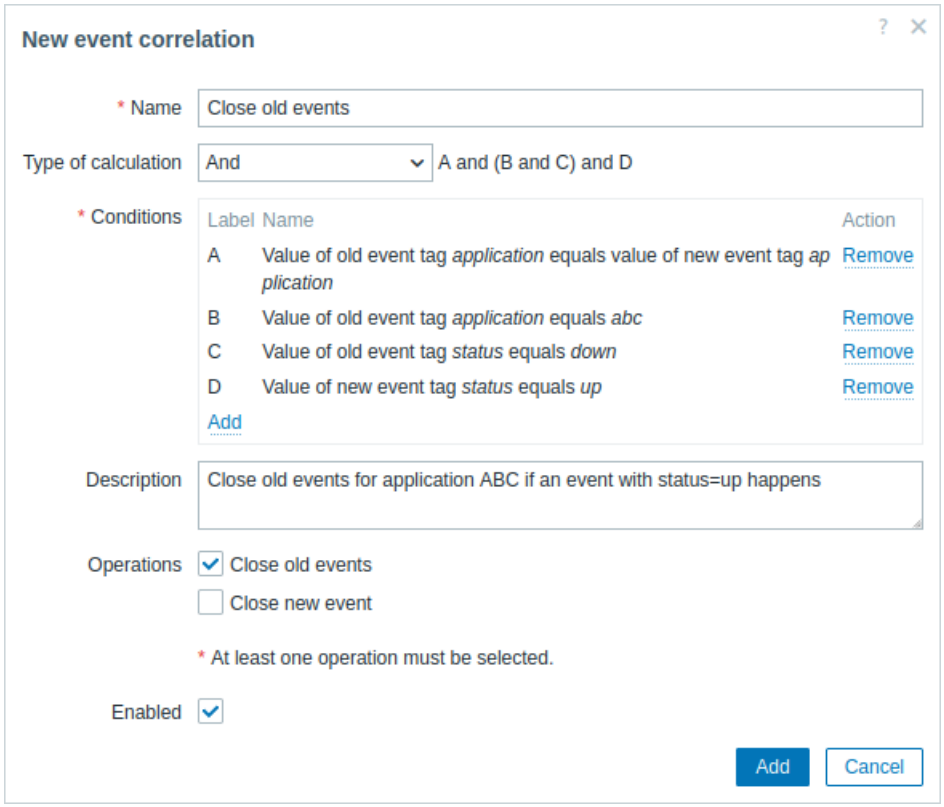
- Évitez d'utiliser des noms de tags courants qui pourraient être utilisés par différentes configurations de corrélation.
- Limitez le nombre de règles de corrélation à celles dont vous avez réellement besoin.

Voir également : problèmes connus.

Configuration

Pour configurer globalement les règles de corrélation d'événements :

1. Allez dans *Data collection* > *Event correlation*.
2. Cliquez sur *Create event correlation* à droite (ou sur le nom de la corrélation pour modifier une règle existante).
3. Saisissez les paramètres de la règle de corrélation dans le formulaire.

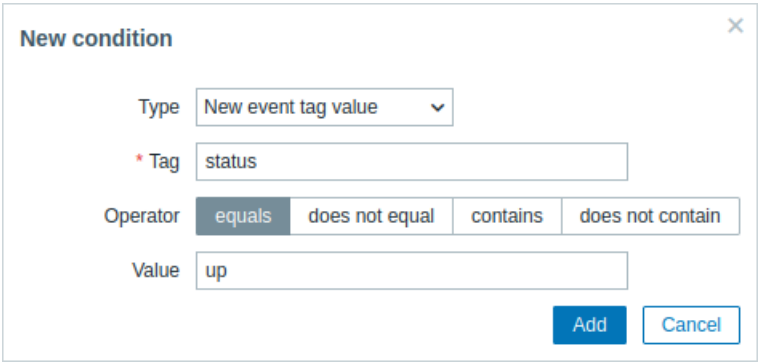


Tous les champs obligatoires sont marqués d'un astérisque rouge.

Parameter	Description
Name	Nom unique de la règle de corrélation.
Type of calculation	Les options suivantes de calcul des conditions sont disponibles : **And** - toutes les conditions doivent être remplies **Or** - il suffit qu'une condition soit remplie **And/Or** - AND avec des types de conditions différents et OR avec le même type de condition **Custom expression** - une formule de calcul définie par l'utilisateur pour évaluer les conditions de l'action. Elle doit inclure toutes les conditions (représentées par des lettres majuscules A, B, C, ...) et peut inclure des espaces, des tabulations, des parenthèses ( ), **and** (sensible à la casse), **or** (sensible à la casse), **not** (sensible à la casse).
Conditions	Liste des conditions. Voir ci-dessous pour plus de détails sur la configuration d'une condition.
Description	Description de la règle de corrélation.

Parameter	Description
Operations	Cochez l'opération à effectuer lorsqu'un événement est corrélé. Les opérations suivantes sont disponibles : **Close old events** - fermer les anciens événements lorsqu'un nouvel événement se produit. Ajoutez toujours une condition basée sur l'ancien événement lors de l'utilisation de l'opération *Close old events*, sinon tous les problèmes existants pourraient être fermés. **Close new event** - fermer le nouvel événement lorsqu'il se produit. **Warning!** Ne laissez pas les conditions d'ancien/nouvel événement vides lors de l'utilisation de *Close old events*/*Close new event*. Si vous sélectionnez l'opération *Close old events* sans ajouter de condition correspondant à l'ancien événement, Zabbix peut faire correspondre tous les anciens événements existants et les fermer. Ajoutez toujours une condition explicite d'ancien événement (par exemple, *Old event tag name* ou *Event tag pair*) lors de l'utilisation de *Close old events*. Par exemple, une règle qui utilise uniquement une *New event condition* et l'opération *Close old events* fera correspondre tous les anciens événements qui satisfont aux critères d'ancien événement (absents) - fermant ainsi effectivement les anciens problèmes.
Enabled	Si vous cochez cette case, la règle de corrélation sera activée.

Pour configurer les détails d'une nouvelle condition, cliquez sur Add dans le bloc Conditions. Une fenêtre contextuelle s'ouvrira, dans laquelle vous pourrez modifier les détails de la condition.



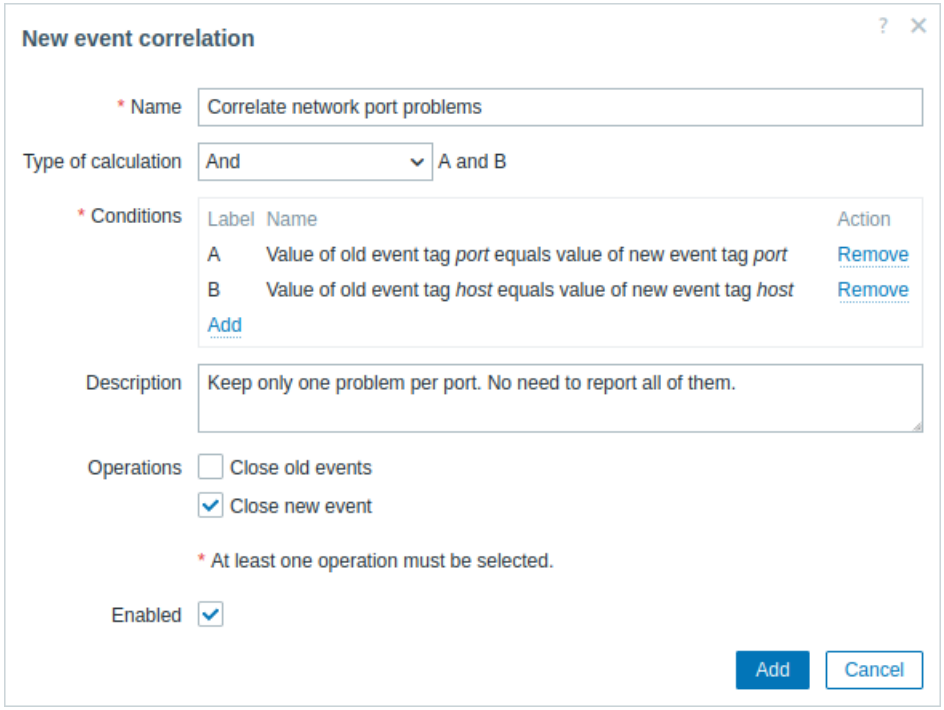
Parameter	Description
New condition	Sélectionnez une condition pour corréler les événements. *Note* que si aucune condition d'ancien événement n'est spécifiée, tous les anciens événements peuvent être correspondus et fermés. De même, si aucune condition de nouvel événement n'est spécifiée, tous les nouveaux événements peuvent être correspondus et fermés. Les conditions suivantes sont disponibles : **Old event tag name** - spécifiez le nom de l'étiquette de l'ancien événement à faire correspondre. **New event tag name** - spécifiez le nom de l'étiquette du nouvel événement à faire correspondre. **New event host group** - spécifiez le groupe d'hôtes du nouvel événement à faire correspondre. **Event tag pair** - spécifiez le nom de l'étiquette du nouvel événement et le nom de l'étiquette de l'ancien événement à faire correspondre. Dans ce cas, il y aura correspondance si les **valeurs** des étiquettes dans les deux événements correspondent. Les *noms* des étiquettes n'ont pas besoin de correspondre. Cette option est utile pour faire correspondre des valeurs d'exécution, qui peuvent ne pas être connues au moment de la configuration (voir aussi Example). **Old event tag value** - spécifiez le nom et la valeur de l'étiquette de l'ancien événement à faire correspondre, en utilisant les opérateurs suivants : *equals* - a la valeur de l'étiquette de l'ancien événement *does not equal* - n'a pas la valeur de l'étiquette de l'ancien événement *contains* - contient la chaîne dans la valeur de l'étiquette de l'ancien événement *does not contain* - ne contient pas la chaîne dans la valeur de l'étiquette de l'ancien événement **New event tag value** - spécifiez le nom et la valeur de l'étiquette du nouvel événement à faire correspondre, en utilisant les opérateurs suivants : *equals* - a la valeur de l'étiquette du nouvel événement *does not equal* - n'a pas la valeur de l'étiquette du nouvel événement *contains* - contient la chaîne dans la valeur de l'étiquette du nouvel événement *does not contain* - ne contient pas la chaîne dans la valeur de l'étiquette du nouvel événement

Warning:
En raison de la possibilité d'une mauvaise configuration, lorsque des étiquettes d'événement similaires peuvent être créées pour des problèmes **sans rapport**, veuillez examiner les cas décrits ci-dessous !

- Les étiquettes réelles et leurs valeurs ne deviennent visibles que lorsqu'un déclencheur se produit. Si l'expression régulière utilisée est invalide, elle est silencieusement remplacée par une chaîne *UNKNOWN*. Si l'événement de problème initial avec une valeur d'étiquette *UNKNOWN* est manqué, il peut apparaître ensuite des événements OK avec la même valeur d'étiquette *UNKNOWN* qui peuvent fermer des événements de problème qu'ils n'auraient pas dû fermer.
- Si un utilisateur utilise la macro {ITEM.VALUE} sans fonctions de macro comme valeur de l'étiquette, la limite de 255 caractères s'applique. Lorsque les messages de journal sont longs et que les 255 premiers caractères ne sont pas spécifiques, cela peut également entraîner des étiquettes d'événement similaires pour des problèmes sans rapport.

Exemple

Empêchez les événements de problème répétitifs provenant du même port réseau.



Cette règle de corrélation globale associera les problèmes si les valeurs des balises *host* et *port* existent sur le déclencheur et si elles sont identiques dans l'événement d'origine et dans le nouveau.

L'opération fermera les nouveaux événements de problème sur le même port réseau, en ne laissant ouvert que le problème d'origine.

Découverte

Vue d'ensemble La découverte offre des moyens automatiques d'ajouter des hôtes et de configurer la supervision sans configuration manuelle. Zabbix prend en charge plusieurs méthodes de découverte afin de s'adapter à différents scénarios.

Méthodes de découverte

- Découverte du réseau - détecter automatiquement les hôtes et les services disponibles sur votre réseau.
- Enregistrement automatique des agents actifs - permettre aux agents actifs de s'enregistrer automatiquement auprès du serveur.
- Découverte de bas niveau - créer automatiquement des éléments, des déclencheurs, des graphiques et des hôtes pour les entités découvertes, comme les systèmes de fichiers ou les interfaces réseau, à l'aide de prototypes et de règles de découverte imbriquées.

Découverte réseau

Vue d'ensemble

Zabbix offre une fonctionnalité de découverte automatique du réseau qui est efficace et très flexible.

Une fois la découverte du réseau correctement configurée, vous pouvez :

- Accélérer le déploiement de Zabbix.
- Simplifier l'administration.
- Utiliser Zabbix dans des environnements en évolution rapide sans administration excessive.

La découverte du réseau Zabbix est basée sur les informations suivantes :

- Plages d'adresses IP
- Disponibilité des services externes (FTP, SSH, WEB, POP3, IMAP, TCP, etc.)
- Informations reçues de l'agent Zabbix (seul le mode non chiffré est pris en charge)
- Informations reçues de l'agent SNMP

Elle ne fournit PAS la découverte de la topologie réseau.

La découverte du réseau se compose essentiellement de deux phases : la découverte et les actions.

Découverte

Zabbix analyse périodiquement les plages d'adresses IP définies dans les règles de découverte réseau. La fréquence de la vérification est configurable individuellement pour chaque règle.

Chaque règle dispose d'un ensemble de vérifications de service définies à exécuter pour la plage d'adresses IP.

Les règles de découverte sont traitées par le gestionnaire de découverte. Le gestionnaire de découverte crée une tâche par règle avec une liste de tâches (vérifications réseau). Les vérifications réseau sont exécutées en parallèle par les workers de découverte disponibles (leur nombre est configurable dans l'interface pour chaque règle). L'exception concerne les vérifications SNMPv3, qui sont traitées par un seul worker. Seules les vérifications avec la même adresse IP et le même port sont planifiées séquentiellement, car certains périphériques n'acceptent pas les connexions parallèles sur le même port.

La taille de la file d'attente des vérifications réseau est limitée à 2000000, soit environ 4 Go de mémoire. Si la file d'attente est pleine, la règle de découverte sera ignorée et un message d'avertissement sera inscrit dans le journal. Vous pouvez utiliser l'élément interne `zabbix[discovery_queue]` pour surveiller le nombre de vérifications de découverte dans la file d'attente.

Les vérifications de découverte sont traitées indépendamment des autres vérifications. Si certaines vérifications ne trouvent pas de service (ou échouent), les autres vérifications continueront d'être traitées.

Note:
Si une règle de découverte est modifiée pendant son exécution, l'exécution de la découverte en cours sera interrompue.

Chaque vérification d'un service et d'un hôte (IP) effectuée par le module de découverte réseau génère un événement de découverte.

Event	Check of service result
Service Discovered	Le service est 'up' après avoir été 'down' ou lorsqu'il est découvert pour la première fois.
Service Up	Le service est 'up', après avoir déjà été 'up'.
Service Lost	Le service est 'down' après avoir été 'up'.
Service Down	Le service est 'down', après avoir déjà été 'down'.
Host Discovered	Au moins un service d'un hôte est 'up' après que tous les services de cet hôte étaient 'down', ou un service est découvert et appartient à un hôte non enregistré.
Host Up	Au moins un service d'un hôte est 'up', après qu'au moins un service était déjà 'up'.
Host Lost	Tous les services d'un hôte sont 'down' après qu'au moins un était 'up'.
Host Down	Tous les services d'un hôte sont 'down', après qu'ils étaient déjà 'down'.

Actions

Les événements de découverte peuvent servir de base à des actions pertinentes, telles que :

- Envoi de notifications
- Ajout/suppression d'hôtes
- Activation/désactivation d'hôtes
- Ajout d'hôtes à un groupe
- Suppression d'hôtes d'un groupe
- Ajout de tags à un hôte
- Suppression de tags d'un hôte
- Liaison d'un modèle à des hôtes/dissociation d'un modèle d'hôtes
- Exécution de scripts distants

Ces actions peuvent être configurées en fonction du type d'appareil, de l'IP, du statut, du temps de fonctionnement/d'arrêt, etc. Pour tous les détails sur la configuration des actions pour les événements basés sur la découverte réseau, consultez les pages operation et conditions des actions.

Étant donné que les actions de découverte réseau sont basées sur des événements, elles seront déclenchées à la fois lorsqu'un hôte découvert est en ligne et lorsqu'il est hors ligne. Il est fortement recommandé d'ajouter la condition d'action *Discovery status: up* afin d'éviter que des actions telles que *Add host* soient déclenchées lors d'événements *Service Lost*/*Service Down*. Sinon, si un hôte découvert est supprimé manuellement, il générera tout de même des événements *Service Lost*/*Service Down* et sera recréé lors du prochain cycle de découverte.

Note:
La liaison de modèles à un hôte découvert échouera globalement si l'un des modèles pouvant être liés possède une entité unique (par exemple, une clé d'élément) identique à une entité unique (par exemple, une clé d'élément) existant déjà sur l'hôte ou sur un autre des modèles pouvant être liés.

Création d'un hôte

Un hôte est ajouté si l'opération *Ajouter un hôte* est sélectionnée. Un hôte est également ajouté, même si l'opération *Ajouter un hôte* est absente, si vous sélectionnez des opérations entraînant des actions sur un hôte. Ces opérations sont les suivantes :

- Activer l'hôte.
- Désactiver l'hôte.
- Ajouter l'hôte à un groupe d'hôtes.
- Lier le modèle à un hôte.

Les hôtes créés sont ajoutés au groupe *Hôtes découverts* (par défaut, configurable dans *Administration* > *Général* > *Autre*). Si vous souhaitez que les hôtes soient ajoutés à un autre groupe, ajoutez une opération *Supprimer des groupes d'hôtes* (en spécifiant "Hôtes découverts") et ajoutez également une opération *Ajouter aux groupes d'hôtes* (en spécifiant un autre groupe d'hôtes), car un hôte doit appartenir à un groupe d'hôtes.

L'adresse IP de l'appareil découvert, ainsi que la source de découverte (serveur Zabbix, proxy Zabbix ou groupe de proxy) et le type d'interface, sont utilisés comme critère pour rechercher un hôte dans le système. Si un hôte avec la même adresse IP, le même type d'interface et la même source de découverte existe déjà, cet hôte sera la cible des opérations. Si la source de découverte diffère, l'entité découverte est traitée comme un hôte différent et un nouvel hôte peut être créé.

Si l'adresse IP de l'hôte découvert est modifiée ou si l'interface est supprimée, un nouvel hôte sera créé lors de la découverte suivante.

Nommage des hôtes

Lors de l'ajout d'hôtes, le nom d'hôte résulte d'une résolution DNS inverse ou de l'adresse IP si la résolution inverse échoue. La résolution est effectuée depuis le serveur Zabbix ou le proxy Zabbix, selon celui qui réalise la découverte. Si la résolution échoue sur le proxy, elle n'est pas retentée sur le serveur. Si un hôte portant déjà ce nom existe, le nom de l'hôte suivant se verra ajouter **_2**, puis **_3**, et ainsi de suite.

Il est également possible de remplacer la résolution DNS/IP et d'utiliser à la place la valeur d'un élément comme nom d'hôte, par exemple :

- Vous pouvez découvrir plusieurs serveurs exécutant l'agent Zabbix à l'aide d'un élément d'agent Zabbix pour la découverte et leur attribuer automatiquement des noms appropriés, en fonction de la valeur de chaîne renvoyée par cet élément
- Vous pouvez découvrir plusieurs équipements réseau SNMP à l'aide d'un élément d'agent SNMP pour la découverte et leur attribuer automatiquement des noms appropriés, en fonction de la valeur de chaîne renvoyée par cet élément

Si le nom d'hôte a été défini à l'aide de la valeur d'un élément, il n'est pas mis à jour lors des vérifications de découverte suivantes. S'il n'est pas possible de définir le nom d'hôte à l'aide de la valeur d'un élément, la valeur par défaut (nom DNS) est utilisée.

Si un hôte existe déjà avec l'adresse IP découverte et que la source de découverte (serveur Zabbix, proxy ou groupe de proxys) n'a pas changé, aucun nouvel hôte n'est créé. Si la source de découverte diffère, l'entité découverte est traitée comme distincte et un nouvel hôte peut être créé. Cependant, si l'action de découverte contient des opérations (lier un modèle, ajouter à un groupe d'hôtes, etc.), elles sont exécutées sur l'hôte existant qui correspond à l'adresse IP, au type d'interface et à la source de découverte.

Suppression d'hôte

Les hôtes découverts par une règle de découverte réseau sont supprimés automatiquement de *Monitoring* > *Discovery* si une entité découverte ne se trouve plus dans la plage d'adresses IP de la règle. Les hôtes sont supprimés immédiatement.

Création d'interface lors de l'ajout d'hôtes

Lorsque des hôtes sont ajoutés à la suite d'une découverte réseau, des interfaces leur sont créées selon les règles suivantes :

- Les services détectés - par exemple, si une vérification SNMP a réussi, une interface SNMP sera créée.
- Si un hôte a répondu à la fois aux requêtes de l'agent Zabbix et de SNMP, les deux types d'interfaces seront créés.
- Si les critères d'unicité sont des données renvoyées par l'agent Zabbix ou SNMP, la première interface trouvée pour un hôte sera créée comme interface par défaut. Les autres adresses IP seront ignorées, sauf si l'interface par défaut tombe en panne. Dans le cas où l'interface par défaut est en panne, la plus petite des autres adresses IP sera utilisée en remplacement de l'interface par défaut. Les conditions de l'action (telles que Host IP) n'ont pas d'incidence sur l'ajout d'interfaces. *Note* que cela fonctionnera si toutes les interfaces sont découvertes par la même règle de découverte. Si une règle de découverte différente découvre une autre interface du même hôte, un hôte supplémentaire sera ajouté.

- Si un hôte a répondu uniquement aux vérifications de l'agent, il sera créé avec une seule interface d'agent. S'il commence plus tard à répondre à SNMP, des interfaces SNMP supplémentaires seront ajoutées.
- Si 3 hôtes distincts ont d'abord été créés, après avoir été découverts selon les critères d'unicité "IP", puis que la règle de découverte est modifiée de sorte que les hôtes A, B et C aient un résultat de critères d'unicité identique, B et C sont créés comme interfaces supplémentaires pour A, le premier hôte. Les hôtes individuels B et C restent inchangés. Dans *Monitoring > Discovery*, les interfaces ajoutées seront affichées dans la colonne "Discovered device", en police noire et avec un retrait, mais la colonne "Monitored host" n'affichera que A, le premier hôte créé. "Uptime/Downtime" n'est pas mesuré pour les IP considérées comme des interfaces supplémentaires.

Modification du paramètre du proxy

Les hôtes découverts par différents proxies ne sont pas toujours traités comme des hôtes différents. Les vérifications de découverte et d'unicité dépendent de la structure du groupe de proxies : lorsqu'un proxy exécute une règle de découverte et crée un hôte, cet hôte est ajouté au groupe de proxies parent du proxy, et non attribué au proxy lui-même. Lorsque Zabbix évalue l'unicité des adresses IP pendant la découverte, il vérifie les hôtes surveillés par le groupe de proxies parent. Les hôtes surveillés par des proxies individuels au sein de ce groupe, y compris le proxy qui a exécuté la découverte, sont ignorés pour la vérification d'unicité, ce qui peut entraîner des hôtes en double si plusieurs proxies surveillent des sous-réseaux qui se chevauchent.

Bien que ce comportement permette à la découverte de fonctionner sur des plages d'adresses IP qui se chevauchent et qui sont utilisées par différents sous-réseaux, le changement du proxy attribué à un sous-réseau déjà surveillé est plus complexe, car les modifications de proxy doivent être appliquées de manière cohérente aux hôtes découverts et à l'appartenance au groupe de proxies parent afin d'éviter les doublons.

Par exemple, les étapes pour remplacer le proxy dans une règle de découverte :

1. Désactiver la règle de découverte.
2. Synchroniser la configuration du proxy.
3. Remplacer le proxy dans la règle de découverte.
4. Remplacer le proxy pour tous les hôtes découverts par cette règle (veillez à mettre à jour les hôtes du groupe de proxies parent ainsi que tous les hôtes surveillés par des proxies individuels dans ce groupe afin d'éviter les doublons).
5. Activer la règle de découverte.

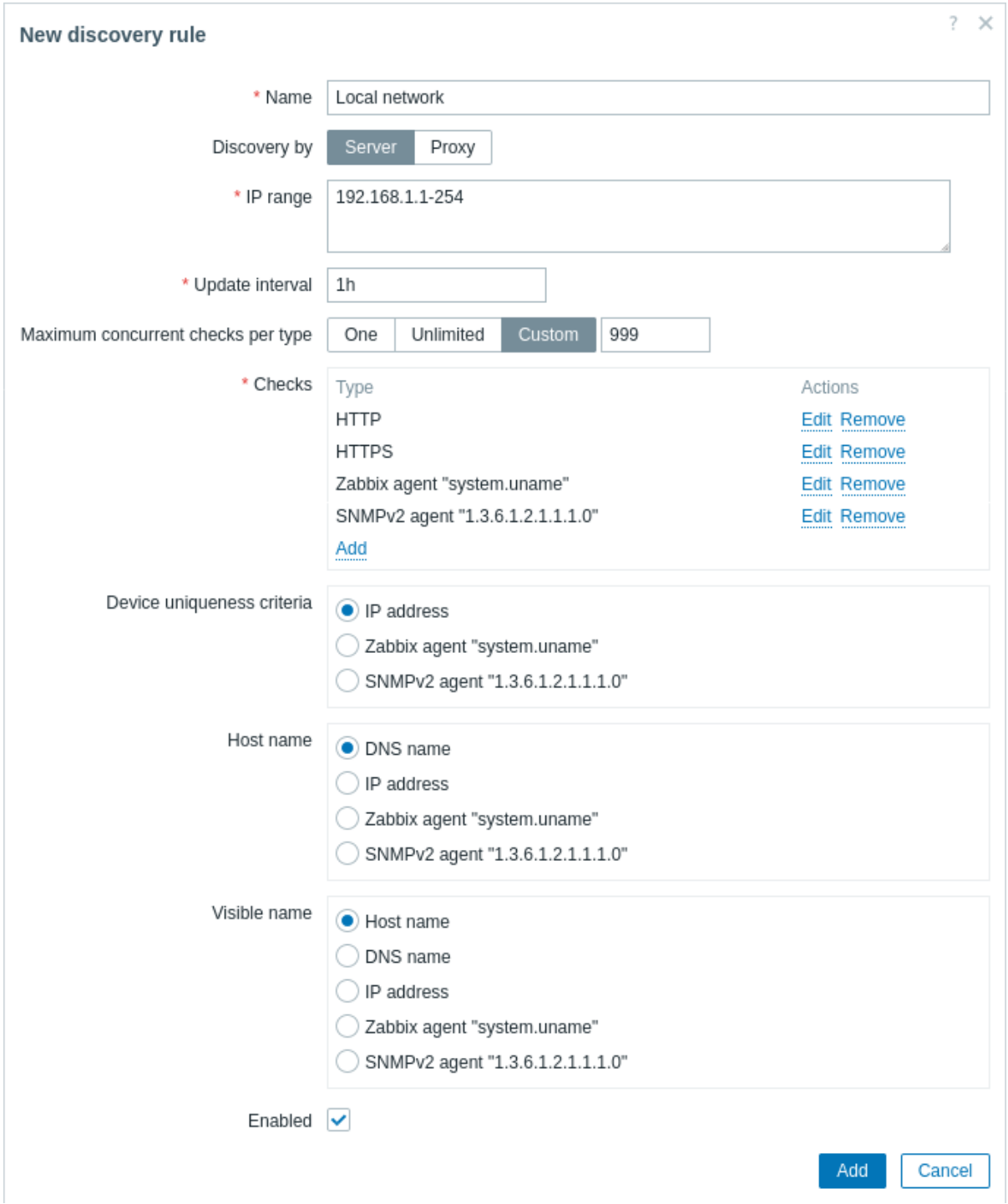
Configuration d'une règle de découverte réseau

Vue d'ensemble

Pour configurer une règle de découverte réseau utilisée par Zabbix pour découvrir des hôtes et des services :

1. Accédez à *Collecte de données > Découverte*.
2. Cliquez sur *Créer une règle de découverte* (ou sur le nom de la règle pour modifier une règle existante).
3. Modifiez les attributs de la règle de découverte.

Attributs de la règle



Tous les champs obligatoires sont marqués d'un astérisque rouge.

Parameter	Description
Name	Nom unique de la règle. Par exemple, "Réseau local".
Discovery by	La découverte est effectuée par: **Server** - par le serveur Zabbix **Proxy** - par le proxy Zabbix (sélectionné dans le champ du nom du proxy)
IP range	Plage d'adresses IP à découvrir. Elle peut avoir les formats suivants: IP unique: 192.168.1.33 Plage d'adresses IP: 192.168.1-10.1-255. La plage est limitée par le nombre total d'adresses couvertes (moins de 64K). Masque IP: 192.168.4.0/24 masques IP pris en charge: /16 - /30 pour les adresses IPv4 /112 - /128 pour les adresses IPv6 Liste: 192.168.1.1-255, 192.168.2.1-100, 192.168.2.200, 192.168.4.0/24 Ce champ prend en charge les espaces, les tabulations et plusieurs lignes.

Parameter	Description
Update interval	Ce paramètre définit la fréquence à laquelle Zabbix exécutera la règle. L'intervalle est mesuré après la fin de l'exécution de l'instance de découverte précédente, de sorte qu'il n'y a pas de chevauchement. Les suffixes de temps sont pris en charge, par exemple 30s, 1m, 2h, 1d. Les macros utilisateur sont prises en charge. *Notez* que si une macro utilisateur est utilisée et que sa valeur est modifiée (par exemple, 1w → 1h), la prochaine vérification sera exécutée selon la valeur précédente (très loin dans le futur avec les valeurs de l'exemple).
Maximum concurrent checks per type	Définissez le nombre maximal de threads de découverte (workers) par vérification de service pour le traitement parallèle des vérifications de découverte: **One** - un thread **Unlimited** - nombre illimité de threads (mais pas plus que dans le paramètre StartDiscoverers) **Custom** - définissez un nombre personnalisé de threads (0-999). Les vérifications de service asynchrones SNMPv1 et SNMPv2 sont traitées simultanément par les workers disponibles. Toutes les règles de découverte avec des vérifications de service asynchrones SNMPv3 sont traitées par un seul worker en raison des particularités de l'implémentation libsnmp; par conséquent, augmenter le nombre de workers n'augmentera pas la vitesse de découverte.
Checks	Zabbix utilisera cette liste de vérifications pour la découverte. Cliquez sur Add pour configurer une nouvelle vérification dans une fenêtre contextuelle. Vérifications prises en charge: SSH, LDAP, SMTP, FTP, HTTP, HTTPS, POP, NNTP, IMAP, TCP, Telnet, Zabbix agent, SNMPv1 agent, SNMPv2 agent, SNMPv3 agent, ICMP ping. Une découverte basée sur un protocole utilise la fonctionnalité **net.tcp.service[]** pour tester chaque hôte, sauf pour SNMP qui interroge un OID SNMP. Zabbix agent est testé en interrogeant un élément en mode non chiffré. Veuillez consulter éléments agent pour plus de détails. Le paramètre 'Ports' peut être l'un des suivants: Port unique: 22 Plage de ports: 22-45 Liste: 22-45,55,60-70 Depuis Zabbix 7.0, toutes les vérifications de service sont effectuées de manière asynchrone, à l'exception des vérifications LDAP. Depuis Zabbix 7.0, la vérification HTTP/HTTPS est effectuée via libcurl. Si Zabbix serveur/proxy est compilé sans libcurl, alors les vérifications HTTP fonctionneront comme dans les versions précédentes (c'est-à-dire comme des vérifications TCP), mais les vérifications HTTPS ne fonctionneront pas.
Device uniqueness criteria	Les critères d'unicité peuvent être: **IP address** - ne pas traiter plusieurs appareils à IP unique. Si un appareil avec la même IP existe déjà, il sera considéré comme déjà découvert et un nouvel hôte ne sera pas ajouté. **<discovery check>** - vérification de Zabbix agent ou de SNMP agent. Notez que les critères d'unicité utilisés pendant la découverte ne sont pas les mêmes que l'identification de l'hôte dans le système lors de l'exécution des actions. Les critères d'unicité pendant la découverte définissent si deux appareils découverts ou plus sont identiques (ou différents), tandis que seule l'adresse IP est le critère d'identification de l'hôte dans Zabbix (voir Création d'hôte).
Host name	Définissez le nom technique de l'hôte créé en utilisant: **DNS name** - nom DNS (par défaut) **IP address** - adresse IP **<discovery check>** - valeur chaîne reçue de la vérification de découverte (par exemple, vérification Zabbix agent, SNMP agent) Voir aussi: Nomination des hôtes.
Visible name	Définissez le nom visible de l'hôte créé en utilisant: **Host name** - nom technique de l'hôte (par défaut) **DNS name** - nom DNS **IP address** - adresse IP **<discovery check>** - valeur chaîne reçue de la vérification de découverte (par exemple, vérification Zabbix agent, SNMP agent) Voir aussi: Nomination des hôtes.
Enabled	Lorsque la case est cochée, la règle est active et sera exécutée par le serveur Zabbix. Si elle n'est pas cochée, la règle n'est pas active. Elle ne sera pas exécutée.

Dépassement de la limite des descripteurs de fichiers

En cas de grand nombre de vérifications simultanées, il est possible d'atteindre la limite des descripteurs de fichiers pour le gestionnaire de découverte.

Le nombre de descripteurs de fichiers requis pour la détection correspond au nombre de workers de découverte * 1000. Par défaut, il y a 5 workers de découverte, tandis que la limite souple du système est d'environ 1024.

Si cette limite est approchée, Zabbix réduira le nombre par défaut de vérifications simultanées par type pour chaque worker et écrira un avertissement dans le fichier journal. Cependant, si l'utilisateur a défini une valeur plus élevée pour *Maximum concurrent checks per type* que la valeur calculée par Zabbix, Zabbix utilisera la valeur définie par l'utilisateur pour un worker.

Scénario concret

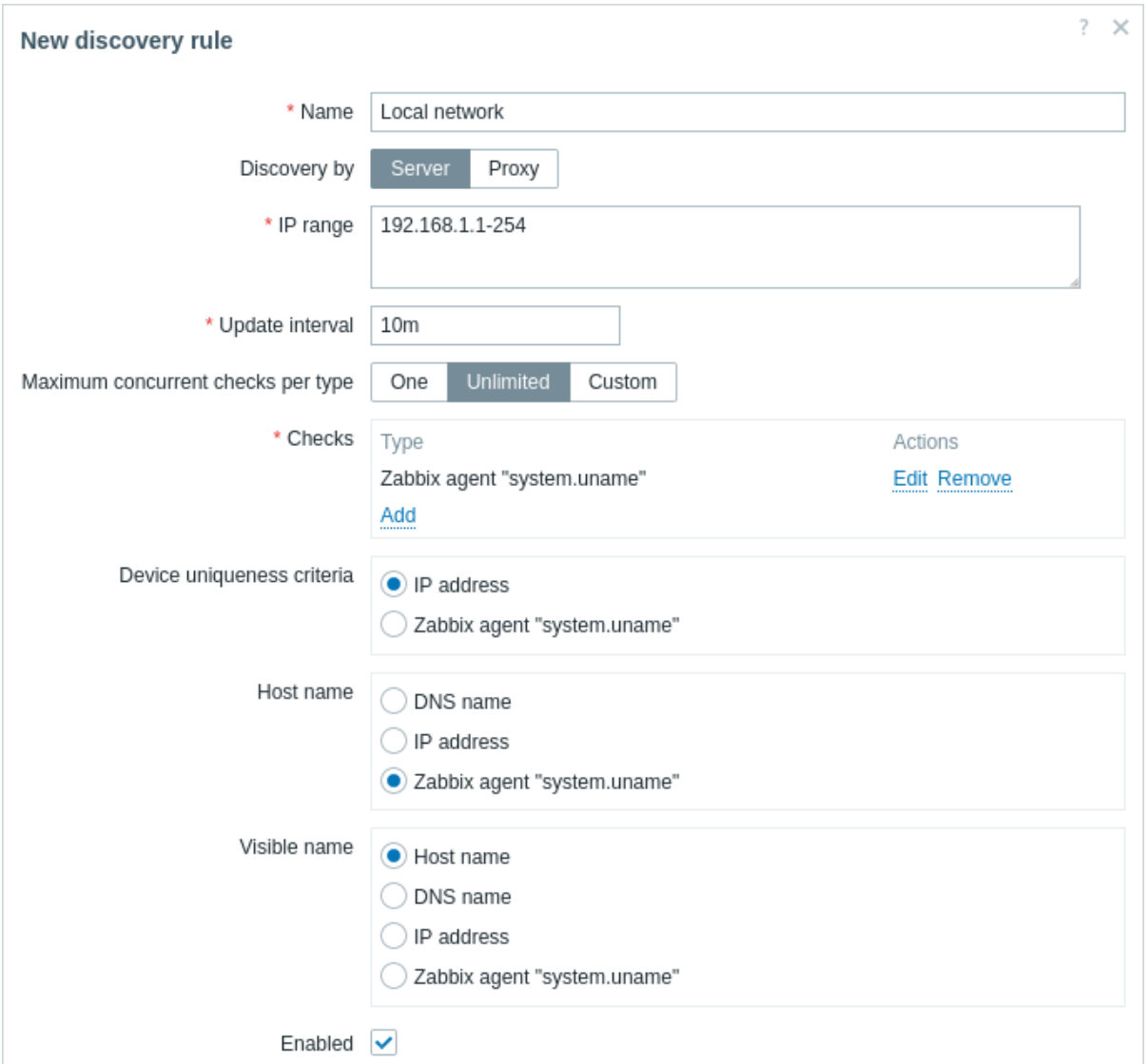
Dans cet exemple, nous souhaiterions configurer la détection de réseau pour le réseau local ayant une plage d'adresses IP : 192.168.1.1-192.168.1.254.

Dans notre scénario, nous voulons :

- découvrir les hôtes sur lesquels l'agent Zabbix est en cours d'exécution
- lancer la découverte toutes les 10 minutes
- ajouter un hôte à la surveillance si le temps de disponibilité de l'hôte est supérieur à 1 heure
- supprimer des hôtes si le temps d'inactivité de l'hôte est supérieur à 24 heures
- ajouter des hôtes Linux au groupe "Linux servers"
- ajouter des hôtes Windows au groupe "Windows servers"
- utiliser *Template OS Linux* pour les hôtes Linux
- utiliser *Template OS Windows* pour les hôtes Windows

Étape 1

Définition d'une règle de découverte réseau pour notre plage d'adresses IP.



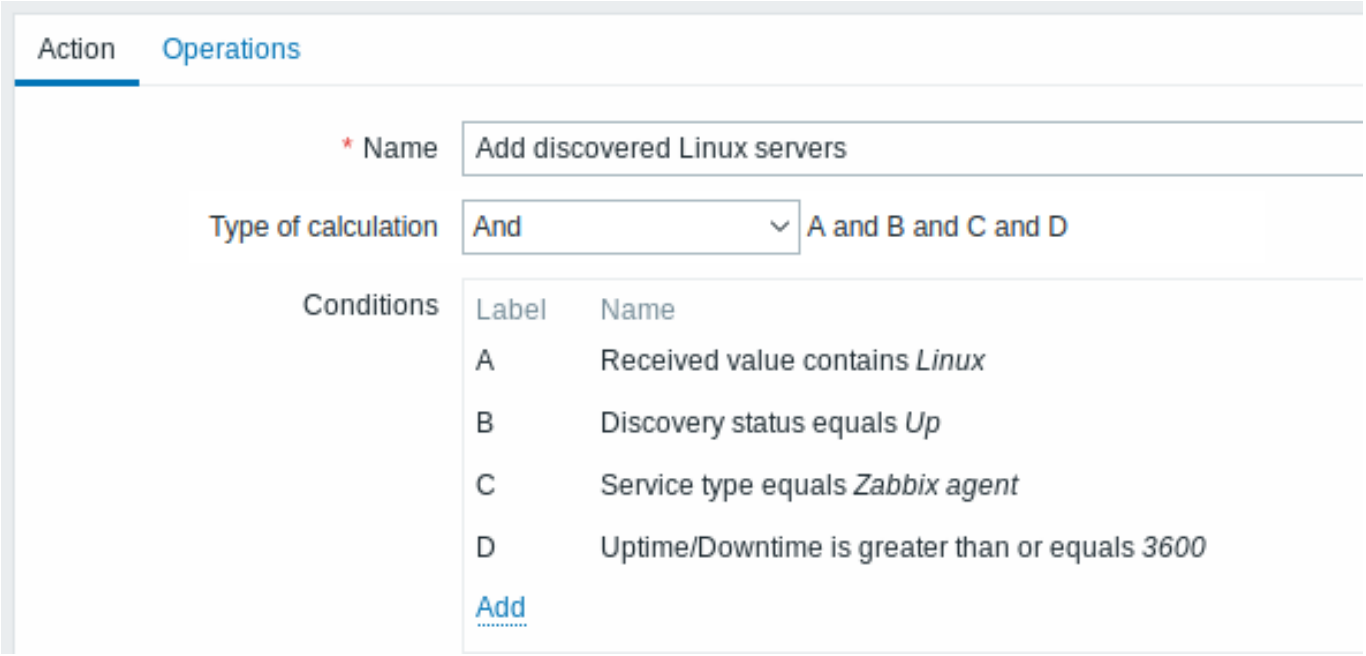
Zabbix tentera de découvrir les hôtes dans la plage d'adresses IP 192.168.1.1-192.168.1.254 en se connectant aux agents Zabbix et en récupérant la valeur de la clé **system.uname**. La valeur reçue de l'agent peut être utilisée pour nommer les hôtes et également pour appliquer différentes actions selon les différents systèmes d'exploitation. Par exemple, lier les serveurs Windows au modèle *Windows*, les serveurs Linux au modèle *Linux*.

La règle sera exécutée toutes les 10 minutes.

Lorsque cette règle est ajoutée, Zabbix démarrera automatiquement la découverte et la génération des événements basés sur la découverte pour un traitement ultérieur.

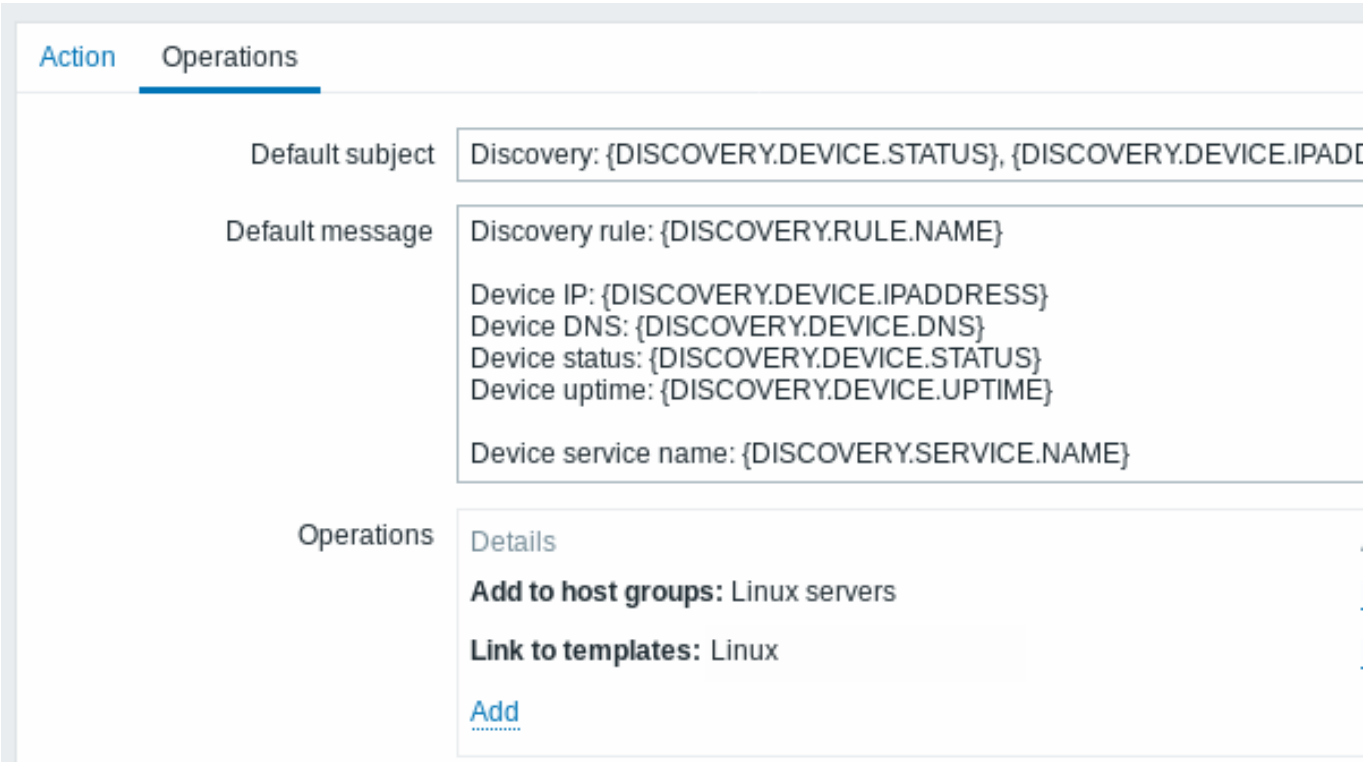
Étape 2

Définir une action pour ajouter les serveurs Linux découverts au groupe/modèle correspondant.



L'action sera activée si :

- le service "agent Zabbix" est "démarré"
- la valeur de system.uname (la clé de l'agent Zabbix que nous avons utilisée dans la définition de la règle) contient "Linux"
- La disponibilité est de 1 heure (3600 secondes) ou plus

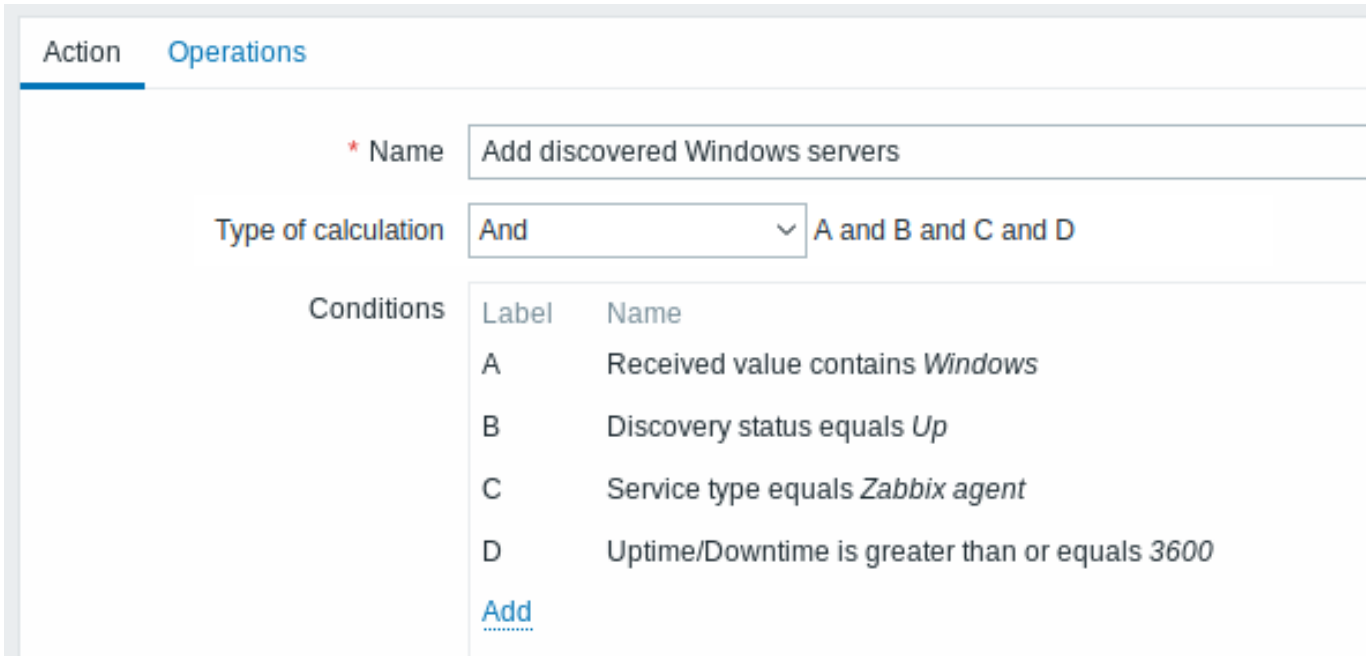


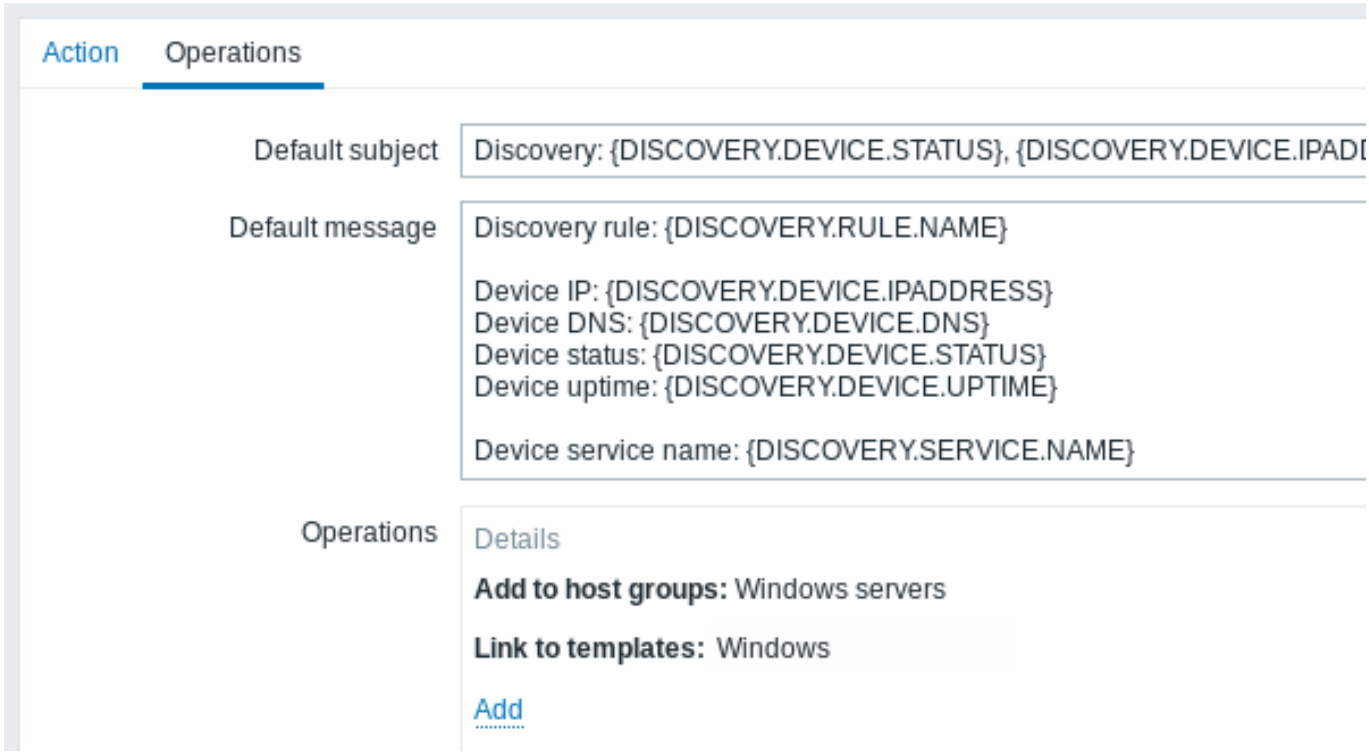
L'action exécutera les opérations suivantes :

- ajoute l'hôte découvert au groupe "serveurs Linux" (et ajoute également l'hôte s'il n'a pas été ajouté précédemment)
- lie l'hôte au "Template OS Linux". Zabbix commencera automatiquement à surveiller l'hôte à l'aide d'éléments et de déclencheurs de "Template OS Linux".

Étape 3

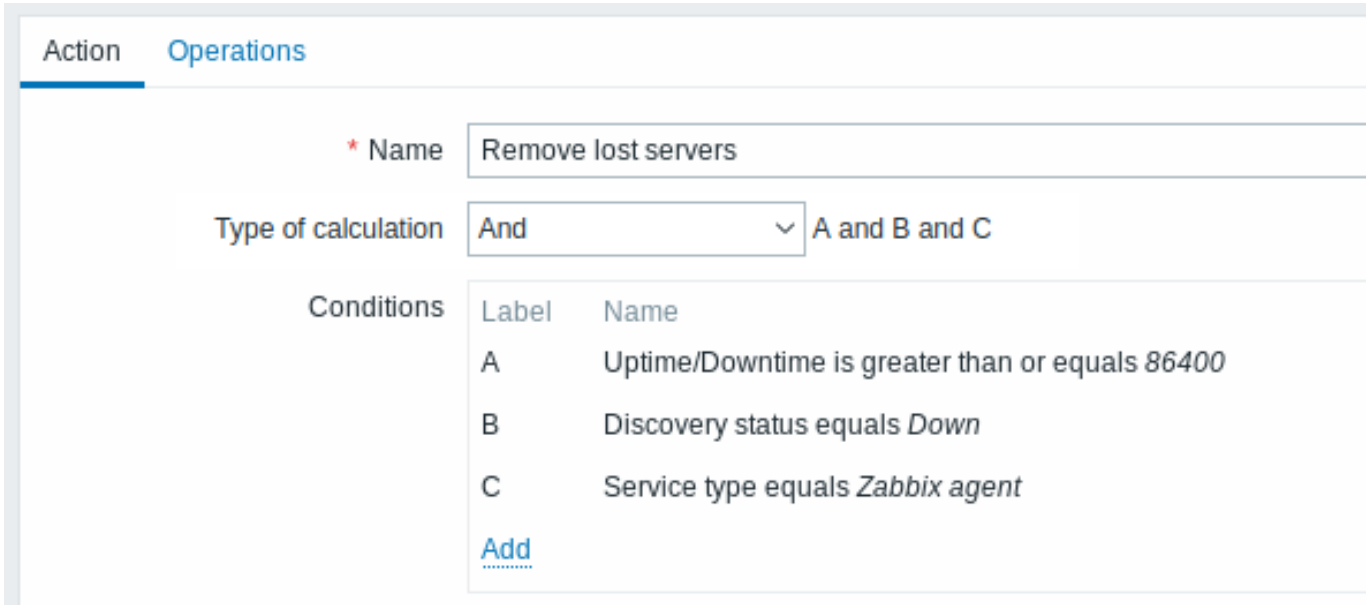
Définir une action pour ajouter les serveurs Windows découverts au groupe/modèle correspondant.

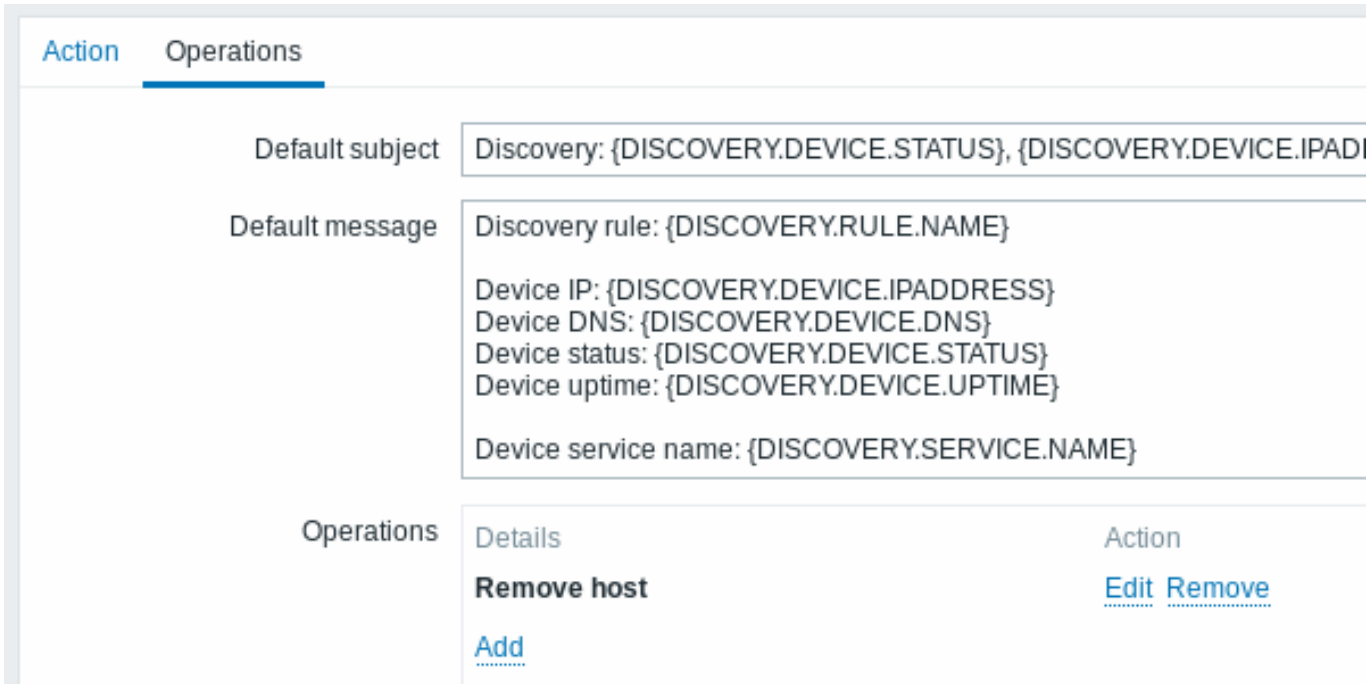




Étape 4

Définir une action pour supprimer les serveurs perdus.





Un serveur sera supprimé si le service ”Agent Zabbix” est ’arrêté’ pendant plus de 24 heures (86400 secondes).

Enregistrement automatique de l’agent actif

Vue d’ensemble

Il est possible d’autoriser l’autoregistration de l’agent Zabbix actif, après quoi le serveur peut commencer à les surveiller. De cette manière, de nouveaux hôtes peuvent être ajoutés à la surveillance sans être configurés manuellement sur le serveur.

L’autoregistration peut se produire lorsqu’un agent actif jusque-là inconnu demande des vérifications.

Cette fonctionnalité peut s’avérer très utile pour la surveillance automatique de nouveaux nœuds cloud. Dès qu’un nouveau nœud est présent dans Zabbix Cloud, la collecte des données de performance et de disponibilité de l’hôte démarre automatiquement.

L’autoregistration de l’agent actif prend également en charge la surveillance des hôtes ajoutés avec des vérifications passives. Lorsque l’agent actif demande des vérifications, à condition que les paramètres de configuration `ListenIP` ou `ListenPort` soient définis dans le fichier de configuration, ceux-ci sont transmis au serveur. Si plusieurs adresses IP sont spécifiées, la première est envoyée au serveur.

Le serveur, lors de l’ajout du nouvel hôte autoregistré, utilise l’adresse IP et le port reçus pour configurer l’agent. Si aucune valeur d’adresse IP n’est reçue, celle utilisée pour la connexion entrante est utilisée. Si aucune valeur de port n’est reçue, 10050 est utilisé.

Il est possible de spécifier que l'hôte doit être autoregistré avec un nom DNS comme interface d'agent par défaut.

L'autoregistration est relancée :

- si les informations de métadonnées de l'hôte changent :
  - en raison d'une modification de HostMetadata et du redémarrage de l'agent
  - en raison d'une modification de la valeur renvoyée par HostMetadataItem
- pour les hôtes créés manuellement avec des métadonnées manquantes
- si un hôte est modifié manuellement pour être surveillé par un autre proxy Zabbix
- si l'autoregistration pour le même hôte provient d'un nouveau proxy Zabbix

Le heartbeat d'autoregistration de l'agent actif pour Zabbix serveur et Zabbix proxy est de 120 secondes. Ainsi, dans le cas où un hôte découvert est supprimé, l'autoregistration sera relancée dans 120 secondes.

Configuration

Spécifier le serveur

Assurez-vous que le serveur Zabbix est indiqué dans le fichier de configuration de l'agent (`zabbix_agentd.conf`) :

```
ServerActive=10.0.0.1
```

Sauf si vous définissez spécifiquement un *Hostname* dans `zabbix_agentd.conf`, le nom d'hôte système de l'emplacement de l'agent sera utilisé par le serveur pour nommer le hôte. Sous Linux, le nom d'hôte système peut être obtenu en exécutant la commande `hostname`.

Si *Hostname* est défini dans la configuration de l'agent Zabbix sous forme de liste d'hôtes séparés par des virgules, des hôtes seront créés pour tous les noms d'hôte indiqués.

Redémarrez l'agent après toute modification du fichier de configuration.

Action pour l'enregistrement automatique des agents actifs

Lorsque le serveur reçoit une demande d'enregistrement automatique d'un agent, il appelle une action. Une action dont la source d'événement est « Enregistrement automatique » doit être configurée pour l'enregistrement automatique des agents.

> **Note:**
> La configuration de la découverte réseau n'est pas requise pour que les agents actifs s'enregistrent automatiquement.

Dans l'interface Zabbix :

1. Accédez à *Alertes* > *Actions*.
2. Sélectionnez *Actions d'enregistrement automatique*.
3. Cliquez sur *Créer une action*.
4. Dans l'onglet *Action*, donnez un nom à votre action.
5. Vous pouvez éventuellement spécifier des conditions. Vous pouvez effectuer une recherche par sous-chaîne ou par expression régulière dans les conditions portant sur le nom du hôte ou les métadonnées du hôte. Si vous comptez utiliser le type de condition *Métadonnées du hôte*, consultez Utilisation des métadonnées du hôte.
6. Dans l'onglet *Opérations*, ajoutez les opérations pertinentes, telles que *Ajouter un hôte*, *Ajouter au groupe de hôtes* (par exemple, *Hôtes découverts*), *Lier des modèles*, etc.

> **Note:**
> Si les hôtes qui vont s'enregistrer automatiquement sont susceptibles d'être pris en charge uniquement pour la supervision active (par exemple, des hôtes protégés par un pare-feu qui les empêche d'être accessibles depuis votre serveur Zabbix), vous pouvez créer un modèle spécifique, tel que *Template_Linux-active*, auquel les lier.

Les hôtes créés sont ajoutés au groupe *Hôtes découverts* (par défaut, ce paramètre est configurable dans *Administration* > *Général* > *Autres*). Si vous souhaitez que les hôtes soient ajoutés à un autre groupe, ajoutez une opération *Supprimer du groupe de hôtes* (en spécifiant « Hôtes découverts »), puis ajoutez également une opération *Ajouter au groupe de hôtes* (en spécifiant un autre groupe de hôtes), car un hôte doit appartenir à un groupe de hôtes.

Autoregistration sécurisée

Une méthode sécurisée d'autoregistration est possible en configurant une authentification basée sur une clé PSK avec des connexions chiffrées.

Le niveau de chiffrement est configuré globalement dans *Administration* > *Général* > *Autoregistration*. Il est possible de sélectionner l'absence de chiffrement, le chiffrement TLS avec authentification PSK, ou les deux (afin que certains hôtes puissent s'enregistrer sans chiffrement tandis que d'autres utilisent une connexion chiffrée).

L'authentification par PSK est vérifiée par le serveur Zabbix avant l'ajout d'un hôte. Si elle réussit, l'hôte est ajouté et *Connexions depuis/vers l'hôte* sont définies sur « PSK » uniquement, avec une identité et une clé prépartagée identiques à celles du paramètre global d'autoregistration.

Attention:
Pour garantir la sécurité de l'autoregistration dans les installations utilisant des proxy, le chiffrement entre le serveur Zabbix et le proxy doit être activé.

Utiliser le DNS comme interface par défaut

HostInterface et HostInterfaceItem paramètres de configuration vous permettent de spécifier une valeur personnalisée pour l'interface de l'hôte lors de l'enregistrement automatique.

Plus précisément, ils sont utiles si l'hôte doit être enregistré automatiquement avec un nom DNS comme interface agent par défaut plutôt qu'avec son adresse IP. Dans ce cas, le nom DNS doit être spécifié ou renvoyé comme valeur de l'un des paramètres HostInterface ou HostInterfaceItem. Si la valeur de l'un de ces paramètres change, par exemple d'une adresse IP vers un nom DNS ou inversement, l'interface par défaut de l'hôte enregistré automatiquement est mise à jour en conséquence. Cette mise à jour est appliquée à l'hôte existant, sans en créer un nouveau. Pour envoyer la nouvelle valeur, l'agent doit être redémarré afin de relancer le processus d'enregistrement automatique.

Note:
Si les paramètres HostInterface ou HostInterfaceItem ne sont pas configurés, le paramètre listen_dns est utilisé à la place. Cette valeur est déterminée en effectuant une recherche DNS inversée de l'adresse IP de l'agent. Si le DNS inversé n'est pas correctement configuré ou renvoie un nom non valide, cela peut entraîner un enregistrement automatique incorrect ou échouer en raison d'une valeur d'interface non valide.

Utilisation des métadonnées de l'hôte

Lorsque l'agent envoie une demande d'enregistrement automatique au serveur, il envoie son nom d'hôte. Dans certains cas (par exemple, pour les nœuds cloud Amazon), un nom d'hôte ne suffit pas au serveur Zabbix pour différencier les hôtes découverts. Les métadonnées de l'hôte peuvent être utilisées facultativement pour envoyer d'autres informations d'un agent au serveur.

Les métadonnées de l'hôte sont configurées dans le fichier de configuration de l'agent (`zabbix_agentd.conf`). Il existe 2 façons de spécifier les métadonnées de l'hôte dans le fichier de configuration :

```
HostMetadata
HostMetadataItem
```

Consultez la description des options dans le lien ci-dessus.

Le paramètre HostMetadataItem peut renvoyer jusqu'à 65535 points de code UTF-8. Une valeur plus longue sera tronquée.

Notez que sur MySQL, la longueur maximale effective en caractères sera inférieure si la valeur renvoyée contient des caractères multioctets. Par exemple, une valeur contenant uniquement des caractères codés sur 3 octets sera limitée à 21844 caractères au total, tandis qu'une valeur contenant uniquement des caractères codés sur 4 octets sera limitée à 16383 symboles.

Attention:
Une tentative d'enregistrement automatique a lieu chaque fois qu'un agent actif envoie une demande d'actualisation des contrôles actifs au serveur. Le délai entre les demandes est spécifié dans le paramètre RefreshActiveChecks de l'agent. La première demande est envoyée immédiatement après le redémarrage de l'agent.

Exemples

Enregistrement automatique par OS à l'aide de HostMetadata

Supposons que vous souhaitiez que les hôtes soient enregistrés automatiquement par le serveur Zabbix. Vous disposez d'agents Zabbix actifs (voir la section « Configuration » ci-dessus) sur votre réseau. Il y a des hôtes Windows et des hôtes Linux sur votre réseau, et vous avez les modèles « Linux by Zabbix agent » et « Windows by Zabbix agent » disponibles dans votre interface Zabbix. Ainsi, lors de l'enregistrement de l'hôte, vous souhaitez que le modèle Linux/Windows approprié soit appliqué à l'hôte en cours d'enregistrement. Par défaut, seul le nom d'hôte est envoyé au serveur lors de l'enregistrement automatique, ce qui peut ne pas être suffisant. Afin de vous assurer que le modèle approprié est appliqué à l'hôte, vous devez utiliser les métadonnées de l'hôte.

Configuration du frontend

La première chose à faire est de configurer le frontend. Créez 2 actions. La première action :

- Nom : Autorenregistrement de l'hôte Linux
- Conditions : Les métadonnées de l'hôte contiennent *Linux*

- Opérations : Lier les modèles : Linux by Zabbix agent

> **Note:**
> Vous pouvez ignorer une opération « Add host » dans ce cas. Lier un modèle à un hôte nécessite d'abord l'ajout de l'hôte ; le serveur le fera donc automatiquement.

La deuxième action :

- Nom : Autorenregistrement de l'hôte Windows
- Conditions : Les métadonnées de l'hôte contiennent *Windows*
- Opérations : Lier les modèles : Windows by Zabbix agent

Configuration de l'agent

Vous devez maintenant configurer les agents. Ajoutez la ligne suivante aux fichiers de configuration de l'agent :

```
HostMetadataItem=system.uname
```

De cette façon, vous vous assurez que les métadonnées de l'hôte contiendront « Linux » ou « Windows » selon l'hôte sur lequel un agent s'exécute. Voici un exemple de métadonnées d'hôte dans ce cas :

```
Linux: Linux server3 3.2.0-4-686-pae #1 SMP Debian 3.2.41-2 i686 GNU/Linux
Windows: Windows WIN-0PXGGSTYNHO 6.0.6001 Windows Server 2008 Service Pack 1 Intel IA-32
```

N'oubliez pas de redémarrer l'agent après avoir apporté des modifications au fichier de configuration.

Utilisation de HostMetadata pour contrôler l'autorégistration et les modèles

Étape 1 - Protéger l'enregistrement avec HostMetadata

Utilisez les métadonnées d'hôte pour offrir une protection de base contre l'enregistrement d'hôtes indésirables.

Configuration du frontend

Créez une action dans le frontend en utilisant un code secret difficile à deviner afin d'empêcher les hôtes indésirables :

- Nom : Action d'autorégistration Linux
- Conditions :
  - Type de calcul : AND
  - Condition (A) : Les métadonnées de l'hôte contiennent //Linux//
  - Condition (B) : Les métadonnées de l'hôte contiennent //21df83bf21bf0be663090bb8d4128558ab9b95fba66a6dbf834f8b91ae5e0
- Opérations :
  - Envoyer un message aux utilisateurs : Admin via tous les médias
  - Ajouter aux groupes d'hôtes : Serveurs Linux
  - Lier les modèles : Linux by Zabbix agent

Veuillez noter que cette méthode seule n'offre pas une protection forte, car les données sont transmises en texte brut. Un rechargement du cache de configuration est nécessaire pour que les modifications prennent effet immédiatement.

Configuration de l'agent

Ajoutez la ligne suivante au fichier de configuration de l'agent :

```
HostMetadata=Linux    21df83bf21bf0be663090bb8d4128558ab9b95fba66a6dbf834f8b91ae5e08ae
```

où « Linux » est une plateforme, et le reste de la chaîne est le texte secret difficile à deviner.

N'oubliez pas de redémarrer l'agent après avoir apporté des modifications au fichier de configuration.

Étape 2 - Ajouter un modèle à l'hôte enregistré

Il est possible d'ajouter des modèles supplémentaires à un hôte déjà enregistré. Dans ce cas, le modèle MySQL by Zabbix agent sera lié uniquement aux hôtes dont le HostMetadata contient le jeton `MySQL`.

Configuration du frontend

Mettez à jour l'action dans le frontend :

- Nom : Action d'autorégistration Linux
- Conditions :
  - Type de calcul : AND
  - Condition (A) : Les métadonnées de l'hôte contiennent Linux
  - Condition (B) : Les métadonnées de l'hôte contiennent 21df83bf21bf0be663090bb8d4128558ab9b95fba66a6dbf834f8b91ae5e08
  - Condition (C) : Les métadonnées de l'hôte contiennent MySQL

- Opérations :
  - Envoyer un message aux utilisateurs : Admin via tous les médias
  - Ajouter aux groupes d'hôtes : Serveurs Linux
  - Lier les modèles : Linux by Zabbix agent
  - Lier les modèles : MySQL by Zabbix Agent

Configuration de l'agent

Mettez à jour la ligne suivante dans le fichier de configuration de l'agent :

```
HostMetadata=MySQL on Linux 21df83bf21bf0be663090bb8d4128558ab9b95fba66a6dbf834f8b91ae5e08ae
```

N'oubliez pas de redémarrer l'agent après avoir apporté des modifications au fichier de configuration.

Découverte de bas niveau

Vue d'ensemble La découverte de bas niveau (LLD) détecte automatiquement les entités (par exemple, les systèmes de fichiers, les interfaces réseau) sur un hôte et crée les éléments, déclencheurs et graphiques correspondants sans configuration manuelle pour chacun d'eux.

Pour utiliser la LLD, vous créez une **règle de découverte** qui collecte des données JSON décrivant les entités, ainsi que des **prototypes** (élément, déclencheur et graphique) pour chaque entité.

Le JSON renvoyé par la règle doit être un tableau d'objets, où chaque objet représente une entité découverte à l'aide de paires clé-valeur. Par exemple, une règle de découverte avec `net.if.discovery` peut renvoyer :

```
[
  {"{#IFNAME}": "lo"},
  {"{#IFNAME}": "eth0"}
]
```

Un ensemble correspondant d'éléments, de déclencheurs et de graphiques est alors créé à partir des prototypes, un par interface (`lo` et `eth0`).

Si les objets contiennent des clés au format `{#MACRO}`, ces macros sont utilisées directement dans les prototypes. Sinon, ou si des macros supplémentaires/personnalisées sont nécessaires, les macros peuvent être définies manuellement et associées aux champs JSON à l'aide de JSONPath.

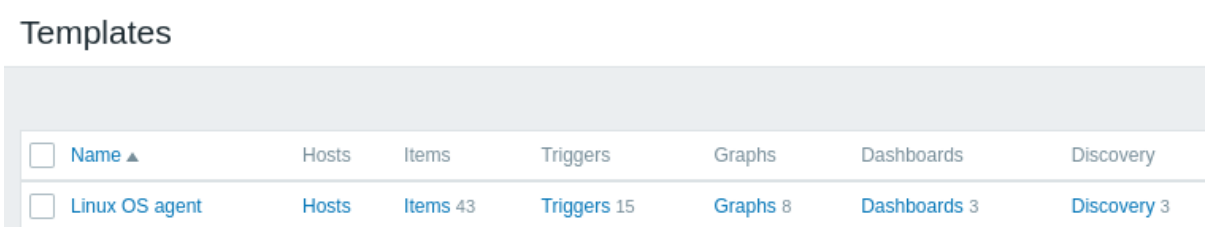
La LLD peut également créer des hôtes (par exemple, pour des machines virtuelles découvertes sur un hyperviseur) et prend en charge des règles de découverte imbriquées pour une découverte à plusieurs niveaux.

Les entités qui ne sont plus découvertes peuvent être automatiquement désactivées ou supprimées.

Configuration de la découverte de bas niveau Nous allons illustrer la découverte de bas niveau à partir d'un exemple de découverte du système de fichiers.

Pour configurer la découverte, procédez comme suit :

1. Accédez à *Collecte de données* > *Modèles* ou *Hôtes*.
2. Cliquez sur *Découverte* dans la ligne du modèle/de l'hôte approprié.



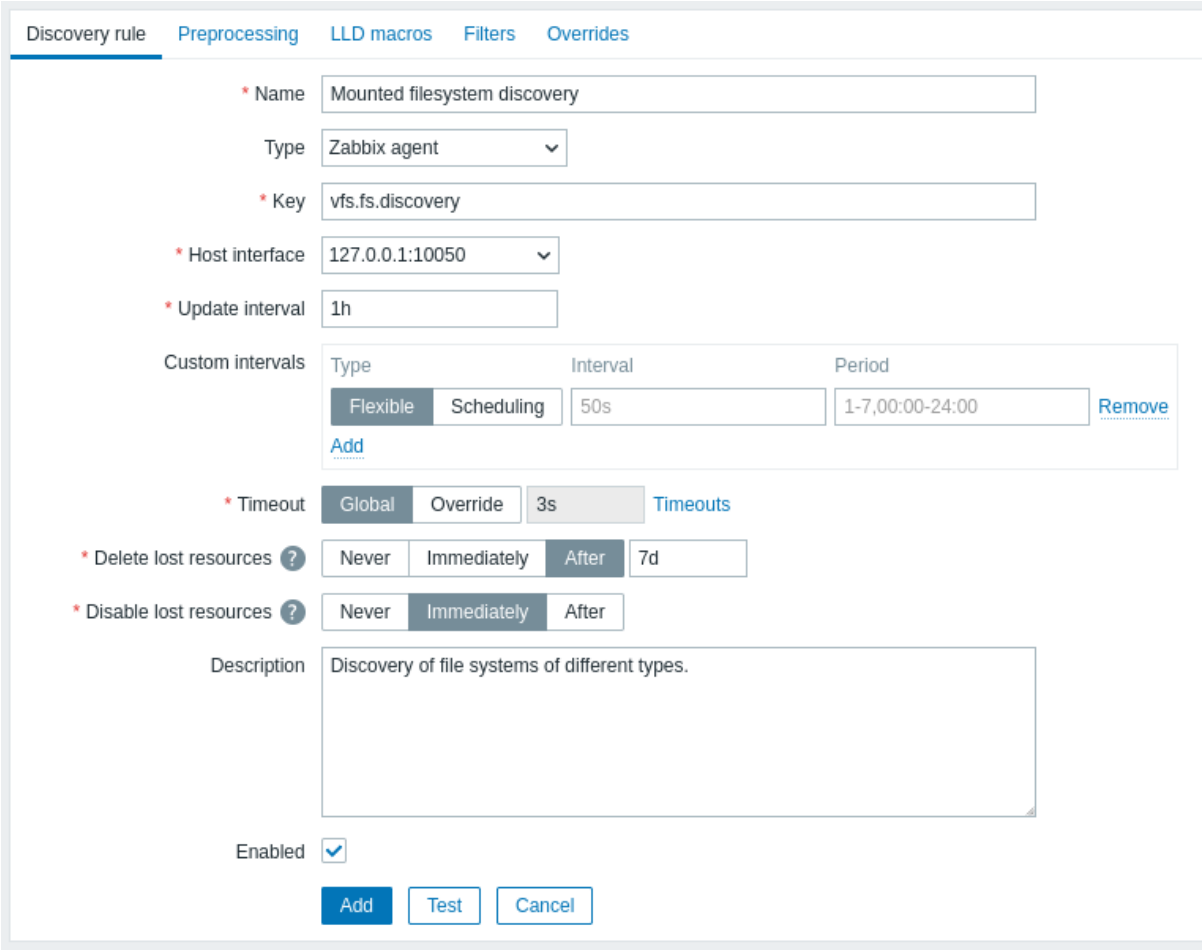
3. Cliquez sur *Créer une règle de découverte* dans le coin supérieur droit de l'écran.
4. Renseignez le formulaire de la règle de découverte avec les informations requises.

Règle de découverte

Le formulaire de règle de découverte contient cinq onglets représentant, de gauche à droite, le flux de données lors de la découverte :

- *Règle de découverte* - spécifiez notamment l'élément intégré ou le script personnalisé permettant de récupérer les données de découverte.
- *Prétraitement* - appliquez un prétraitement aux données découvertes.
- *Macros LLD* - permet d'extraire les valeurs des macros à utiliser dans les éléments découverts, les déclencheurs, etc.
- *Filtres* - permet de filtrer les valeurs découvertes.
- *Remplacements* - permet de modifier les éléments, les déclencheurs, les graphiques ou les prototypes d'hôtes lors de l'application à des objets découverts spécifiques.

L'onglet **Règle de découverte** contient la clé de l'élément à utiliser pour la découverte (ainsi que certains attributs généraux de la règle de découverte) :



Tous les champs de saisie obligatoires sont marqués d'un astérisque rouge.

Paramètre	Description
Nom	Nom de la règle de découverte.
Type	Type de vérification à effectuer pour la découverte. Dans cet exemple, nous utilisons un type d'élément *Zabbix agent*. La règle de découverte peut également être un élément dépendant, dépendant d'un élément standard. Elle ne peut pas dépendre d'une autre règle de découverte. Pour un élément dépendant, sélectionnez le type correspondant (*Élément dépendant*) et spécifiez l'élément maître dans le champ « Élément maître ». L'élément maître doit exister.
Clé	Saisissez la clé de l'élément de découverte (jusqu'à 2048 caractères). Par exemple, vous pouvez utiliser la clé d'élément intégrée « vfs.fs.discovery » pour renvoyer une chaîne JSON contenant la liste des systèmes de fichiers présents sur l'ordinateur, leurs types et leurs options de montage. Notez qu'une autre possibilité pour découvrir les systèmes de fichiers consiste à utiliser les résultats de découverte obtenus avec la clé d'agent « vfs.fs.get » (voir l'exemple).

Paramètre	Description
Intervalle de mise à jour	Ce champ spécifie la fréquence à laquelle Zabbix effectue la découverte. Au début, lors de la configuration de la découverte des systèmes de fichiers, vous pouvez souhaiter définir un intervalle court, mais une fois que vous savez qu'elle fonctionne, vous pouvez le définir sur 30 minutes ou davantage, car les systèmes de fichiers ne changent généralement pas très souvent. Les suffixes de temps sont pris en charge, par exemple 30s, 1m, 2h, 1d. Les macros utilisateur sont prises en charge. *Remarque* : l'intervalle de mise à jour ne peut être défini sur « 0 » que si des intervalles personnalisés existent avec une valeur différente de zéro. Si la valeur est « 0 » et qu'un intervalle personnalisé (flexible ou planifié) existe avec une valeur différente de zéro, l'élément sera interrogé pendant la durée de l'intervalle personnalisé. Les nouvelles règles de découverte seront vérifiées dans les 60 secondes suivant leur création, sauf si elles disposent d'un intervalle de mise à jour planifié ou flexible et que l'*Intervalle de mise à jour* est défini sur 0. *Remarque* : pour une règle de découverte existante, la découverte peut être effectuée immédiatement en cliquant sur le bouton *Exécuter maintenant*.
Intervalles personnalisés	Vous pouvez créer des règles personnalisées pour vérifier l'élément : **Flexible** - crée une exception à l'*Intervalle de mise à jour* (intervalle avec une fréquence différente) **Planification** - crée un calendrier d'interrogation personnalisé. Pour plus d'informations, consultez Intervalles personnalisés.
Délai d'expiration	Définissez le délai d'expiration de la vérification de découverte. Sélectionnez l'option de délai d'expiration : **Global** - le délai d'expiration du proxy/global est utilisé (affiché dans le champ *Délai d'expiration* grisé) ; **Remplacement** - un délai d'expiration personnalisé est utilisé (défini dans le champ *Délai d'expiration* ; plage autorisée : 1 - 600s). Les suffixes de temps, par exemple 30s et 1m, ainsi que les macros utilisateur sont pris en charge. Le lien *Délais d'expiration* permet de configurer les délais d'expiration du proxy ou les délais d'expiration globaux (si aucun proxy n'est utilisé). Notez que le lien *Délais d'expiration* est visible uniquement pour les utilisateurs de type *Super administrateur* disposant des autorisations nécessaires pour accéder aux sections de l'interface *Administration* > *Général* ou *Administration* > *Proxys*.
Supprimer les ressources perdues	Spécifiez le délai au terme duquel l'entité découverte sera supprimée lorsque son état de découverte devient « Plus découverte » : *Jamais* - elle ne sera pas supprimée ; *Immédiatement* - elle sera supprimée immédiatement ; *Après* - elle sera supprimée après la période spécifiée. La valeur doit être supérieure à la valeur de *Désactiver les ressources perdues*. Les suffixes de temps sont pris en charge, par exemple 2h et 1d. Les macros utilisateur sont prises en charge. *Remarque* : l'utilisation de l'option « Immédiatement » n'est pas recommandée, car une simple modification incorrecte du filtre peut entraîner la suppression de l'entité et de toutes ses données historiques. Notez que les ressources désactivées manuellement ne seront pas supprimées par la découverte de bas niveau.
Désactiver les ressources perdues	Spécifiez le délai au terme duquel l'entité découverte sera désactivée lorsque son état de découverte devient « Plus découverte » : *Jamais* - elle ne sera pas désactivée ; *Immédiatement* - elle sera désactivée immédiatement ; *Après* - elle sera désactivée après la période spécifiée. La valeur doit être supérieure à l'intervalle de mise à jour de la règle de découverte. Notez que les ressources désactivées automatiquement seront de nouveau activées si elles sont redécouvertes par la découverte de bas niveau. Les ressources désactivées manuellement ne seront pas réactivées si elles sont redécouvertes. Ce champ n'est pas affiché si *Supprimer les ressources perdues* est défini sur « Immédiatement ». Les suffixes de temps sont pris en charge, par exemple 2h et 1d. Les macros utilisateur sont prises en charge.
Description	Saisissez une description.
Activé	Si cette option est cochée, la règle sera traitée.

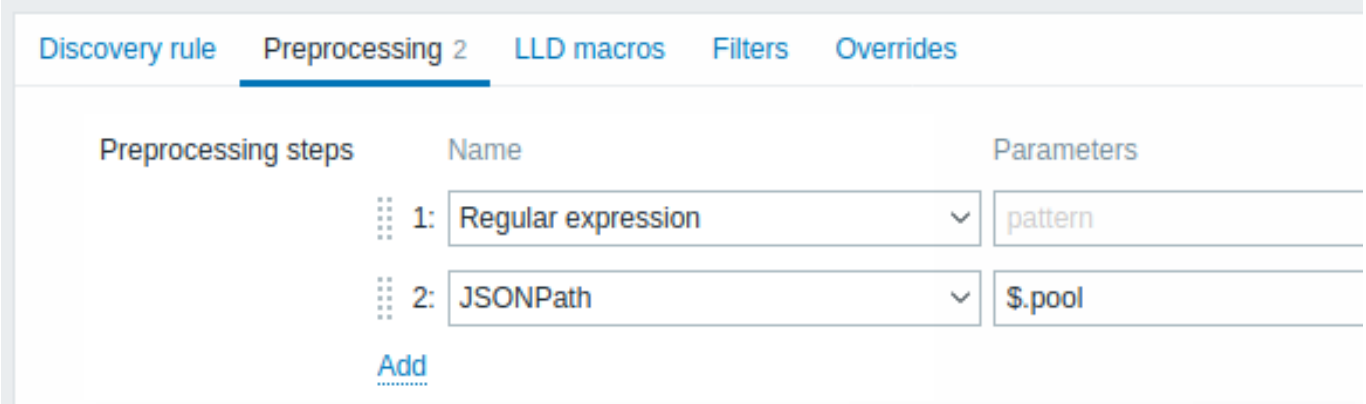
Note:
L'historique de la règle de découverte n'est pas conservé.

Prétraitement

L'onglet **Prétraitement** vous permet de définir des règles de transformation à appliquer au résultat de la découverte. Une ou plusieurs transformations sont possibles à cette étape. Les transformations sont exécutées dans l'ordre dans lequel elles sont définies. Tout le prétraitement est effectué par le serveur Zabbix.

Voir également :

- Détails du prétraitement
- Test du prétraitement



Type		
	Transformation	Description
Texte		
	Expression régulière	Faire correspondre la valeur reçue à l'expression régulière <pattern> et remplacer la valeur par la valeur extraite <output>. L'expression régulière prend en charge l'extraction de 10 groupes capturés au maximum avec la séquence \N. Paramètres : **pattern** - expression régulière **output** - modèle de formatage de la sortie. Une séquence d'échappement \N (où N=1...9) est remplacée par le Nième groupe correspondant. Une séquence d'échappement \0 est remplacée par le texte correspondant. Si vous cochez la case *Personnalisé en cas d'échec*, vous pouvez spécifier des options personnalisées de gestion des erreurs : ignorer la valeur, définir une valeur spécifiée ou définir un message d'erreur spécifié.
	Remplacer	Rechercher la chaîne et la remplacer par une autre chaîne (ou par rien). Toutes les occurrences de la chaîne recherchée seront remplacées. Paramètres : **search string** - chaîne à rechercher et à remplacer, sensible à la casse (obligatoire) **replacement** - chaîne par laquelle remplacer la chaîne recherchée. La chaîne de remplacement peut également être vide, ce qui permet effectivement de supprimer la chaîne recherchée lorsqu'elle est trouvée. Il est possible d'utiliser des séquences d'échappement pour rechercher ou remplacer des sauts de ligne, des retours chariot, des tabulations et des espaces "\n \r \t \s" ; la barre oblique inverse peut être échappée avec "\\" et les séquences d'échappement peuvent être échappées avec "\\n". L'échappement des sauts de ligne, des retours chariot et des tabulations est effectué automatiquement pendant la découverte de bas niveau.
Données struc-turées		
	JSONPath	Extraire une valeur ou un fragment de données JSON à l'aide de la fonctionnalité JSONPath. Si vous cochez la case *Personnalisé en cas d'échec*, vous pouvez spécifier des options personnalisées de gestion des erreurs : ignorer la valeur, définir une valeur spécifiée ou définir un message d'erreur spécifié.

Type	

XPath XML	Extraire une valeur ou un fragment de données XML à l'aide de la fonctionnalité XPath.
	Pour que cette option fonctionne, le serveur Zabbix doit être compilé avec la prise en charge de libxml.
	Exemples :
	`number(/document/item/value)` extraira 10 de `<document><item><value>10</value></item></document>`
	`number(/document/item/@attribute)` extraira 10 de `<document><item attribute="10"></item></document>`
	`/document/item` extraira `<item><value>10</value></item>` de `<document><item><value>10</value></item></document>`
	Notez que les espaces de noms ne sont pas pris en charge.
	Si vous cochez la case *Personnalisé en cas d'échec*, vous pouvez spécifier des options personnalisées de gestion des erreurs : ignorer la valeur, définir une valeur spécifiée ou définir un message d'erreur spécifié.

CSV vers JSON	Convertir les données d'un fichier CSV au format JSON. Pour plus d'informations, voir : Prétraitement CSV vers JSON.
XML vers JSON	Convertir des données au format XML en JSON. Pour plus d'informations, voir : Règles de sérialisation. Si vous cochez la case *Personnalisé en cas d'échec*, vous pouvez spécifier des options personnalisées de gestion des erreurs : ignorer la valeur, définir une valeur spécifiée ou définir un message d'erreur spécifié.
SNMP	
Valeur de parcours SNMP	Extraire la valeur à l'aide du nom OID/MIB spécifié et appliquer les options de formatage : **Inchangée** - renvoyer HEX-STRING sous forme de chaîne hexadécimale non échappée (*notez* que les indications d'affichage sont toujours appliquées) ; **UTF-8 depuis HEX-STRING** - convertir HEX-STRING en chaîne UTF-8 ; **MAC depuis HEX-STRING** - convertir HEX-STRING en chaîne d'adresse MAC (où `' '` sera remplacé par `':'`) ; **Entier depuis BITS** - convertir les 8 premiers octets d'une chaîne de bits exprimée sous forme de séquence de caractères hexadécimaux (par exemple, "1A 2B 3C 4D") en entier non signé de 64 bits. Dans les chaînes de bits de plus de 8 octets, les octets suivants seront ignorés. Si vous cochez la case *Personnalisé en cas d'échec*, vous pouvez spécifier des options personnalisées de gestion des erreurs : ignorer la valeur, définir une valeur spécifiée ou définir un message d'erreur spécifié.
Parcours SNMP vers JSON	Convertir les valeurs SNMP en JSON. Spécifier un nom de champ dans le JSON et le chemin OID SNMP correspondant. Les valeurs des champs seront renseignées avec les valeurs du chemin OID SNMP spécifié. Vous pouvez utiliser cette étape de prétraitement pour la découverte des OID SNMP. Des options de formatage des valeurs similaires à celles de l'étape *Valeur de parcours SNMP* sont disponibles. Si vous cochez la case *Personnalisé en cas d'échec*, vous pouvez spécifier des options personnalisées de gestion des erreurs : ignorer la valeur, définir une valeur spécifiée ou définir un message d'erreur spécifié.
Valeur de récupération SNMP	Appliquer les options de formatage à la valeur récupérée via SNMP : **UTF-8 depuis HEX-STRING** - convertir HEX-STRING en chaîne UTF-8 ; **MAC depuis HEX-STRING** - convertir HEX-STRING en chaîne d'adresse MAC (où `' '` sera remplacé par `':'`) ; **Entier depuis BITS** - convertir les 8 premiers octets d'une chaîne de bits exprimée sous forme de séquence de caractères hexadécimaux (par exemple, "1A 2B 3C 4D") en entier non signé de 64 bits. Dans les chaînes de bits de plus de 8 octets, les octets suivants seront ignorés. Si vous cochez la case *Personnalisé en cas d'échec*, vous pouvez spécifier des options personnalisées de gestion des erreurs : ignorer la valeur, définir une valeur spécifiée ou définir un message d'erreur spécifié.
Scripts personnalisés	

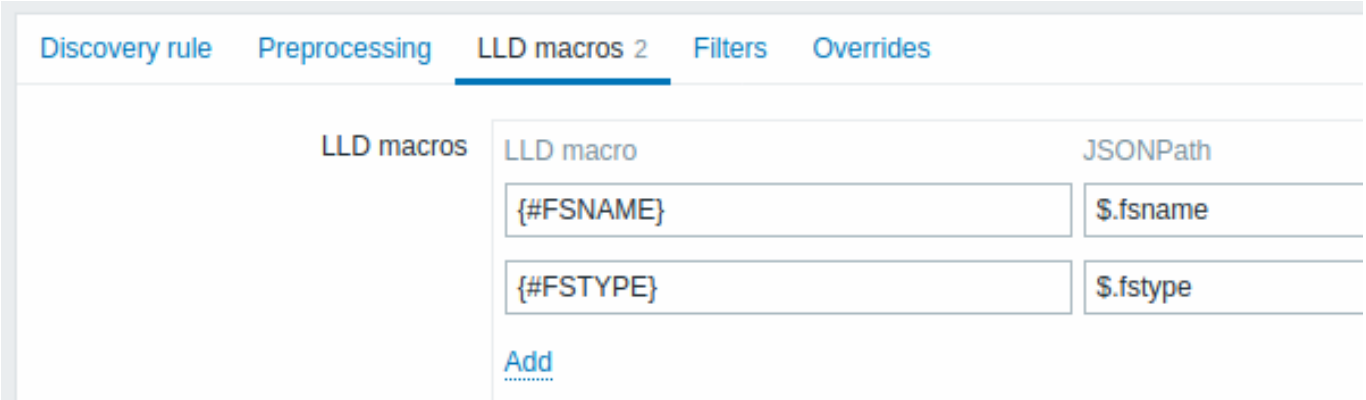
Type	
JavaScript	Saisir du code JavaScript dans l'éditeur modal qui s'ouvre lorsque vous cliquez dans le champ de paramètre ou sur l'icône en forme de crayon située à côté. Notez que la longueur de JavaScript disponible dépend de la base de données utilisée. Pour plus d'informations, voir : Prétraitement Javascript
Validation	
Ne correspond pas à l'expression régulière	Spécifier une expression régulière à laquelle une valeur ne doit pas correspondre. Par exemple, `Error:(.*?)\.` Si vous cochez la case *Personnalisé en cas d'échec*, vous pouvez spécifier des options personnalisées de gestion des erreurs : ignorer la valeur, définir une valeur spécifiée ou définir un message d'erreur spécifié.
Rechercher une erreur dans JSON	Rechercher un message d'erreur au niveau de l'application situé au chemin JSONPath. Arrêter le traitement si la recherche aboutit et que le message n'est pas vide ; sinon, continuer le traitement avec la valeur qui précédait cette étape de prétraitement. Notez que ces erreurs de services externes sont signalées à l'utilisateur telles quelles, sans ajouter d'informations sur l'étape de prétraitement. Par exemple, `$.errors`. Si un JSON tel que `{"errors":"e1"}` est reçu, l'étape de prétraitement suivante ne sera pas exécutée. Si vous cochez la case *Personnalisé en cas d'échec*, vous pouvez spécifier des options personnalisées de gestion des erreurs : ignorer la valeur, définir une valeur spécifiée ou définir un message d'erreur spécifié.
Rechercher une erreur dans XML	Rechercher un message d'erreur au niveau de l'application situé à l'emplacement XPath. Arrêter le traitement si la recherche aboutit et que le message n'est pas vide ; sinon, continuer le traitement avec la valeur qui précédait cette étape de prétraitement. Notez que ces erreurs de services externes sont signalées à l'utilisateur telles quelles, sans ajouter d'informations sur l'étape de prétraitement. Aucune erreur ne sera signalée en cas d'échec de l'analyse d'un XML non valide. Si vous cochez la case *Personnalisé en cas d'échec*, vous pouvez spécifier des options personnalisées de gestion des erreurs : ignorer la valeur, définir une valeur spécifiée ou définir un message d'erreur spécifié.
Correspond à l'expression régulière	Spécifier une expression régulière à laquelle une valeur doit correspondre. Si vous cochez la case *Personnalisé en cas d'échec*, vous pouvez spécifier des options personnalisées de gestion des erreurs : ignorer la valeur, définir une valeur spécifiée ou définir un message d'erreur spécifié.
Limitation	
Ignorer les valeurs inchangées avec battement de cœur	Ignorer une valeur si elle n'a pas changé pendant la période définie (en secondes). Les valeurs entières positives sont prises en charge pour spécifier le nombre de secondes (minimum : 1 seconde). Les suffixes de temps peuvent être utilisés dans ce champ (par exemple, `30s`, `1m`, `2h`, `1d`). Les macros utilisateur et les macros de découverte de bas niveau peuvent être utilisées dans ce champ. Une seule option de limitation peut être spécifiée pour un élément de découverte. Par exemple, `1m`. Si un texte identique est transmis deux fois à cette règle dans un intervalle de 60 secondes, il sera ignoré. *Remarque* : la modification des prototypes d'éléments ne réinitialise pas la limitation. La limitation est réinitialisée uniquement lorsque les étapes de prétraitement sont modifiées.
Prometheus	
Prometheus vers JSON	Convertir les métriques Prometheus requises en JSON. Voir Vérifications Prometheus pour plus d'informations.

Notez que si la règle de découverte a été appliquée au hôte via un modèle, le contenu de cet onglet est en lecture seule.

Macros personnalisées

L'onglet **Macros LLD** vous permet de spécifier des macros LLD personnalisées lorsque le JSON renvoyé ne contient pas déjà de clés au format `{#MACRO}`.

Par exemple, la clé d'élément agent `vfs.fs.get` renvoie des données JSON de système de fichiers sans aucune macro LLD prédéfinie.
Ici, vous définissez les macros manuellement et les associez à des valeurs dans le JSON à l'aide d'expressions JSONPath :



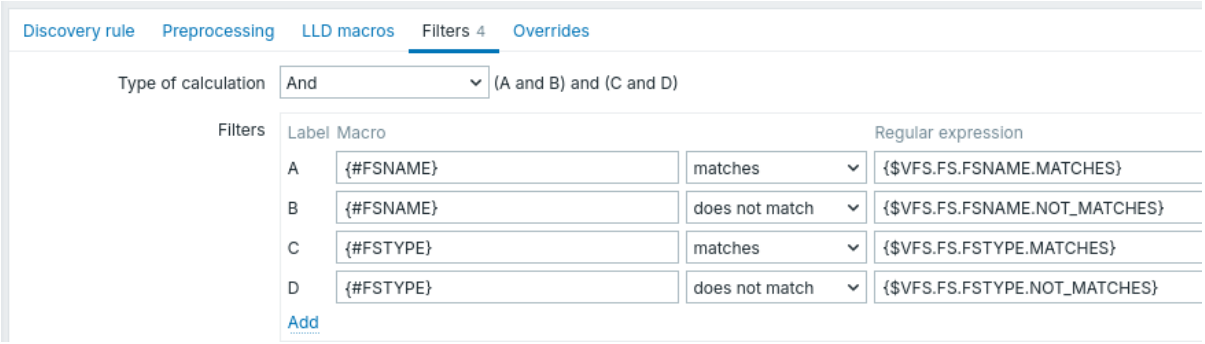
Parameter	Description
LLD macro	Nom de la macro LLD au format `{#MACRO}`.
JSONPath	Expression qui définit quelle valeur extraire de chaque entité découverte dans le tableau JSON. Par exemple, `$.foo` extrait `"bar"` et `"baz"` de `[{"foo":"bar"}, {"foo":"baz"}]`. JSONPath prend en charge la notation par points et la notation par crochets. La notation par crochets doit être utilisée pour les clés contenant des caractères spéciaux ou Unicode, par exemple, `$['unicode + special chars #1']['unicode + special chars #2']`.

La règle de découverte doit renvoyer un tableau JSON à la racine ($.).
Pour des raisons de compatibilité ascendante, un objet JSON contenant un seul tableau `"data"` est également accepté. Dans ce cas, le tableau `"data"` est automatiquement déballé :

```
{"data":[{"foo":"bar"}, {"foo":"baz"}]}
```

Filtre

Un filtre peut être utilisé pour générer des éléments, des déclencheurs et des graphiques réels uniquement pour les entités correspondant aux critères. L'onglet **Filtres** contient les définitions des filtres des règles de découverte, qui vous permettent de filtrer les valeurs découvertes :



Paramètre	Description
Type de calcul	Les options suivantes sont disponibles pour le calcul des filtres : **Et** - tous les filtres doivent être validés ; **Ou** - un seul filtre doit être validé ; **Et/Ou** - utilise *Et* avec des noms de macros différents et *Ou* avec le même nom de macro ; **Expression personnalisée** - permet de définir un calcul personnalisé des filtres. La formule doit inclure tous les filtres de la liste. Limitée à 255 caractères.

Paramètre	Description
Filtres	Les opérateurs de condition de filtre suivants sont disponibles : *correspond à*, *ne correspond pas à*, *existe*, *n'existe pas*. Les opérateurs *correspond à* et *ne correspond pas à* attendent une expression régulière compatible avec Perl (PCRE). Vous pouvez saisir une expression régulière ou référencer une expression régulière globale dans le champ *Expression régulière*. Les opérateurs *existe* et *n'existe pas* permettent de filtrer les entités selon la présence ou l'absence de la macro LLD spécifiée dans la réponse. Notez que si une macro du filtre est absente de la réponse, l'entité trouvée sera ignorée, sauf si la condition « n'existe pas » est spécifiée pour cette macro. Un avertissement s'affiche si l'absence d'une macro affecte le résultat de l'expression. Par exemple, si `{#B}` est absente dans : `{#A}` correspond à 1 et `{#B}` correspond à 2 - un avertissement s'affiche `{#A}` correspond à 1 ou `{#B}` correspond à 2 - aucun avertissement ne s'affiche.

Par exemple, si vous vous intéressez uniquement aux systèmes de fichiers C:, D: et E:, vous pouvez saisir `{#FSNAME}` dans le champ *Macro* et `^C|^D|^E` dans le champ *Expression régulière*.
 Le filtrage est également possible par types de systèmes de fichiers à l'aide de la macro `{#FSTYPE}` (par exemple, `^ext|^reiserfs`) et par types de lecteurs (pris en charge uniquement par Windows agent) à l'aide de la macro `{#FSDRIVETYPE}` (par exemple, `fixed`).

Pour tester une expression régulière, vous pouvez utiliser `grep -E`, par exemple :

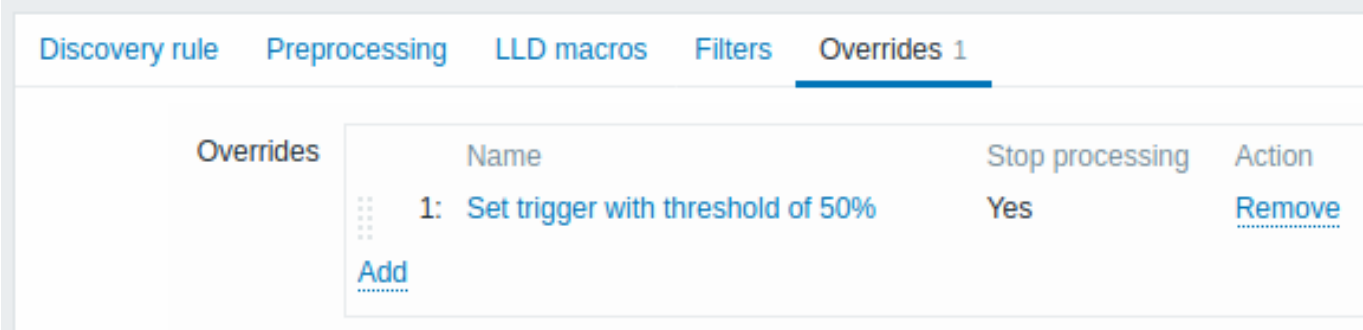
```
for f in ext2 nfs reiserfs smbfs;
do echo $f | grep -E '^ext|^reiserfs' || echo "SKIP: $f";
done
```

Warning:
Une erreur ou une faute de frappe dans l'expression régulière utilisée dans la règle LLD (par exemple, une expression régulière incorrecte dans « Systèmes de fichiers à découvrir ») peut entraîner la suppression de milliers d'éléments de configuration, de valeurs historiques et d'événements pour de nombreux hôtes.

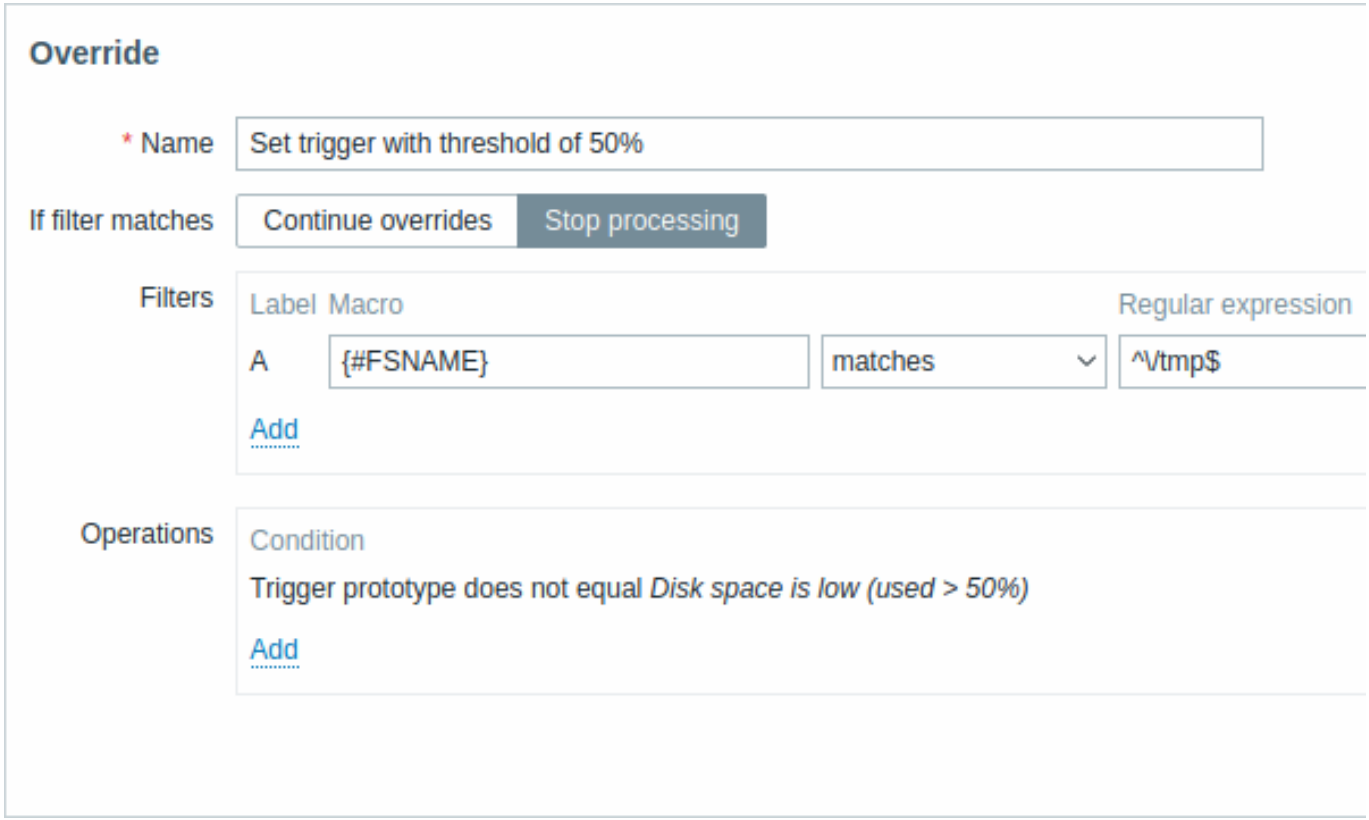
Attention:
La base de données Zabbix dans MySQL doit être créée en respectant la casse pour que les noms de systèmes de fichiers qui diffèrent uniquement par la casse soient correctement découverts.

Remplacements

L'onglet **Remplacements** permet de définir des règles pour modifier la liste des prototypes d'élément, de déclencheur, de graphique, d'hôte et de découverte, ou leurs attributs, pour les objets découverts qui répondent à certains critères.



Les remplacements (le cas échéant) sont affichés dans une liste réorganisable par glisser-déposer et sont exécutés dans l'ordre dans lequel ils sont définis. Pour configurer les détails d'un nouveau remplacement, cliquez sur Add dans le bloc *Remplacements*. Pour modifier un remplacement existant, cliquez sur son nom. Une fenêtre contextuelle s'ouvre et vous permet de modifier les détails de la règle de remplacement.

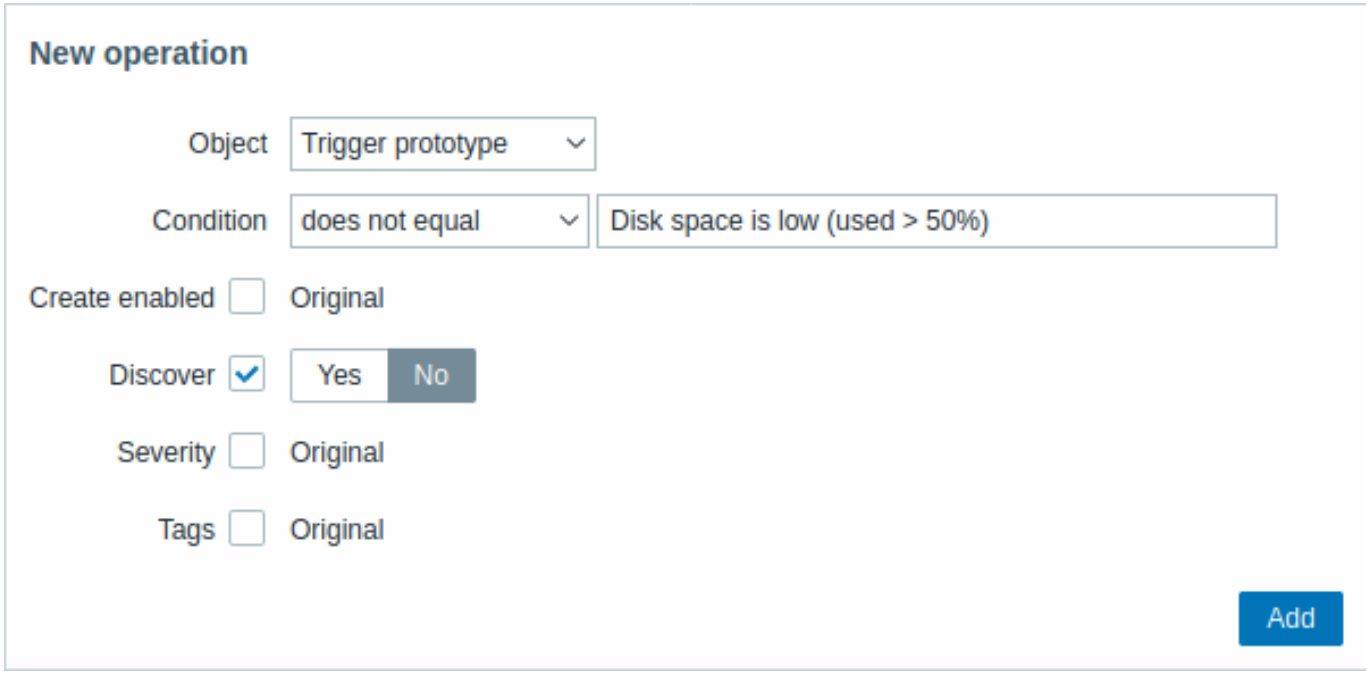


Tous les paramètres obligatoires sont signalés par des astérisques rouges.

Paramètre	Description
Nom	Nom unique (pour chaque règle LLD) du remplacement.
Si le filtre correspond	Définit si les remplacements suivants doivent être traités lorsque les conditions du filtre sont remplies : **Continuer les remplacements** - les remplacements suivants seront traités. **Arrêter le traitement** - les opérations des remplacements précédents (le cas échéant) et de ce remplacement seront exécutées ; les remplacements suivants seront ignorés pour les lignes LLD correspondantes.
Filtres	Détermine à quelles entités découvertes le remplacement doit être appliqué. Les filtres de remplacement sont traités après les filtres de la règle de découverte et ont la même fonctionnalité.
Opérations	Les opérations de remplacement sont affichées avec les détails suivants : **Condition** - un type d'objet et une condition à remplir pour le nom de l'objet ; par exemple : le prototype de déclencheur n'est pas égal à *L'espace disque est faible (utilisé > 50 %)*. **Actions** - des liens permettant de modifier et de supprimer une opération sont affichés.

Configuration d'une opération

Pour configurer les détails d'une nouvelle opération, cliquez sur Add dans le bloc Opérations. Pour modifier une opération existante, cliquez sur Edit à côté de l'opération. Une fenêtre contextuelle s'ouvre et vous permet de modifier les détails de l'opération.



Paramètre	Description
Objet	Cinq types d'objets sont disponibles : Prototype d'élément Prototype de déclencheur Prototype de graphique Prototype d'hôte Prototype de découverte
Condition	Permet de filtrer les entités auxquelles l'opération doit être appliquée.
Opérateur	Opérateurs pris en charge : **est égal à** - appliquer à ce prototype **n'est pas égal à** - appliquer à tous les prototypes sauf celui-ci **contient** - appliquer si le nom du prototype contient cette chaîne **ne contient pas** - appliquer si le nom du prototype ne contient pas cette chaîne **correspond** - appliquer si le nom du prototype correspond à l'expression régulière **ne correspond pas** - appliquer si le nom du prototype ne correspond pas à l'expression régulière
Modèle	Une expression régulière ou une chaîne à rechercher.
Objet : *Prototype d'élément*	
Créer comme activé	Lorsque la case est cochée, les boutons apparaissent et permettent de remplacer les paramètres du prototype d'élément d'origine : *Oui* - l'élément sera ajouté à l'état activé. *Non* - l'élément sera ajouté à une entité découverte, mais à l'état désactivé.
Découvrir	Lorsque la case est cochée, les boutons apparaissent et permettent de remplacer les paramètres du prototype d'élément d'origine : *Oui* - l'élément sera ajouté. *Non* - l'élément ne sera pas ajouté.
Intervalle de mise à jour	Lorsque la case est cochée, deux options apparaissent et permettent de définir un intervalle différent pour l'élément : *Délai* - intervalle de mise à jour de l'élément. Les macros utilisateur, les macros LLD et les suffixes temporels (par exemple, 30s, 1m, 2h, 1d) sont pris en charge. Doit être défini sur 0 si un *intervalle personnalisé* est utilisé. *Intervalle personnalisé* - cliquez sur Add pour spécifier des intervalles flexibles ou planifiés. Pour plus d'informations, consultez Intervalles personnalisés.

Paramètre	Description
Historique	Lorsque la case est cochée, les boutons apparaissent et permettent de définir différentes périodes de conservation de l'historique pour l'élément : *Ne pas stocker* - si cette option est sélectionnée, l'historique ne sera pas stocké. *Stocker pendant* - si cette option est sélectionnée, un champ de saisie permettant de spécifier la période de conservation apparaît à droite. Les macros utilisateur et les macros LLD sont prises en charge.
Tendances	Lorsque la case est cochée, les boutons apparaissent et permettent de définir différentes périodes de conservation des tendances pour l'élément : *Ne pas stocker* - si cette option est sélectionnée, les tendances ne seront pas stockées. *Stocker pendant* - si cette option est sélectionnée, un champ de saisie permettant de spécifier la période de conservation apparaît à droite. Les macros utilisateur et les macros LLD sont prises en charge.
Étiquettes	Lorsque la case est cochée, un nouveau bloc apparaît et permet de spécifier des paires étiquette-valeur. Les macros utilisateur et les macros LLD sont prises en charge. Ces étiquettes seront ajoutées aux étiquettes spécifiées dans le prototype d'élément, même si les noms des étiquettes sont identiques.
Objet : *Prototype de déclencheur*	
Créer comme activé	Lorsque la case est cochée, les boutons apparaissent et permettent de remplacer les paramètres du prototype de déclencheur d'origine : *Oui* - le déclencheur sera ajouté à l'état activé. *Non* - le déclencheur sera ajouté à une entité découverte, mais à l'état désactivé.
Découvrir	Lorsque la case est cochée, les boutons apparaissent et permettent de remplacer les paramètres du prototype de déclencheur d'origine : *Oui* - le déclencheur sera ajouté. *Non* - le déclencheur ne sera pas ajouté.
Gravité	Lorsque la case est cochée, les boutons de gravité du déclencheur apparaissent et permettent de modifier la gravité du déclencheur.
Étiquettes	Lorsque la case est cochée, un nouveau bloc apparaît et permet de spécifier des paires étiquette-valeur. Les macros utilisateur et les macros LLD sont prises en charge. Ces étiquettes seront ajoutées aux étiquettes spécifiées dans le prototype de déclencheur, même si les noms des étiquettes sont identiques.
Objet : *Prototype de graphique*	
Découvrir	Lorsque la case est cochée, les boutons apparaissent et permettent de remplacer les paramètres du prototype de graphique d'origine : *Oui* - le graphique sera ajouté. *Non* - le graphique ne sera pas ajouté.
Objet : *Prototype d'hôte*	
Créer comme activé	Lorsque la case est cochée, les boutons apparaissent et permettent de remplacer les paramètres du prototype d'hôte d'origine : *Oui* - l'hôte sera créé à l'état activé. *Non* - l'hôte sera créé à l'état désactivé.

Paramètre		Description
	Découvrir	Lorsque la case est cochée, les boutons apparaissent et permettent de remplacer les paramètres du prototype d'hôte d'origine : *Oui* - l'hôte sera découvert. *Non* - l'hôte ne sera pas découvert.
	Lier des modèles	Lorsque la case est cochée, un champ de saisie permettant de spécifier des modèles apparaît. Commencez à saisir le nom du modèle ou cliquez sur *Sélectionner* à côté du champ, puis sélectionnez les modèles dans la liste d'une fenêtre contextuelle. Les modèles de ce remplacement sont ajoutés à tous les modèles déjà liés au prototype d'hôte.
	Étiquettes	Lorsque la case est cochée, un nouveau bloc apparaît et permet de spécifier des paires étiquette-valeur. Les macros utilisateur et les macros LLD sont prises en charge. Ces étiquettes seront ajoutées aux étiquettes spécifiées dans le prototype d'hôte, même si les noms des étiquettes sont identiques.
	Inventaire de l'hôte	Lorsque la case est cochée, les boutons apparaissent et permettent de sélectionner un mode d'inventaire différent pour le prototype d'hôte : *Désactivé* - ne pas renseigner l'inventaire de l'hôte *Manuel* - fournir les détails manuellement *Automatisé* - renseigner automatiquement les données d'inventaire de l'hôte à partir des métriques collectées.
Objet : *Prototype de découverte*		
	Créer comme activé	Lorsque la case est cochée, les boutons apparaissent et permettent de remplacer les paramètres du prototype de découverte d'origine : *Oui* - la règle de découverte sera ajoutée à l'état activé. *Non* - la règle de découverte sera ajoutée à l'état désactivé.
	Découvrir	Lorsque la case est cochée, les boutons apparaissent et permettent de remplacer les paramètres du prototype de découverte d'origine : *Oui* - la règle de découverte sera ajoutée. *Non* - la règle de découverte ne sera pas ajoutée.
	Intervalle de mise à jour	Lorsque la case est cochée, deux options apparaissent et permettent de définir un intervalle différent pour la règle de découverte : *Délai* - intervalle de mise à jour de la règle. Les macros utilisateur, les macros LLD et les suffixes temporels (par exemple, 30s, 1m, 2h, 1d) sont pris en charge. Doit être défini sur 0 si un *intervalle personnalisé* est utilisé. *Intervalle personnalisé* - cliquez sur Add pour spécifier des intervalles flexibles ou planifiés. Pour plus d'informations, consultez Intervalles personnalisés.

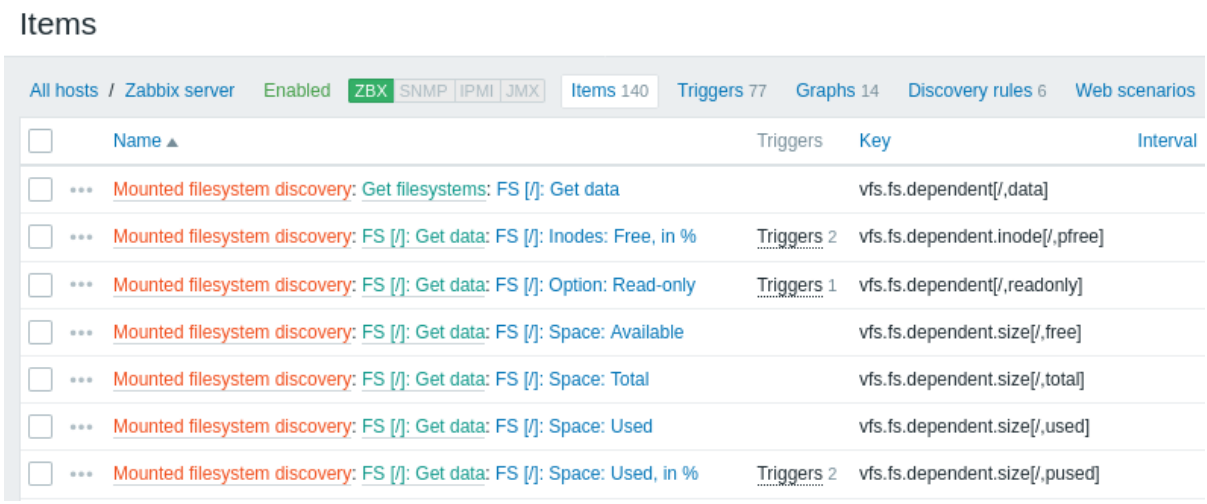
Boutons du formulaire

Les boutons situés en bas du formulaire permettent d'effectuer plusieurs opérations.

Add	Ajouter une règle de découverte. Ce bouton est disponible uniquement pour les nouvelles règles de découverte.
Update	Mettre à jour les propriétés d'une règle de découverte. Ce bouton est disponible uniquement pour les règles de découverte existantes.
Clone	Créer une autre règle de découverte basée sur les propriétés de la règle de découverte actuelle.
Execute now	Effectuer immédiatement la découverte en fonction de la règle de découverte. La règle de découverte doit déjà exister. Voir plus de détails. *Remarque* : lors de l'exécution immédiate de la découverte, le cache de configuration n'est pas mis à jour ; le résultat ne reflétera donc pas les modifications très récentes apportées à la configuration de la règle de découverte.
Test	Tester la configuration de la règle de découverte. Utilisez ce bouton pour vérifier les paramètres de configuration (tels que la connectivité et l'exactitude des paramètres) sans appliquer définitivement de modifications.

Delete	Supprimer la règle de découverte.
Cancel	Annuler la modification des propriétés de la règle de découverte.

Entités découvertes Les captures d'écran ci-dessous illustrent l'apparence des éléments, déclencheurs et graphiques découverts dans la configuration de l'hôte. Les entités découvertes sont précédées d'un lien orange vers la règle de découverte dont elles proviennent.



Notez que les entités découvertes ne seront pas créées s'il existe déjà des entités avec les mêmes critères d'unicité, par exemple un élément avec la même clé ou un graphique portant le même nom. Dans ce cas, un message d'erreur s'affiche dans l'interface web indiquant que la règle de découverte de bas niveau n'a pas pu créer certaines entités. La règle de découverte elle-même ne deviendra toutefois pas non prise en charge parce qu'une entité n'a pas pu être créée et a dû être ignorée. La règle de découverte continuera à créer/mettre à jour les autres entités.

Si une entité découverte (hôte, système de fichiers, interface, etc.) cesse d'être découverte (ou ne correspond plus au filtre), les entités créées à partir de celle-ci peuvent être automatiquement désactivées puis finalement supprimées.

Les ressources perdues peuvent être automatiquement désactivées selon la valeur du paramètre *Disable lost resources*. Cela affecte les hôtes, éléments et déclencheurs perdus.

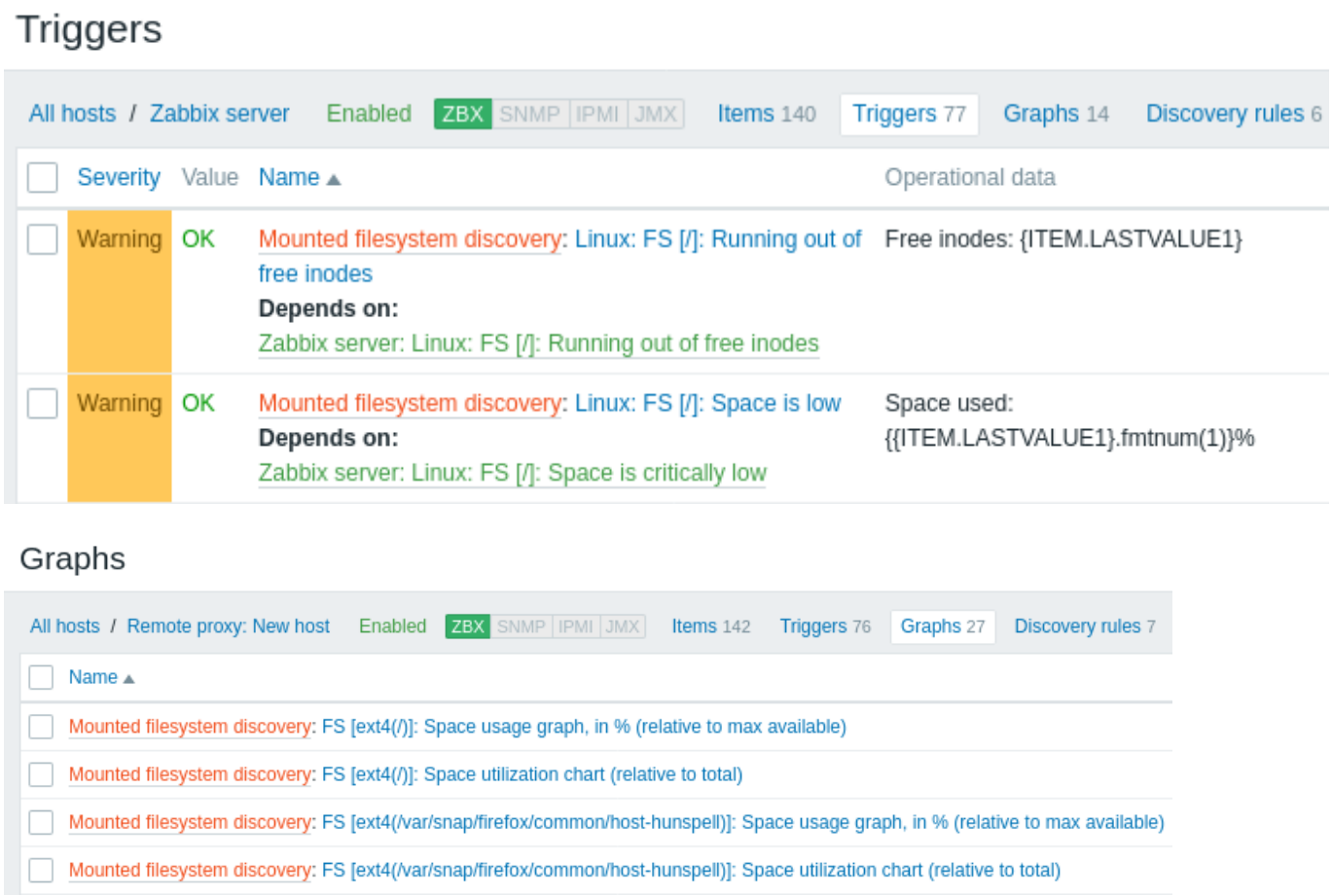
Les ressources perdues peuvent être automatiquement supprimées selon la valeur du paramètre *Delete lost resources*. Cela affecte les hôtes, groupes d'hôtes, éléments, déclencheurs et graphiques perdus.

Lorsque des entités découvertes deviennent « Non découvertes désormais », un indicateur de durée de vie s'affiche dans la liste des entités. Placez le pointeur de la souris dessus et un message s'affichera indiquant les détails de son statut.



Si des entités ont été marquées pour suppression, mais n'ont pas été supprimées au moment prévu (règle de découverte désactivée ou hôte de l'élément), elles seront supprimées lors du prochain traitement de la règle de découverte.

Les entités contenant d'autres entités marquées pour suppression ne seront pas mises à jour si elles sont modifiées au niveau de la règle de découverte. Par exemple, les déclencheurs basés sur LLD ne seront pas mis à jour s'ils contiennent des éléments marqués pour suppression.



Autres types de découverte Vous trouverez plus de détails et des guides pratiques sur d'autres types de découverte prêts à l'emploi dans les sections suivantes :

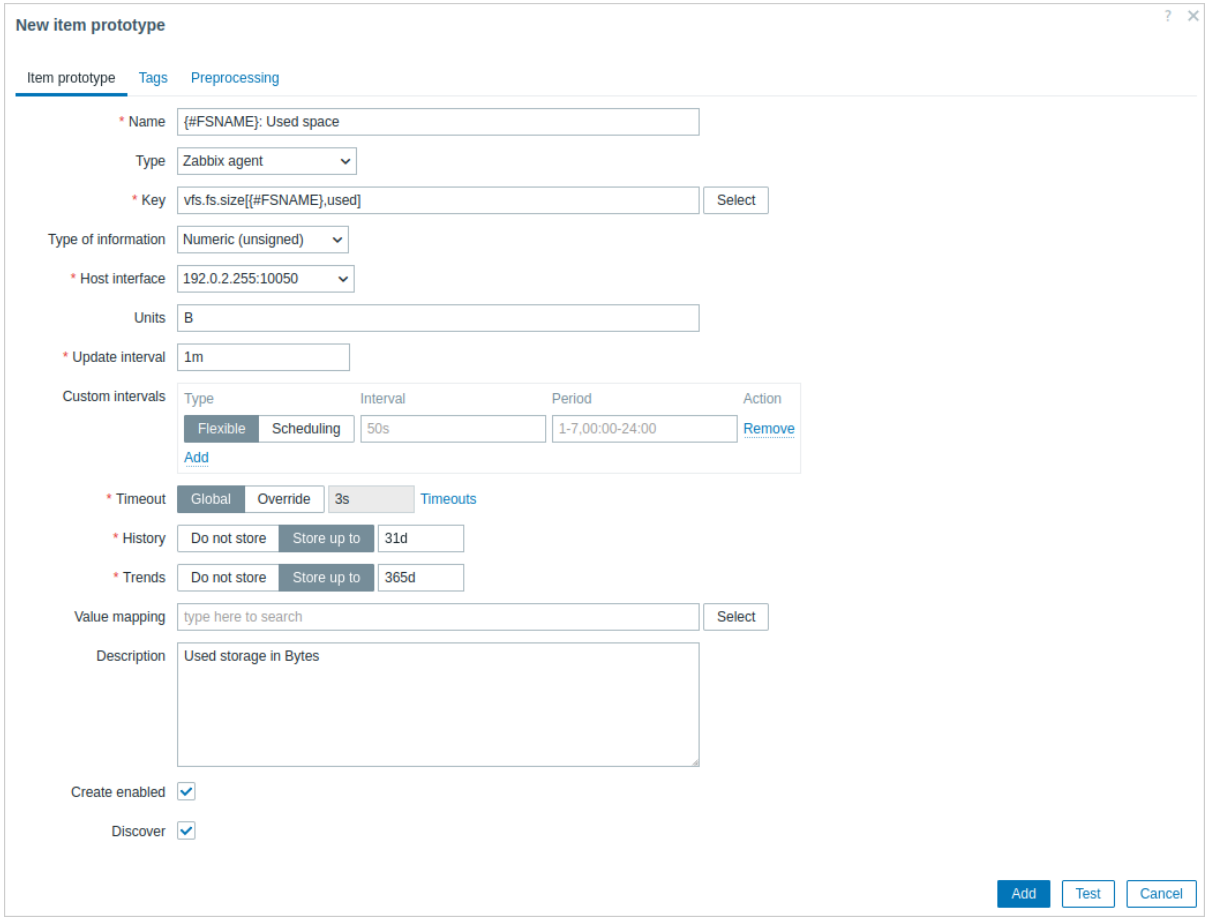
- découverte des interfaces réseau
- découverte des processeurs et des cœurs de processeur
- découverte des OID SNMP
- découverte des objets JMX ;
- découverte à l'aide de requêtes SQL ODBC
- découverte des services Windows
- découverte des interfaces d'hôte dans Zabbix

Pour plus de détails sur le format JSON des éléments de découverte et un exemple de mise en œuvre de votre propre outil de découverte du système de fichiers sous forme de script Perl, consultez la création de règles LLD personnalisées.

Prototypes d'éléments

Une fois qu'une règle est créée, accédez aux éléments de cette règle et cliquez sur *Créer un prototype d'élément* pour créer un prototype d'élément.

Notez comment la macro `{#FSNAME}` est utilisée lorsqu'un nom de système de fichiers est requis. L'utilisation d'une macro de découverte de bas niveau est obligatoire dans la clé de l'élément afin de garantir le traitement correct de la découverte. Lorsque la règle de découverte est traitée, cette macro est remplacée par le système de fichiers découvert.



Note:
Lorsque les paramètres de la clé de l'élément ne sont pas utilisés, placez la macro LLD entre les crochets de la clé de l'élément `[...]` (par exemple, `v_[{#MACRO}]`).

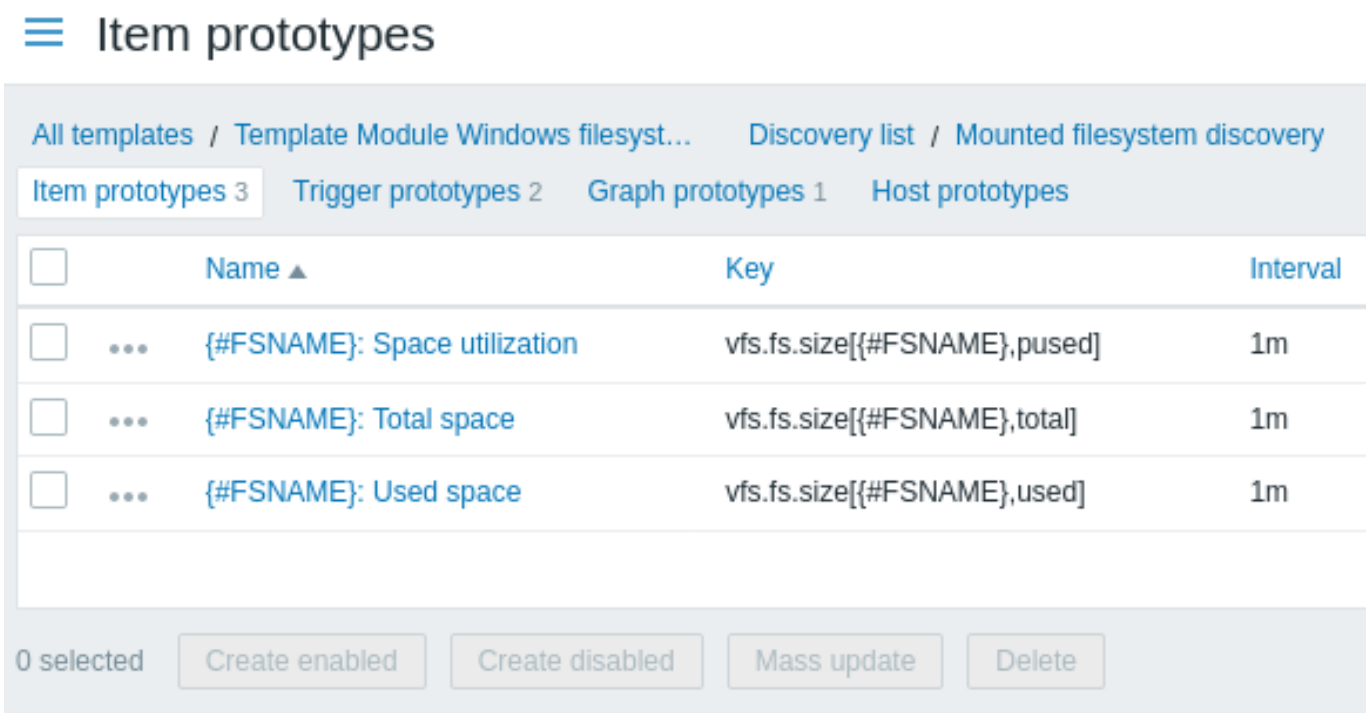
Les macros de découverte de bas niveau et les macros utilisateur sont prises en charge dans la configuration des prototypes d'éléments et dans les paramètres de prétraitement des valeurs des éléments.Notez que lorsqu'une seule macro est utilisée dans les intervalles de mise à jour, elle doit remplir l'intégralité du champ. Plusieurs macros dans un même champ ou des macros mélangées à du texte ne sont pas prises en charge.

Note:
L'échappement contextuel des macros de découverte de bas niveau est effectué pour permettre une utilisation sûre dans les paramètres de prétraitement des expressions régulières et de XPath.

Attributs spécifiques aux prototypes d'éléments :

Paramètre	Description
Créer comme activé	Si cette option est cochée, l'élément sera ajouté à l'état activé. Si elle n'est pas cochée, l'élément sera ajouté à une entité découverte, mais à l'état désactivé.
Découvrir	Si cette option est cochée (par défaut), l'élément sera ajouté à une entité découverte. Si elle n'est pas cochée, l'élément ne sera pas ajouté à une entité découverte, sauf si ce paramètre est remplacé dans la règle de découverte.

Nous pouvons créer plusieurs prototypes d'éléments pour chaque métrique de système de fichiers qui nous intéresse :



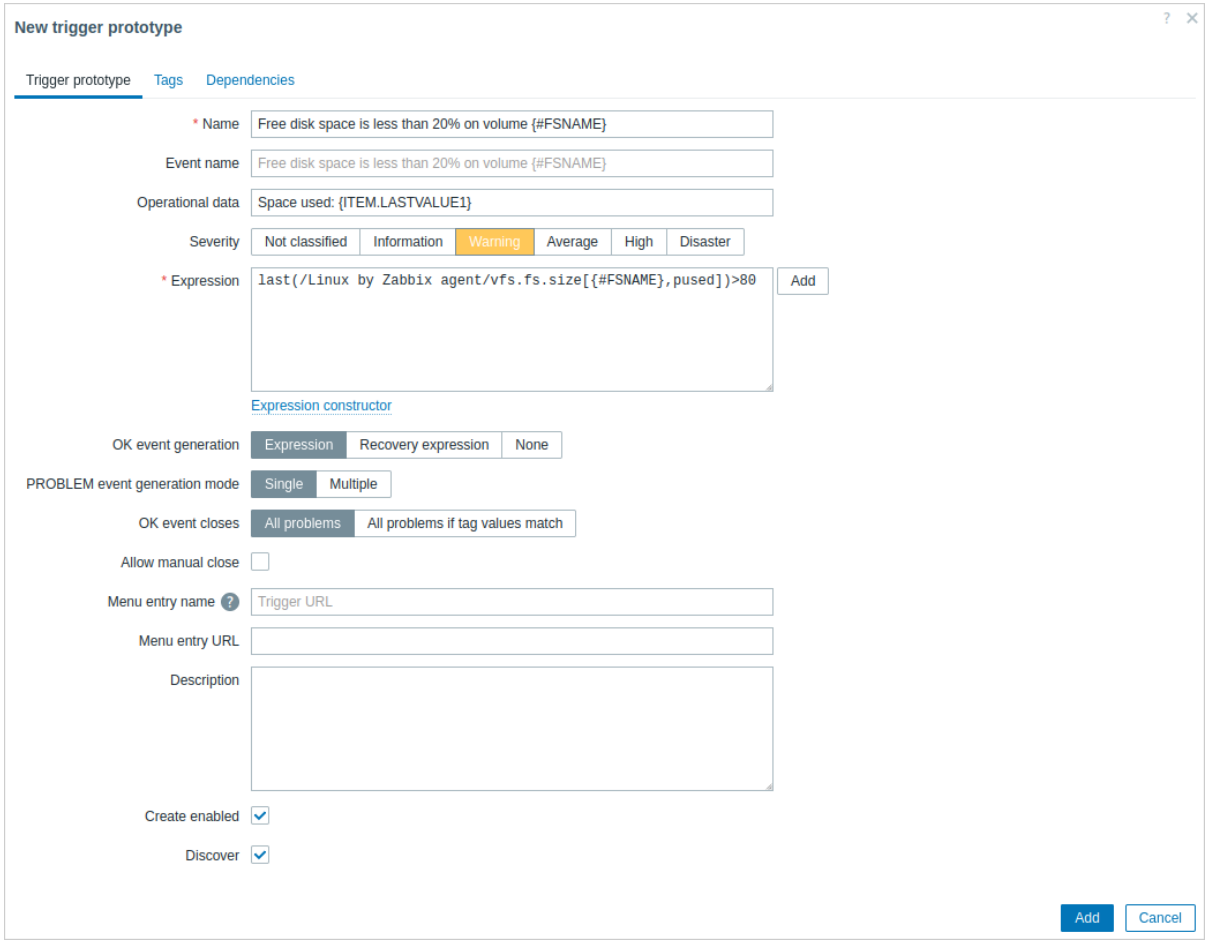
Cliquez sur l'icône représentant trois points pour ouvrir le menu du prototype d'élément concerné, qui propose les options suivantes :

- *Créer un prototype de déclencheur* - créer un prototype de déclencheur basé sur ce prototype d'élément
- *Prototypes de déclencheurs* - cliquer pour afficher une liste contenant des liens vers les prototypes de déclencheurs déjà configurés pour ce prototype d'élément
- *Créer un élément dépendant* - créer un élément dépendant pour ce prototype d'élément

L'option *Mise à jour en masse* est disponible si vous souhaitez mettre à jour simultanément les propriétés de plusieurs prototypes d'éléments.

Prototypes de déclencheur

Nous créons les prototypes de déclencheur de la même manière que les prototypes d'élément:

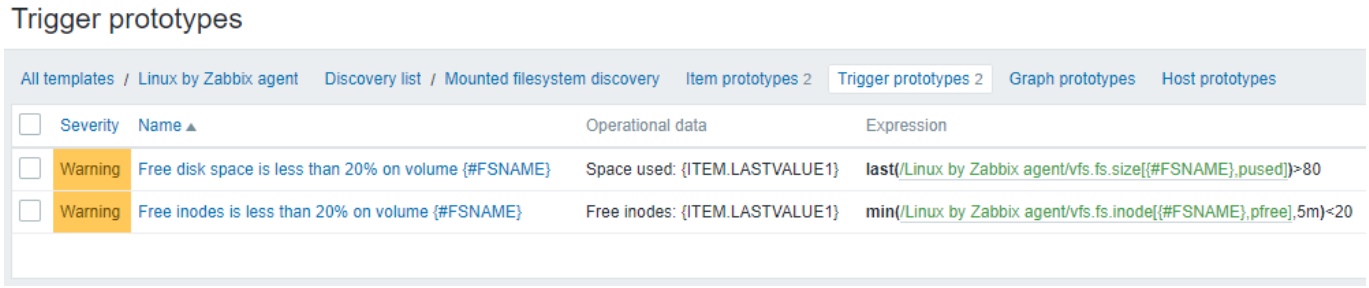


Attributs spécifiques aux prototypes de déclencheur:

Parameter	Description
Create enabled	Si cette option est cochée, le déclencheur sera ajouté dans un état activé. Si elle n'est pas cochée, le déclencheur sera ajouté à une entité découverte, mais dans un état désactivé.
Discover	Si cette option est cochée (par défaut), le déclencheur sera ajouté à une entité découverte. Si elle n'est pas cochée, le déclencheur ne sera pas ajouté à une entité découverte, sauf si ce paramètre est remplacé dans la règle de découverte.

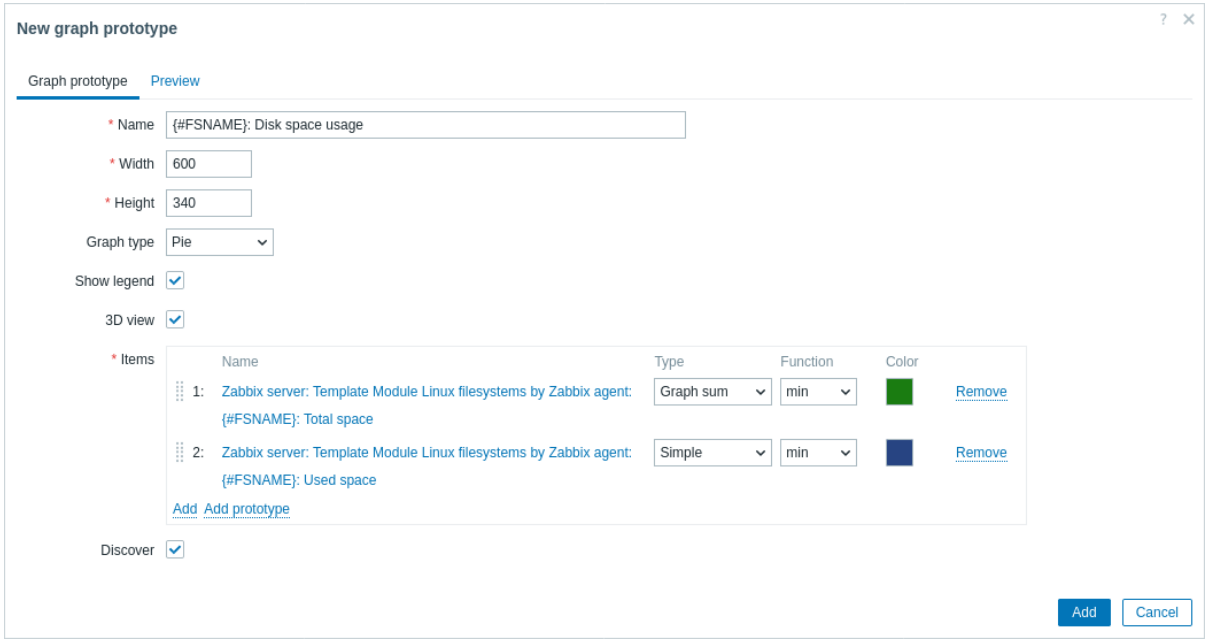
Lorsque de vrais déclencheurs sont créés à partir des prototypes, il peut être nécessaire d'être flexible quant à la constante utilisée ('20' dans notre exemple) pour la comparaison dans l'expression. Voir comment les macros utilisateur avec contexte peuvent être utiles pour obtenir une telle flexibilité.

Vous pouvez définir des dépendances entre des prototypes de déclencheur. Pour cela, allez dans l'onglet *Dependencies*. Un prototype de déclencheur peut dépendre d'un autre prototype de déclencheur issu de la même règle de découverte de bas niveau (LLD) ou d'un déclencheur normal. Un prototype de déclencheur ne peut pas dépendre d'un prototype de déclencheur provenant d'une autre règle LLD ni d'un déclencheur créé à partir d'un prototype de déclencheur. Un prototype de déclencheur d'hôte ne peut pas dépendre d'un déclencheur d'un modèle.



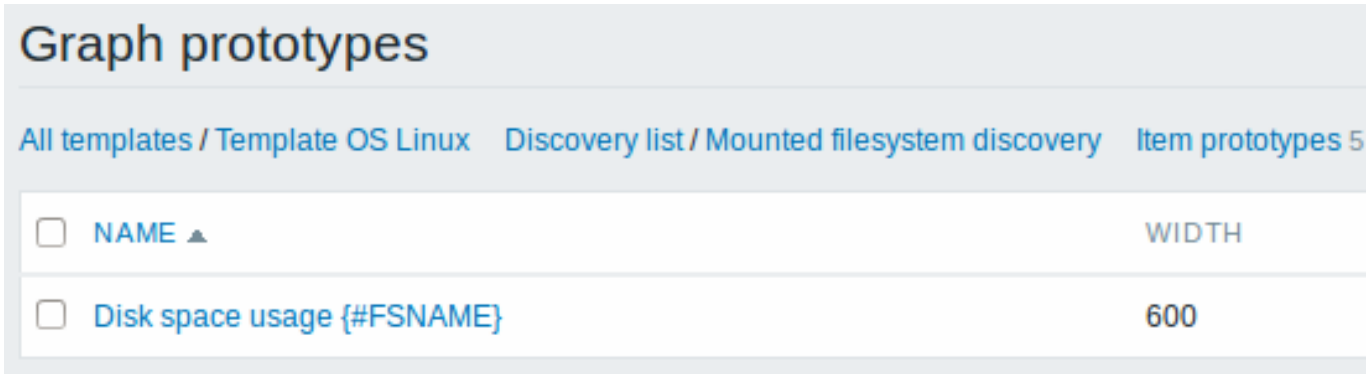
Modèles de graphique

Nous pouvons également créer des modèles de graphique:

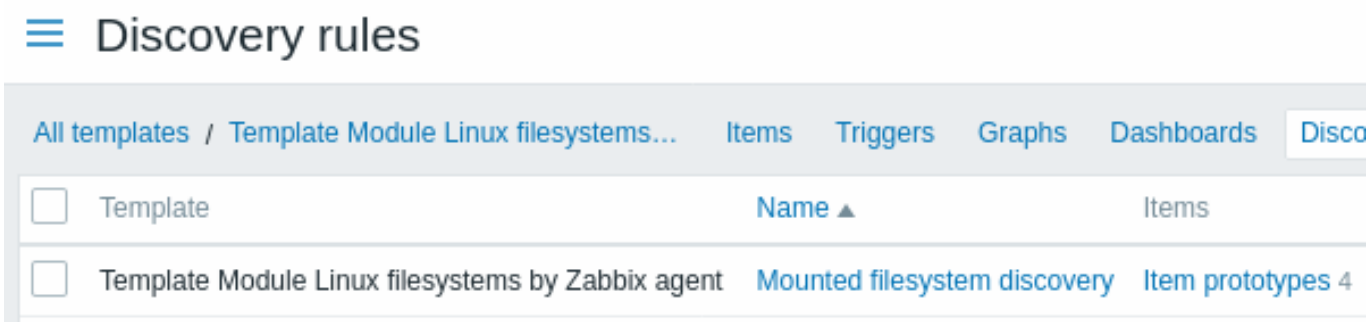


Attributs spécifiques aux modèles de graphique:

Parameter	Description
Discover	Si cette option est cochée (par défaut), le graphique sera ajouté à une entité découverte. Si elle n'est pas cochée, le graphique ne sera pas ajouté à une entité découverte, sauf si ce paramètre est remplacé dans la règle de découverte.



Enfin, nous avons créé une règle de découverte qui ressemble à ce qui est montré ci-dessous. Elle comporte cinq modèles d'élément, deux modèles de déclencheur et un modèle de graphique.



Prototypes d'hôte

Les prototypes d'hôte sont des modèles servant à créer des hôtes via des règles de découverte de bas niveau. Avant d'être découverts en tant qu'hôtes, ces prototypes ne peuvent pas avoir d'éléments de données ni de déclencheurs, à l'exception de ceux liés à partir de modèles.

Configuration

Les prototypes d'hôte sont configurés sous les règles de découverte de bas niveau.

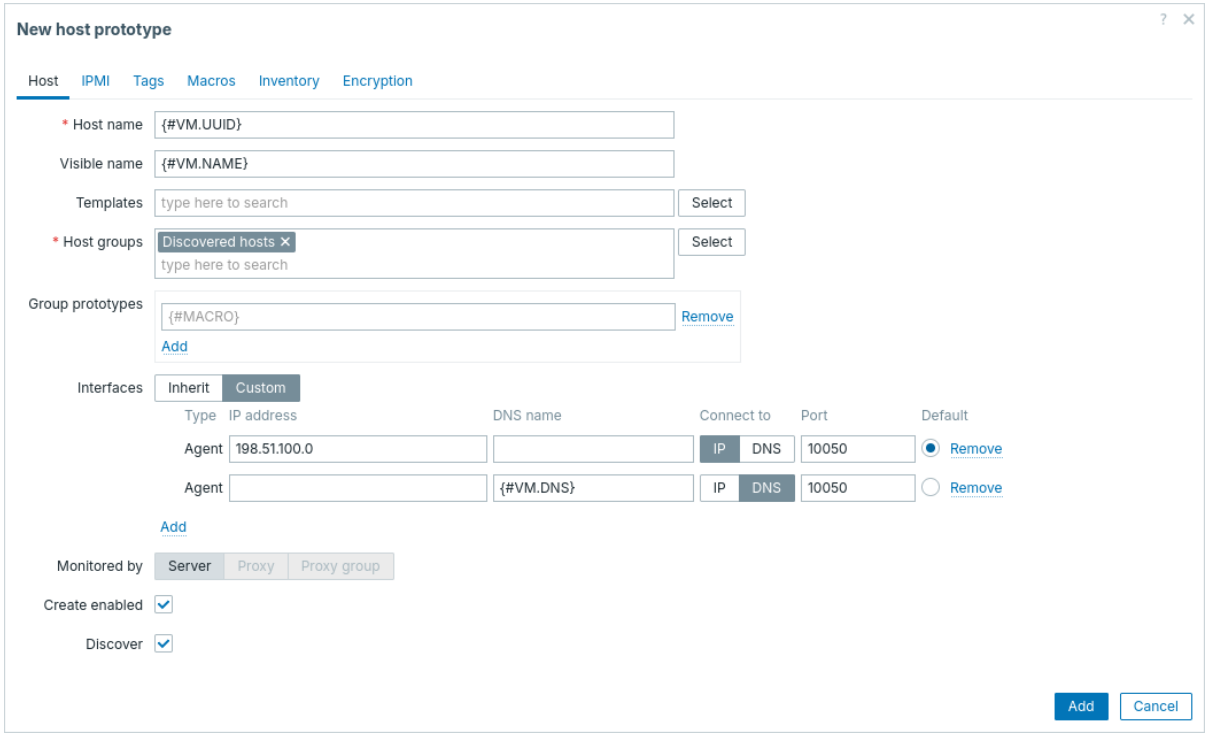
Pour créer un prototype d'hôte :

1. Accédez à *Collecte de données > Hôtes*.

2. Cliquez sur *Découverte* pour l'hôte requis afin d'accéder à la liste des règles de découverte de bas niveau configurées pour cet hôte.

3. Cliquez sur *Prototypes d'hôte* pour la règle de découverte requise.

4. Cliquez sur le bouton *Créer un prototype d'hôte* dans le coin supérieur droit et complétez le formulaire de configuration.



Tous les champs obligatoires sont marqués d'un astérisque rouge.

Parameter	Description
Host name	Ce paramètre doit contenir au moins une macro de découverte de bas niveau afin de garantir des noms d'hôte uniques pour les hôtes créés.
Visible name	Les macros de découverte de bas niveau sont prises en charge.
Group prototypes	Permet de spécifier des prototypes de groupes d'hôtes à l'aide de macros de découverte de bas niveau. Sur la base des prototypes de groupes spécifiés, des groupes d'hôtes seront découverts, créés et liés aux hôtes créés ; les groupes découverts qui ont déjà été créés par d'autres règles de découverte de bas niveau seront également liés aux hôtes créés. Toutefois, les groupes d'hôtes découverts qui correspondent à des groupes d'hôtes créés manuellement ne seront pas liés aux hôtes créés.
Interfaces	Définissez si les hôtes découverts héritent de l'adresse IP de l'hôte auquel appartient la règle de découverte (par défaut), ou s'ils obtiennent des interfaces personnalisées. Les macros de découverte de bas niveau et les macros utilisateur sont prises en charge.
Monitored by	Ce champ est synchronisé automatiquement avec le champ correspondant dans la configuration de l'hôte.
Create enabled	Définissez l'état des hôtes découverts ; si cette option n'est pas cochée, les hôtes seront créés comme désactivés.
Discover	Définissez si des hôtes seront créés à partir du prototype d'hôte ; si cette option n'est pas cochée, les hôtes ne seront pas créés à partir du prototype d'hôte (sauf si ce paramètre est remplacé dans la règle de découverte de bas niveau).

Note:
Les macros de découverte de bas niveau sont également prises en charge pour les valeurs de balises et les valeurs de macros utilisateur du prototype d'hôte.
 Les *mappages de valeurs* ne sont pas pris en charge pour les prototypes d'hôte.

Pour un exemple de configuration d'un prototype d'hôte, consultez *Surveillance de machine virtuelle*.

Interfaces d'hôte

Pour ajouter des interfaces personnalisées, basculez le sélecteur *Interfaces* de « Inherit » à « Custom ». Cliquez sur Add et sélectionnez le type d'interface : agent Zabbix, SNMP, JMX, IPMI.

Note:
Si *Custom* est sélectionné, mais qu'aucune interface n'a été définie, les hôtes seront créés sans interfaces.
 Si *Inherit* est sélectionné et que le prototype d'hôte appartient à un modèle, tous les hôtes découverts hériteront de l'interface d'hôte de l'hôte auquel le modèle est lié.

Si plusieurs interfaces personnalisées sont spécifiées, l'interface principale peut être définie dans la colonne *Default*.

Pour un exemple de configuration d'une interface d'hôte personnalisée, voir *Exemple de configuration de la supervision VMware*.

Warning:
Un hôte ne sera créé que si une interface d'hôte contient des données correctes.

Hôtes découverts

Dans la liste des hôtes, les hôtes découverts sont préfixés par le nom de la règle de découverte qui les a créés.

Les hôtes découverts héritent de la plupart des paramètres des prototypes d'hôte en *lecture seule*. Seuls les paramètres d'hôte découvert suivants peuvent être configurés :

- *Templates* - lier de nouveaux modèles ou délier ceux ajoutés manuellement. Les modèles hérités d'un prototype d'hôte ne peuvent pas être déliés.
- *Description* - ajouter ou modifier la description de l'hôte.
- *Status* - activer ou désactiver l'hôte.
- *Tags* - ajouter de nouveaux tags ou supprimer ceux ajoutés manuellement. Les tags hérités d'un prototype d'hôte ne peuvent pas être supprimés. Notez que les tags ne peuvent pas avoir de doublons (tags ayant le même nom et la même valeur). Si un nouveau tag est ajouté au prototype d'hôte et qu'il correspond à un tag ajouté manuellement sur un hôte découvert, le tag ajouté manuellement sera remplacé par le tag hérité lors de la découverte.
- *Macros* - ajouter de nouvelles macros d'hôte ou supprimer celles ajoutées manuellement, ainsi que celles héritées d'un prototype d'hôte ; modifier les valeurs des macros et les types.

Note:
Les hôtes découverts héritent également des macros utilisateur de l'hôte sur lequel la règle de découverte est configurée. Ces macros héritées peuvent être supprimées, et leurs valeurs et types peuvent être modifiés.

Les hôtes découverts peuvent également découvrir d'autres hôtes. Par exemple, si un prototype d'hôte (utilisé pour découvrir des hyperviseurs) a un modèle lié qui inclut une règle de découverte de bas niveau avec son propre prototype d'hôte (utilisé pour découvrir des machines virtuelles), Zabbix découvrira à la fois les hyperviseurs et leurs machines virtuelles. Pour découvrir en plus des conteneurs sur ces VM, vous pouvez créer des règles de découverte de bas niveau ou lier un nouveau modèle avec des prototypes d'hôte aux VM découvertes, ou préconfigurer le prototype d'hôte de la VM en liant un modèle qui inclut lui-même des prototypes d'hôte.

Les hôtes découverts peuvent être supprimés manuellement. Notez toutefois qu'ils seront découverts à nouveau si la découverte est activée pour eux.

Les hôtes qui ne sont plus découverts peuvent être :

- automatiquement désactivés (selon la valeur de *Disable lost resources* de la règle de découverte)
- automatiquement supprimés (selon la valeur de *Delete lost resources* de la règle de découverte).

Prototypes de découverte

Vue d'ensemble

Les prototypes de découverte sont des règles de **découverte de bas niveau imbriquées** au sein d'une règle de découverte « parente », permettant une découverte à plusieurs niveaux d'objets avec leurs propres éléments, déclencheurs, etc. Par exemple, vous pouvez souhaiter découvrir toutes les instances de bases de données sur un serveur de bases de données, puis découvrir les espaces de tables de chaque instance, et enfin découvrir les tables de chaque espace de tables.

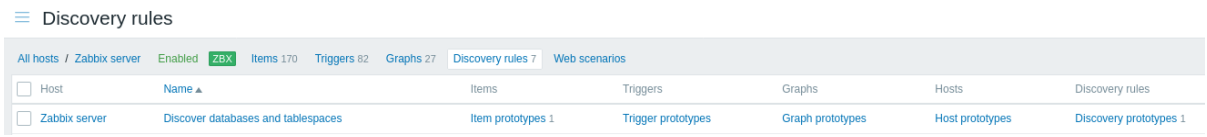
Les prototypes de découverte disposent de leurs propres prototypes d'élément, de déclencheur, de graphique, d'hôte et de découverte. Un prototype de découverte imbriqué utilise la même valeur JSON que la règle parente si vous spécifiez un type *Nested*.

Le nombre de niveaux d'imbrication des prototypes de découverte est illimité.

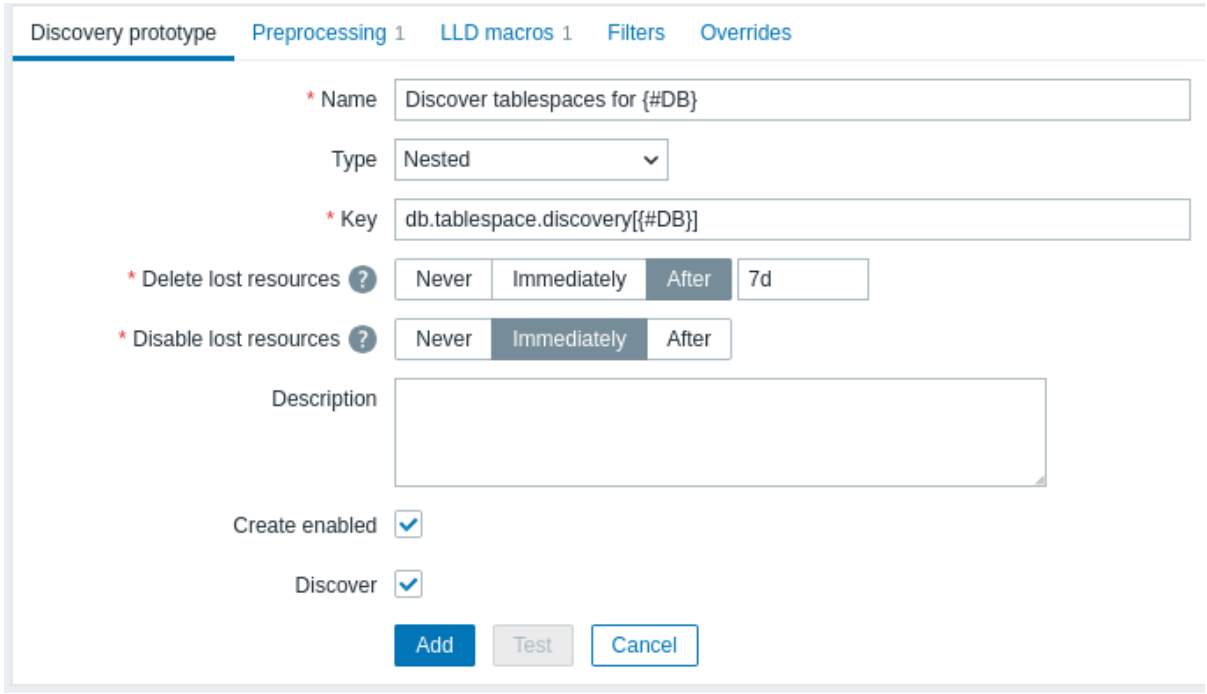
Configuration

Pour créer un prototype de découverte :

- Cliquez sur *Discovery prototypes* dans la ligne d'une règle de découverte existante



- Cliquez sur *Create discovery prototype*



Les champs de configuration de ce formulaire sont partagés avec la découverte de bas niveau standard.

Si vous sélectionnez "Nested" comme *Type* dans le formulaire du prototype de découverte ouvert, alors des règles de découverte (à partir du prototype de découverte) sont générées sur la base d'un objet JSON provenant de la même valeur JSON que la règle de découverte parente. Par exemple, si le JSON d'origine est [<object A>, <object B>] et qu'il existe un prototype de règle de découverte imbriquée, alors deux règles de découverte seront générées sur la base des données de l'objet A et de l'objet B, respectivement.

Dans ce cas, le prototype de découverte est activé en même temps que la règle parente. La règle imbriquée peut donc utiliser le prétraitement pour travailler sur une autre "tranche" des mêmes données, déjà acquises par la parente.

Les macros LLD de la règle LLD parente sont disponibles pour les règles de découverte imbriquées.

Règles LLD imbriquées sur les hôtes découverts

Une règle de découverte de bas niveau *Nested* peut être utilisée sur un modèle d'hôte attribué à un prototype d'hôte. Si une règle de découverte *Nested* existe sur un hôte découvert, alors l'objet JSON utilisé pour découvrir l'hôte est également envoyé à toutes les règles LLD de type imbriqué sur cet hôte. Pour plus de détails, voir l'exemple.

Les macros LLD de la règle de découverte qui a créé l'hôte sont disponibles pour les règles de découverte imbriquées.

Exemple

Illustrons une application possible des prototypes de découverte, à partir de la réception de l'exemple suivant de JSON à plusieurs niveaux.

```
[
  {
    "database": "db1",
    "created_at": "2024-02-01T12:30:00Z",
    "encoding": "UTF8",
    "tablespaces": [
      { "name": "ts1", "max_size": "10GB" },
      { "name": "ts2", "max_size": "20GB" },
      { "name": "ts3", "max_size": "15GB" }
    ]
  },
  {
    "database": "db2",
    "created_at": "2023-11-15T08:45:00Z",
    "encoding": "UTF16",
    "tablespaces": [
      { "name": "ts1", "max_size": "5GB" },
      { "name": "ts2", "max_size": "25GB" },
      { "name": "ts3", "max_size": "30GB" }
    ]
  },
  {
    "database": "db3",
    "created_at": "2024-01-05T15:10:00Z",
    "encoding": "UTF8",
    "tablespaces": [
      { "name": "ts1", "max_size": "12GB" },
      { "name": "ts2", "max_size": "18GB" },
      { "name": "ts3", "max_size": "22GB" }
    ]
  }
]
```


Cas 1

Découverte des instances de base de données sur un serveur de base de données, puis découverte des tablespaces pour chaque instance.

1. Vous avez au moins un hôte lié à la découverte du serveur de base de données.
2. Créez une règle LLD pour cet hôte nommée *Découvrir les bases de données et les tablespaces*.
3. Passez à l'onglet *Macros LLD* de cette règle, puis ajoutez la macro `{#DB}=$.database`.
4. Ajoutez un prototype d'élément pour cette règle nommé *Connexions actives à {#DB}* (Type : Agent, Clé : `db.connections[{#DB}]`).
5. Les éléments liés à chaque base de données sont découverts :

```
Active connections to db1, Active connections to db2, Active connections to db3.
```

6. Créez un prototype de découverte pour cette règle nommé *Découvrir les tablespaces pour {#DB}* (Type : Nested, Clé : `db.tablespace.discovery[{#DB}]`).
7. Passez à l'onglet *Prétraitement* de ce prototype de découverte et ajoutez l'étape `JSONPath=$.tablespaces`.
8. Passez à l'onglet *Macros LLD* de ce prototype de découverte, puis ajoutez la macro `{#TSNAME}=$.name`.
9. Créez un prototype d'élément pour ce prototype de découverte nommé *Taille du tablespace {#TSNAME} pour {#DB}* (Type : Agent, Clé : `db.ts.size[{#DB}, {#TSNAME}]`).
10. Les éléments liés à chaque tablespace de chaque base de données sont découverts :

```
Size of tablespace ts1 for db1, Size of tablespace ts2 for db1, Size of tablespace ts3 for db1,
Size of tablespace ts1 for db2, Size of tablespace ts2 for db2, Size of tablespace ts3 for db2,
Size of tablespace ts1 for db3, Size of tablespace ts2 for db3, Size of tablespace ts3 for db3.
```

avec les clés `db.ts.size[db1,ts1]`, `db.ts.size[db1,ts2]`, ... `db.ts.size[db3,ts3]`.

Cas 2

Découverte des instances de base de données sur le serveur de base de données en les représentant comme des hôtes découverts, puis découverte des tablespaces pour chaque instance.

1. Vous disposez d'au moins un hôte (hôte racine) lié à la découverte du serveur de base de données.
2. Créez un modèle pour découvrir les tablespaces de chaque base de données.
3. Créez un élément dans ce modèle nommé *Connexions actives à {#DB}* (Type : Agent, clé : `db.connections[{#DB}]`).
4. Créez une règle LLD pour ce modèle nommée *Découvrir les tablespaces* (Type : imbriqué).
5. Passez à l'onglet *Preprocessing* de cette règle et ajoutez l'étape `JSONPath=$.tablespaces`.
6. Passez à l'onglet *Macros LLD* de cette règle, ajoutez la macro `{#TSNAME}=$.name`.
7. Créez un prototype d'élément pour cette règle nommé *Taille du tablespace {#TSNAME} pour {#DB}* (Type : Agent, clé : `db.ts.size[{#DB}, {#TSNAME}]`).
8. De retour sur l'hôte racine, créez une règle LLD pour cet hôte nommée *Découvrir les bases de données et les tablespaces*.
9. Passez à l'onglet *Macros LLD* de cette règle, ajoutez la macro `{#DB}=$.database`.
10. Ajoutez un prototype d'hôte pour cette règle nommé *Hôte pour la base de données {#DB}*.
11. Passez à l'onglet *Macros* de ce prototype d'hôte, ajoutez la macro `{$DB}={#DB}` (pour le nom et la clé de l'élément de l'étape 3).
12. Liez le modèle de l'étape 2 à ce prototype d'hôte.
13. Les hôtes découverts contiennent les éléments découverts liés à chaque base de données et à ses tablespaces :

Hôte	Éléments
Hôte pour la base de données db1	Connexions actives à db1
	Taille du tablespace ts1 pour db1
	Taille du tablespace ts2 pour db1
	Taille du tablespace ts3 pour db1
Hôte pour la base de données db2	Connexions actives à db2
	Taille du tablespace ts1 pour db2
	Taille du tablespace ts2 pour db2
	Taille du tablespace ts3 pour db2
Hôte pour la base de données db3	Connexions actives à db3
	Taille du tablespace ts1 pour db3
	Taille du tablespace ts2 pour db3
	Taille du tablespace ts3 pour db3

Notes sur la découverte de bas niveau

Utilisation des macros LLD dans les contextes de macros utilisateur

Les macros LLD peuvent être utilisées dans le contexte des macros utilisateur, par exemple, dans les prototypes de déclencheur.

Plusieurs règles de découverte de bas niveau pour le même élément

Il est possible de définir plusieurs règles de découverte de bas niveau avec le même élément de découverte.

Pour ce faire, vous devez définir le paramètre Alias de l'agent, ce qui vous permet d'utiliser des clés d'élément de découverte modifiées dans différentes règles de découverte, par exemple `vfs.fs.discovery[foo]`, `vfs.fs.discovery[bar]`, etc.

Limites de données pour les valeurs de retour

Il n'y a pas de limite pour les données JSON des règles de découverte de bas niveau si elles sont reçues directement par le serveur Zabbix. Cela s'explique par le fait que les valeurs de retour sont traitées sans être stockées dans une base de données.

Il n'y a pas non plus de limite pour les règles de découverte de bas niveau personnalisées. Cependant, si les données d'une règle de découverte de bas niveau personnalisée sont récupérées à l'aide d'un paramètre utilisateur, la limite de valeur de retour du paramètre utilisateur s'applique.

Si les données doivent transiter par un proxy Zabbix, celui-ci doit stocker ces données dans la base de données. Dans ce cas, les limites de la base de données s'appliquent.

Exemples de règles de découverte

Cette section fournit des exemples pratiques de configuration de règles de découverte pour différents types d'entités, telles que les systèmes de fichiers, les interfaces réseau, les services, les capteurs, les objets SNMP, les applications Java et les bases de données.

Chaque exemple illustre la manière dont une règle de découverte peut être structurée, y compris la clé d'élément ou la requête utilisée, la sortie JSON avec les macros LLD, ainsi que la façon dont ces résultats sont utilisés pour créer des prototypes d'élément, de déclencheur et de graphique.

- Découverte des périphériques de bloc
- Découverte des CPU et des cœurs CPU
- Découverte des interfaces de l'hôte dans Zabbix
- Découverte des capteurs IPMI
- Découverte des objets JMX
- Découverte des systèmes de fichiers montés
- Découverte des interfaces réseau
- Découverte des OID SNMP
- Découverte des OID SNMP (héritée)
- Découverte des services systemd
- Découverte des instances de compteurs de performance Windows
- Découverte des services Windows
- Découverte à l'aide de requêtes WMI
- Découverte à l'aide de requêtes SQL ODBC
- Découverte à l'aide de données Prometheus

Règles LLD personnalisées

Aperçu

Il est également possible de créer une règle LLD entièrement personnalisée, permettant de découvrir tout type d'entités, par exemple des bases de données sur un serveur de bases de données.

Pour ce faire, un élément personnalisé doit être créé afin de renvoyer une chaîne JSON, en spécifiant les objets trouvés et, éventuellement, certaines de leurs propriétés. Le nombre de macros par entité n'est pas limité : alors que les règles de découverte intégrées renvoient une ou deux macros (par exemple, deux pour la découverte des systèmes de fichiers), il est possible d'en renvoyer davantage.

Exemple

Le format requis de chaîne JSON est mieux illustré par un exemple. Supposons que vous exécutiez un ancien agent Zabbix 1.8 (qui ne prend pas en charge la clé `vfs.fs.discovery`), mais que vous ayez tout de même besoin de découvrir les systèmes de fichiers. Voici un script Perl simple pour Linux qui découvre les systèmes de fichiers montés et produit une sortie au format JSON, incluant à la fois le nom et le type du système de fichiers. Une façon de l'utiliser serait comme UserParameter avec la clé `vfs.fs.discovery_perl` :

```
####!/usr/bin/perl

$first = 1;

print "[\n";

for (`cat /proc/mounts`)
{
    ($fsname, $fstype) = m/\S+ (\S+) (\S+)/;

    print "\t,\n" if not $first;
    $first = 0;

    print "\t{\n";
    print "\t\t\"{#FSNAME}\":\"$fsname\",\n";
    print "\t\t\"{#FSTYPE}\":\"$fstype\"\n";
    print "\t}\n";
}
```

```
print "]\n";
```

> **Attention:**
> Les symboles autorisés pour les noms de macros LLD sont **0-9** , **A-Z** , **_** , **.**. Les lettres minuscules ne sont pas prises en charge dans les noms.

Un exemple de sa sortie (reformaté pour plus de clarté) est présenté ci-dessous. Le JSON pour les vérifications de découverte personnalisées doit suivre le même format.

```
[
    { "{#FSNAME}":"/",                          "{#FSTYPE}":"rootfs"   },
    { "{#FSNAME}":"/sys",                       "{#FSTYPE}":"sysfs"    },
    { "{#FSNAME}":"/proc",                      "{#FSTYPE}":"proc"     },
    { "{#FSNAME}":"/dev",                       "{#FSTYPE}":"devtmpfs" },
    { "{#FSNAME}":"/dev/pts",                   "{#FSTYPE}":"devpts"   },
    { "{#FSNAME}":"/lib/init/rw",               "{#FSTYPE}":"tmpfs"    },
    { "{#FSNAME}":"/dev/shm",                   "{#FSTYPE}":"tmpfs"    },
    { "{#FSNAME}":"/home",                      "{#FSTYPE}":"ext3"     },
    { "{#FSNAME}":"/tmp",                       "{#FSTYPE}":"ext3"     },
    { "{#FSNAME}":"/usr",                       "{#FSTYPE}":"ext3"     },
    { "{#FSNAME}":"/var",                       "{#FSTYPE}":"ext3"     },
    { "{#FSNAME}":"/sys/fs/fuse/connections",   "{#FSTYPE}":"fusectl"  }
]
```

Dans l'exemple précédent, il est nécessaire que les clés correspondent aux noms de macros LLD utilisés dans les prototypes ; l'alternative consiste à extraire les valeurs des macros LLD à l'aide de JSONPath `{#FSNAME}` → `$.fsname` et `{#FSTYPE}` → `$.fstype`, rendant ainsi possible un script comme celui-ci :

```
####!/usr/bin/perl

$first = 1;

print "[\n";

for (`cat /proc/mounts`)
{
    ($fsname, $fstype) = m/\S+ (\S+) (\S+)/;

    print "\t,\n" if not $first;
    $first = 0;

    print "\t{\n";
    print "\t\t\"fsname\":\"$fsname\",\n";
    print "\t\t\"fstype\":\"$fstype\"\n";
    print "\t}\n";
}

print "]\n";
```

Un exemple de sa sortie (reformaté pour plus de clarté) est présenté ci-dessous.

```
[
    { "fsname":"/",                         "fstype":"rootfs"   },
    { "fsname":"/sys",                      "fstype":"sysfs"    },
    { "fsname":"/proc",                     "fstype":"proc"     },
    { "fsname":"/dev",                      "fstype":"devtmpfs" },
    { "fsname":"/dev/pts",                  "fstype":"devpts"   },
    { "fsname":"/lib/init/rw",              "fstype":"tmpfs"    },
    { "fsname":"/dev/shm",                  "fstype":"tmpfs"    },
    { "fsname":"/home",                     "fstype":"ext3"     },
    { "fsname":"/tmp",                      "fstype":"ext3"     },
    { "fsname":"/usr",                      "fstype":"ext3"     },
    { "fsname":"/var",                      "fstype":"ext3"     },
    { "fsname":"/sys/fs/fuse/connections",  "fstype":"fusectl"  }
```

```
]
```

Ensuite, dans le champ *Filter* de la règle de découverte, vous pouvez spécifier `{#FSTYPE}` comme macro et `rootfs|ext3` comme expression régulière.

Note:
Vous n'êtes pas obligé d'utiliser des noms de macros comme `FSNAME`/`FSTYPE` avec des règles LLD personnalisées ; vous êtes libre d'utiliser les noms de votre choix. Si JSONPath est utilisé, alors la ligne LLD sera un élément de tableau qui peut être un objet, mais aussi un autre tableau ou une valeur.

Notez que, si vous utilisez un paramètre utilisateur, la valeur de retour est limitée à 16 Mo. Pour plus de détails, voir les limites de données pour les valeurs de retour LLD.

Surveillance des machines virtuelles

Aperçu Zabbix peut utiliser des règles de découverte de bas niveau pour découvrir automatiquement les hyperviseurs VMware et les machines virtuelles, et créer des hôtes pour les surveiller sur la base de prototypes d'hôte prédéfinis.

Zabbix inclut également des modèles prêts à l'emploi pour la supervision de VMware vCenter ou des hyperviseurs ESXi.

La version minimale requise de VMware vCenter ou vSphere est la 5.1.

Collecte de données La surveillance des machines virtuelles se compose de deux étapes :

1. Les processus Zabbix *vmware collector* traitent les données des machines virtuelles : ils obtiennent les informations nécessaires à partir des services web VMware via le protocole SOAP, les prétraitent et les stockent dans la mémoire partagée du serveur Zabbix.
2. Les processus Zabbix *poller* récupèrent les données à l'aide de la vérification simple Zabbix clés d'éléments de surveillance VMware.

Zabbix divise les données collectées en données de configuration VMware et en données de compteurs de performance VMware. Les deux types de données sont collectés indépendamment par les processus *vmware collector*.

Les statistiques suivantes sont disponibles sur la base des informations des compteurs de performance VMware :

- Banque de données
- Périphérique de disque
- Processeur
- Alimentation
- Interface réseau
- Éléments de compteur de performance personnalisés

Pour la liste complète des éléments qui obtiennent des données à partir des compteurs de performance VMware, voir *clés d'éléments de surveillance VMware*.

Notez que la fréquence de récupération des événements VMware dépend de l'intervalle d'interrogation de vmware.eventlog, mais ne peut pas être inférieure à 5 secondes.

Configuration Si le serveur Zabbix est compilé à partir des sources, il doit être compilé avec les options de configuration `--with-libcurl --with-libxml2` pour activer la surveillance des machines virtuelles. Les paquets Zabbix sont compilés avec ces options déjà activées.

Les paramètres suivants du fichier de configuration du serveur Zabbix peuvent être modifiés pour la surveillance des machines virtuelles :

- `StartVMwareCollectors`

Note:
Il est recommandé d'activer un nombre de collecteurs supérieur au nombre de services VMware surveillés ; sinon, la récupération des statistiques des compteurs de performance VMware peut être retardée par la récupération des données de configuration VMware (ce qui prend un certain temps pour les grandes installations).

 En général, la valeur de `StartVMwareCollectors` ne doit pas être inférieure à 2 et ne doit pas dépasser deux fois le nombre de services VMware surveillés : Nombre de services < `StartVMwareCollectors` < (Nombre de services * 2). Par exemple, lors de la surveillance d'un service VMware, définissez `StartVMwareCollectors` sur 2 ; lors de la surveillance de trois services, définissez `StartVMwareCollectors` sur 5.

 Notez que le nombre requis de collecteurs dépend également de l'étendue de l'environnement VMware, ainsi que des paramètres de configuration `VMwareFrequency` et `VMwarePerfFrequency`.

- `VMwareCacheSize`
- `VMwareFrequency`
- `VMwarePerfFrequency`
- `VMwareTimeout`

Attention:
Pour prendre en charge les métriques de capacité des datastores, assurez-vous que la valeur de la clé VMware `vpxd.stats.maxQueryMetrics` est définie sur au moins 64. Pour plus d'informations, consultez l'*article de la base de connaissances VMware*.

Découverte

Zabbix peut utiliser des règles de découverte de bas niveau (par exemple, `vmware.hv.discovery[{$VMWARE.URL}]`) pour découvrir automatiquement les hyperviseurs VMware et les machines virtuelles. De plus, Zabbix peut utiliser des prototypes d'hôte pour générer automatiquement des hôtes réels pour les entités découvertes. Pour plus d'informations, consultez *Prototypes d'hôte*.

Exemples de configuration

Pour un exemple de base montrant comment configurer Zabbix pour surveiller VMware à l'aide du modèle *VMware FQDN*, consultez *Surveiller VMware avec Zabbix*.

Pour un exemple plus détaillé montrant comment créer un hôte, une règle de découverte de bas niveau et un prototype d'hôte pour la surveillance de VMware, consultez *Exemple de configuration*.

Journalisation étendue Les données collectées par les processus *vmware collector* peuvent être journalisées pour un débogage détaillé à l'aide du niveau de débogage 5. Le niveau de débogage peut être configuré dans les fichiers de configuration du serveur et du proxy, ou à l'aide de l'option de contrôle à l'exécution `-R log_level_increase="vmware collector,N"`, où `N` correspond au numéro du processus.

Par exemple, pour augmenter le niveau de débogage de 4 à 5 pour tous les processus *vmware collector*, exécutez la commande suivante :

```
zabbix_server -R log_level_increase="vmware collector"
```

Pour augmenter le niveau de débogage de 4 à 5 pour le deuxième processus *vmware collector*, exécutez la commande suivante :

```
zabbix_server -R log_level_increase="vmware collector,2"
```

Lorsque la journalisation étendue des données du collecteur VMware n'est plus nécessaire, il est recommandé de réduire le niveau de débogage à sa valeur par défaut (3) en exécutant la commande `-R log_level_decrease`.

Dépannage

- En cas de métriques indisponibles, assurez-vous qu'elles n'ont pas été rendues indisponibles ou désactivées par défaut dans les versions récentes de VMware vSphere, ou que certaines limites ne sont pas appliquées aux requêtes de base de données des métriques de performance. Pour plus d'informations, consultez ZBX-12094.
- Si `config.vpxd.stats.maxQueryMetrics` est invalide ou dépasse le nombre maximal de caractères autorisé, ajoutez un paramètre `config.vpxd.stats.maxQueryMetrics` aux paramètres de vCenter Server. La valeur de ce paramètre doit être identique à la valeur de `maxQuerysize` dans le fichier `web.xml` de VMware. Pour plus d'informations, consultez l'article *VMware Knowledge Base*.
- Si vous pensez que votre installation Zabbix utilise trop de mémoire, consultez Profiling excessive memory usage with tcmalloc.

Clés d’élément de surveillance VMware

Aperçu Cette page fournit des détails sur les vérifications simples qui peuvent être utilisées pour surveiller les environnements VMware.
Les métriques sont regroupées par cible de surveillance.

Clés d’élément prises en charge Les clés d’élément sont répertoriées sans paramètres ni informations supplémentaires. Cliquez sur la clé d’élément pour voir les détails complets.

Clé d’élément	Description	Groupe d’éléments
vmware.eventlog	Le journal des événements VMware.	Service général
vmware.fullname	Le nom complet du service VMware.	
vmware.version	La version du service VMware.	
vmware.cl.perfcounter	Les métriques du compteur de performance du cluster VMware.	Cluster
vmware.cluster.alarms.get	Les données des alarmes du cluster VMware.	
vmware.cluster.discovery	La découverte des clusters VMware.	
vmware.cluster.property	La propriété du cluster VMware.	
vmware.cluster.status	L’état du cluster VMware.	
vmware.cluster.tags.get	Le tableau des balises du cluster VMware.	
vmware.datastore.alarms.get	Les données des alarmes du datastore VMware.	Datastore
vmware.datastore.discovery	La découverte des datastores VMware.	
vmware.datastore.hv.list	La liste des hyperviseurs du datastore.	
vmware.datastore.perfcounter	La valeur du compteur de performance du datastore VMware.	
vmware.datastore.property	La propriété du datastore VMware.	
vmware.datastore.read	Le temps nécessaire à une opération de lecture depuis le datastore.	
vmware.datastore.size	L’espace du datastore VMware en octets ou en pourcentage du total.	
vmware.datastore.tags.get	Le tableau des balises du datastore VMware.	
vmware.datastore.write	Le temps nécessaire à une opération d’écriture vers le datastore.	
vmware.dc.alarms.get	Les données des alarmes du centre de données VMware.	Centre de données
vmware.dc.discovery	La découverte des centres de données VMware.	
vmware.dc.tags.get	Le tableau des balises du centre de données VMware.	
vmware.dvswitch.discovery	La découverte des commutateurs distribués vSphere VMware.	vSphere Distributed Switch
vmware.dvswitch.fetchports.get	Les données des ports du commutateur distribué vSphere VMware.	
vmware.hv.alarms.get	Les données des alarmes de l’hyperviseur VMware.	Hyperviseur
vmware.hv.cluster.name	Le nom du cluster de l’hyperviseur VMware.	
vmware.hv.connectionstate	L’état de connexion de l’hyperviseur VMware.	
vmware.hv.cpu.usage	L’utilisation du processeur de l’hyperviseur VMware (Hz).	
vmware.hv.cpu.usage.perf	L’utilisation du processeur de l’hyperviseur VMware en pourcentage pendant l’intervalle.	
vmware.hv.cpu.utilization	L’utilisation du processeur de l’hyperviseur VMware en pourcentage pendant l’intervalle, selon la gestion de l’alimentation ou l’HT.	
vmware.hv.datacenter.name	Le nom du centre de données de l’hyperviseur VMware.	
vmware.hv.datastore.discovery	La découverte des datastores de l’hyperviseur VMware.	
vmware.hv.datastore.list	La liste des datastores de l’hyperviseur VMware.	
vmware.hv.datastore.multipath	Le nombre de chemins de datastore disponibles.	
vmware.hv.datastore.read	Le temps moyen nécessaire à une opération de lecture depuis le datastore.	
vmware.hv.datastore.size	L’espace du datastore VMware en octets ou en pourcentage du total.	
vmware.hv.datastore.write	Le temps moyen nécessaire à une opération d’écriture vers le datastore.	
vmware.hv.discovery	La découverte des hyperviseurs VMware.	
vmware.hv.diskinfo.get	Les données de disque de l’hyperviseur VMware.	
vmware.hv.fullname	Le nom complet du produit, y compris les informations de version.	
vmware.hv.hw.cpu.freq	La fréquence du processeur de l’hyperviseur VMware.	
vmware.hv.hw.cpu.model	Le modèle du processeur de l’hyperviseur VMware.	
vmware.hv.hw.cpu.num	Le nombre de cœurs de processeur sur l’hyperviseur VMware.	
vmware.hv.hw.cpu.threads	Le nombre de threads de processeur sur l’hyperviseur VMware.	

Clé d'élément	Description	Groupe d'éléments
vmware.hv.hw.memory	La taille totale de la mémoire de l'hyperviseur VMware.	
vmware.hv.hw.model	Le modèle de l'hyperviseur VMware.	
vmware.hv.hw.sensors.get	La valeur des capteurs matériels de l'hyperviseur VMware.	
vmware.hv.hw.serialnumber	Le numéro de série de l'hyperviseur VMware.	
vmware.hv.hw.uuid	L'UUID BIOS de l'hyperviseur VMware.	
vmware.hv.hw.vendor	Le nom du fournisseur de l'hyperviseur VMware.	
vmware.hv.maintenance	L'état de maintenance de l'hyperviseur VMware.	
vmware.hv.memory.size.ballooned	La taille de la mémoire ballooned de l'hyperviseur VMware.	
vmware.hv.memory.used	La taille de la mémoire utilisée de l'hyperviseur VMware.	
vmware.hv.net.if.discovery	La découverte des interfaces réseau de l'hyperviseur VMware.	
vmware.hv.network.in	Les statistiques d'entrée réseau de l'hyperviseur VMware.	
vmware.hv.network.linkspeed	La vitesse de l'interface réseau de l'hyperviseur VMware.	
vmware.hv.network.out	Les statistiques de sortie réseau de l'hyperviseur VMware.	
vmware.hv.perfcounter	La valeur du compteur de performance de l'hyperviseur VMware.	
vmware.hv.property	La propriété de l'hyperviseur VMware.	
vmware.hv.power	La consommation électrique de l'hyperviseur VMware.	
vmware.hv.sensor.health.state	Le capteur agrégé de l'état de santé de l'hyperviseur VMware.	
vmware.hv.sensors.get	Les capteurs d'état du fournisseur matériel de l'hyperviseur VMware.	
vmware.hv.status	L'état de l'hyperviseur VMware.	
vmware.hv.tags.get	Le tableau des balises de l'hyperviseur VMware.	
vmware.hv.uptime	Le temps de fonctionnement de l'hyperviseur VMware.	
vmware.hv.version	La version de l'hyperviseur VMware.	
vmware.hv.vm.num	Le nombre de machines virtuelles sur l'hyperviseur VMware.	
vmware.rp.cpu.usage	L'utilisation du CPU en hertz pendant l'intervalle sur le Resource Pool VMware.	Resource pool
vmware.rp.memory	Les métriques de mémoire du resource pool VMware.	
vmware.alarms.get	Les données des alarmes du centre virtuel VMware.	Virtual center
vmware.vm.alarms.get	Les données des alarmes de la machine virtuelle VMware.	Virtual machine
vmware.vm.attribute	La valeur de l'attribut personnalisé de la machine virtuelle VMware.	
vmware.vm.cluster.name	Le nom de la machine virtuelle VMware.	
vmware.vm.consolidationneeded	Le disque de la machine virtuelle VMware nécessite une consolidation.	
vmware.vm.cpu.latency	Le pourcentage de temps pendant lequel la machine virtuelle ne peut pas s'exécuter parce qu'elle est en concurrence pour l'accès au(x) CPU physique(s).	
vmware.vm.cpu.num	Le nombre de processeurs sur la machine virtuelle VMware.	
vmware.vm.cpu.readiness	Le pourcentage de temps pendant lequel la machine virtuelle était prête, mais n'a pas pu être planifiée pour s'exécuter sur le CPU physique.	
vmware.vm.cpu.ready	Le temps pendant lequel la machine virtuelle était prête, mais n'a pas pu être planifiée pour s'exécuter sur le CPU physique.	
vmware.vm.cpu.swapwait	Le pourcentage de temps CPU passé en attente du swap-in.	
vmware.vm.cpu.usage	L'utilisation du processeur de la machine virtuelle VMware (Hz).	
vmware.vm.cpu.usage.perf	L'utilisation du processeur de la machine virtuelle VMware en pourcentage pendant l'intervalle.	
vmware.vm.datacenter.name	Le nom du centre de données de la machine virtuelle VMware.	
vmware.vm.discovery	La découverte des machines virtuelles VMware.	
vmware.vm.guest.memory.size.swapped	La quantité de mémoire physique invité qui est échangée vers l'espace de swap.	
vmware.vm.guest.osuptime	Le temps total écoulé depuis le dernier démarrage du système d'exploitation.	
vmware.vm.hv.maintenance	L'état de maintenance de l'hyperviseur de la machine virtuelle VMware.	
vmware.vm.hv.name	Le nom de l'hyperviseur de la machine virtuelle VMware.	
vmware.vm.memory.size	La taille totale de la mémoire de la machine virtuelle VMware.	
vmware.vm.memory.size.ballooned	La taille de la mémoire ballooned de la machine virtuelle VMware.	
vmware.vm.memory.size.compressed	La taille de la mémoire compressée de la machine virtuelle VMware.	
vmware.vm.memory.size.consumed	La quantité de mémoire physique de l'hôte consommée pour sauvegarder les pages de mémoire physique invité.	
vmware.vm.memory.size.private	La taille de la mémoire privée de la machine virtuelle VMware.	
vmware.vm.memory.size.shared	La taille de la mémoire partagée de la machine virtuelle VMware.	
vmware.vm.memory.size.swapped	La taille de la mémoire échangée de la machine virtuelle VMware.	
vmware.vm.memory.size.usage.guest	L'utilisation de la mémoire invité de la machine virtuelle VMware.	
vmware.vm.memory.size.usage.host	L'utilisation de la mémoire de l'hôte par la machine virtuelle VMware.	
vmware.vm.memory.usage	Le pourcentage de mémoire physique de l'hôte qui a été consommée.	

Clé d'élément	Description	Groupe d'éléments
vmware.vm.net.if.discovery	La découverte des interfaces réseau de la machine virtuelle VMware.	
vmware.vm.net.if.in	Les statistiques d'entrée de l'interface réseau de la machine virtuelle VMware.	
vmware.vm.net.if.out	Les statistiques de sortie de l'interface réseau de la machine virtuelle VMware.	
vmware.vm.net.if.usage	L'utilisation du réseau de la machine virtuelle VMware pendant l'intervalle.	
vmware.vm.perfcounter	La valeur du compteur de performance de la machine virtuelle VMware.	
vmware.vm.powerstate	L'état d'alimentation de la machine virtuelle VMware.	
vmware.vm.property	La propriété de la machine virtuelle VMware.	
vmware.vm.snapshot.get	L'état du snapshot de la machine virtuelle VMware.	
vmware.vm.state	L'état de la machine virtuelle VMware.	
vmware.vm.storage.committed	L'espace de stockage engagé de la machine virtuelle VMware.	
vmware.vm.storage.readoio	Le nombre moyen de requêtes de lecture en attente vers le disque virtuel pendant l'intervalle de collecte.	
vmware.vm.storage.totalreadlatency	Le temps moyen nécessaire à une lecture depuis le disque virtuel.	
vmware.vm.storage.totalwritelatency	Le temps moyen nécessaire à une écriture vers le disque virtuel.	
vmware.vm.storage.uncommitted	L'espace de stockage non engagé de la machine virtuelle VMware.	
vmware.vm.storage.unshared	L'espace de stockage non partagé de la machine virtuelle VMware.	
vmware.vm.storage.writeoio	Le nombre moyen de requêtes d'écriture en attente vers le disque virtuel pendant l'intervalle de collecte.	
vmware.vm.tags.get	Le tableau des balises de la machine virtuelle VMware.	
vmware.vm.tools	L'état ou la version des outils invité de la machine virtuelle VMware.	
vmware.vm.uptime	Le temps de fonctionnement de la machine virtuelle VMware.	
vmware.vm.vfs.dev.discovery	La découverte des périphériques de disque de la machine virtuelle VMware.	
vmware.vm.vfs.dev.read	Les statistiques de lecture du périphérique de disque de la machine virtuelle VMware.	
vmware.vm.vfs.dev.write	Les statistiques d'écriture du périphérique de disque de la machine virtuelle VMware.	
vmware.vm.vfs.fs.discovery	La découverte des systèmes de fichiers de la machine virtuelle VMware.	
vmware.vm.vfs.fs.size	Les statistiques du système de fichiers de la machine virtuelle VMware.	

Détails de la clé d'élément Les paramètres sans crochets angulaires sont obligatoires. Les paramètres marqués par des crochets angulaires **< >** sont facultatifs.

vmware.eventlog[url,<mode>,<severity>]

 Le journal des événements VMware.
 Valeur renvoyée : *Log*.

Paramètres :

- **url** - l'URL du service VMware ;
- **mode** - *all* (par défaut) ou *skip* - ignorer le traitement des anciennes données ;
- **severity** - filtrer par gravité : *error*, *warning*, *info* ou *user*. Ce paramètre doit être placé entre guillemets si plusieurs niveaux de gravité sont spécifiés dans une liste délimitée par des virgules (par ex.`"error,warning,info,user"`). Désactivé par défaut.

Commentaires :

- Il ne doit y avoir qu'une seule clé d'élément `vmware.eventlog` par URL ;
- Voir aussi l'exemple de filtrage des enregistrements du journal des événements VMware.

vmware.fullnameurl

 Le nom complet du service VMware.
 Valeur de retour : *String*.

Paramètres :

- **url** - l'URL du service VMware.

vmware.versionurl

 La version du service VMware.
 Valeur de retour : *String*.

Paramètres :

- **url** - l'URL du service VMware.

vmware.cl.perfcounter[url,id,path,<instance>]

 Les métriques du compteur de performance du cluster VMware.
 Valeur de retour : *Integer*.

Paramètres :

- **url** - l'URL du service VMware ;
- **id** - l'ID du cluster VMware. `id` peut être obtenu à partir de `vmware.cluster.discovery[]` sous la forme {#CLUSTER.ID}.
- **path** - le chemin du compteur de performance[1] ;
- **instance** - l'instance du compteur de performance.

vmware.cluster.alarms.get[url,id]

 Les données des alarmes du cluster VMware.
 Valeur de retour: objet JSON.

Paramètres:

- **url** - l'URL du service VMware;
- **id** - l'ID du cluster VMware.

vmware.cluster.discoveryurl

 La découverte des clusters VMware.
 Valeur de retour : objet JSON.

Paramètres :

- **url** - l'URL du service VMware.

vmware.cluster.property[url,id,prop]

 La propriété du cluster VMware.
 Valeur de retour : *String*.

Paramètres :

- **url** - l'URL du service VMware ;
- **id** - l'ID du cluster VMware ;
- **prop** - le chemin de propriété, qui est le nom d'une propriété de l'objet VM telle que définie dans le VMware SDK.

Exemples :

```
vmware.vm.property[{$VMWARE.URL},{$VMWARE.VM.UUID},overallStatus]
vmware.vm.property[{$VMWARE.URL},{$VMWARE.VM.UUID},runtime.powerState]
```

vmware.cluster.status[url,name]

 Le statut du cluster VMware.
 Valeur de retour : *0* - gris ; *1* - vert ; *2* - jaune ; *3* - rouge.

Paramètres :

- **url** - l'URL du service VMware ;
- **name** - le nom du cluster VMware.

vmware.cluster.tags.get[url,id]

 Le tableau des balises du cluster VMware.
 Valeur de retour: objet JSON.

Paramètres:

- **url** - l'URL du service VMware;
- **id** - l'ID du cluster VMware.

Cet élément fonctionne avec vSphere 6.5 et les versions ultérieures.

vmware.datastore.alarms.get[url,uuid]

 Les données des déclencheurs du datastore VMware.
 Valeur de retour: objet JSON.

Paramètres:

- **url** - l'URL du service VMware;
- **uuid** - l'identifiant unique global du datastore VMware.

vmware.datastore.discovery[url,<filter_uuid>]

 La découverte des datastores VMware.
 Valeur de retour : objet JSON.

Paramètres :

- **url** - l'URL du service VMware ;
- **filter_uuid** - un identifiant unique d'un seul HV ou d'une seule VM (par défaut : vide). S'il est spécifié, seuls les datastores liés seront découverts.

vmware.datastore.hv.list[url,datastore]

 La liste des hyperviseurs du datastore.
 Valeur de retour : *String*.

Paramètres :

- **url** - l'URL du service VMware ;
- **datastore** - l'UUID ou le nom du datastore.

Exemple de sortie :

```
esx7-01-host.zabbix.sandbox
esx7-02-host.zabbix.sandbox
```

vmware.datastore.perfcounter[url,uuid,path,<instance>]

 La valeur du compteur de performance du datastore VMware.
 Valeur de retour : *Integer* [2].

Paramètres :

- **url** - l'URL du service VMware ;
- **uuid** - l'identifiant global unique du datastore VMware ;
- **path** - le chemin du compteur de performance [1] ;
- **instance** - l'instance du compteur de performance. Utilisez une instance vide pour les valeurs agrégées (par défaut). `instance` peut être obtenue à partir de `vmware.datastore.discovery[]` dans le tableau {#DATASTORE.EXTENT}.

vmware.datastore.property[url,uuid,prop]

 La propriété du datastore VMware.
 Valeur de retour : *String*.

Paramètres :

- **url** - l'URL du service VMware ;
- **uuid** - l'identifiant global unique du datastore VMware ;
- **prop** - le chemin de la propriété.

vmware.datastore.read[url,datastore,<mode>]

 Le temps nécessaire pour une opération de lecture depuis le datastore (en millisecondes).
 Valeur de retour : *Integer* [2].

Paramètres :

- **url** - l'URL du service VMware ;
- **datastore** - l'UUID ou le nom du datastore ;
- **mode** - *latency* (valeur moyenne, par défaut) ou *maxlatency* (valeur maximale).

vmware.datastore.size[url,datastore,<mode>]

 L'espace du datastore VMware en octets ou en pourcentage du total.
 Valeur de retour : *Integer* - pour les octets ; *Float* - pour le pourcentage.

Paramètres :

- **url** - l'URL du service VMware ;
- **datastore** - l'UUID ou le nom du datastore ;
- **mode** - valeurs possibles : *total* (par défaut), *free*, *pfree* (pourcentage libre), *uncommitted*.

vmware.datastore.tags.get[url,uuid]

 Le tableau des balises du datastore VMware.
 Valeur de retour: objet JSON.

Paramètres:

- **url** - l'URL du service VMware;
- **uuid** - l'identifiant unique global du datastore VMware.

Cet élément fonctionne avec vSphere 6.5 et les versions ultérieures.

vmware.datastore.write[url,datastore,<mode>]

 Le temps nécessaire pour une opération d'écriture sur le datastore (en millisecondes).
 Valeur de retour : *Integer* [2].

Paramètres :

- **url** - l'URL du service VMware ;
- **datastore** - l'UUID ou le nom du datastore ;
- **mode** - *latency* (valeur moyenne, par défaut) ou *maxlatency* (valeur maximale).

vmware.dc.alarms.get[url,id]

 Les données des alarmes du centre de données VMware.
 Valeur de retour: objet JSON.

Paramètres:

- **url** - l'URL du service VMware;
- **id** - l'ID du centre de données VMware.

vmware.dc.discoveryurl

 La découverte des centres de données VMware.
 Valeur de retour : objet JSON.

Paramètres :

- **url** - l'URL du service VMware.

vmware.dc.tags.get[url,id]

 Le tableau des tags du datacenter VMware.
 Valeur de retour: objet JSON.

Paramètres:

- **url** - l'URL du service VMware;
- **id** - l'ID du datacenter VMware.

Cet élément fonctionne avec vSphere 6.5 et les versions plus récentes.

vmware.dvswitch.discoveryurl

 La découverte des commutateurs distribués VMware vSphere.
 Valeur de retour : objet JSON.

Paramètres :

- **url** - l'URL du service VMware.

vmware.dvswitch.fetchports.get[url,uuid,<filter>,<mode>]

 Les données des ports du VMware vSphere Distributed Switch.
 Valeur de retour: objet JSON.

Paramètres:

- **url** - l'URL du service VMware;
- **uuid** - l'identifiant unique global du DVSwitch VMware.
- **filter** - une chaîne unique avec des critères séparés par des virgules pour sélectionner les ports;
- **mode** - *state* (tout le XML sans les nœuds XML "config", par défaut) ou *full*.

Le paramètre **filter** prend en charge les critères disponibles dans l'objet de données VMware DistributedVirtualSwitchPortCriteria.

Exemple:

```
vmware.dvswitch.fetchports.get[{$VMWARE.URL},{$VMWARE.DVS.UUID},"connected:true,active:true,uplinkPort:fal
```

vmware.hv.alarms.get[url,uuid]

 Les données des alarmes de l'hyperviseur VMware.
 Valeur de retour : objet JSON.

Paramètres :

- **url** - l'URL du service VMware ;
- **uuid** - l'identifiant unique global de l'hyperviseur VMware.

vmware.hv.cluster.name[url,uuid]

 Le nom du cluster de l'hyperviseur VMware.
 Valeur de retour : *String*.

Paramètres :

- **url** - l'URL du service VMware ;
- **uuid** - l'identifiant global unique de l'hyperviseur VMware.

vmware.hv.connectionstate[url,uuid]

 L'état de connexion de l'hyperviseur VMware.
 Valeur renvoyée : *String* : *connected*, *disconnected* ou *notResponding*.

Paramètres :

- **url** - l'URL du service VMware ;
- **uuid** - l'identifiant global unique de l'hyperviseur VMware.

vmware.hv.cpu.usage[url,uuid]

 L'utilisation du processeur de l'hyperviseur VMware (Hz).
 Valeur de retour : *Integer*.

Paramètres :

- **url** - l'URL du service VMware ;
- **uuid** - l'identifiant global unique de l'hyperviseur VMware.

vmware.hv.cpu.usage.perf[url,uuid]

 L'utilisation du processeur de l'hyperviseur VMware en pourcentage pendant l'intervalle.
 Valeur de retour : *Float*.

Paramètres :

- **url** - l'URL du service VMware ;
- **uuid** - l'identifiant global unique de l'hyperviseur VMware.

vmware.hv.cpu.utilization[url,uuid]

 L'utilisation du processeur de l'hyperviseur VMware en pourcentage pendant l'intervalle dépend de la gestion de l'alimentation ou de HT.
 Valeur de retour : *Float*.

Paramètres :

- **url** - l'URL du service VMware ;
- **uuid** - l'identifiant global unique de l'hyperviseur VMware.

vmware.hv.datacenter.name[url,uuid]

 Le nom du centre de données de l'hyperviseur VMware.
 Valeur de retour : *String*.

Paramètres :

- **url** - l'URL du service VMware ;
- **uuid** - l'identifiant global unique de l'hyperviseur VMware.

vmware.hv.datastore.discovery[url,uuid]

 La découverte des banques de données des hyperviseurs VMware.
 Valeur de retour : objet JSON.

Paramètres :

- **url** - l'URL du service VMware ;
- **uuid** - l'identifiant global unique de l'hyperviseur VMware.

vmware.hv.datastore.list[url,uuid]

 La liste des datastores de l'hyperviseur VMware.
 Valeur de retour : *String*.

Paramètres :

- **url** - l'URL du service VMware ;
- **uuid** - l'identifiant global unique de l'hyperviseur VMware.

Exemple de sortie :

```
SSD-RAID1-VAULT1
SSD-RAID1-VAULT2
SSD-RAID10
```

vmware.hv.datastore.multipath[url,uuid,<datastore>,<partitionid>]

 Le nombre de chemins de datastore disponibles.
 Valeur de retour : *Integer*.

Paramètres :

- **url** - l'URL du service VMware ;
- **uuid** - l'identifiant global unique de l'hyperviseur VMware ;
- **datastore** - l'UUID ou le nom du datastore ;
- **partitionid** - l'ID interne du périphérique physique provenant de `vmware.hv.datastore.discovery`.

vmware.hv.datastore.read[url,uuid,datastore,<mode>]

 Le temps moyen d'une opération de lecture depuis le datastore (en millisecondes).
 Valeur de retour : *Integer* [2].

Paramètres :

- **url** - l'URL du service VMware ;

- **uuid** - l'identifiant global unique de l'hyperviseur VMware ;
- **datastore** - l'UUID ou le nom du datastore ;
- **mode** - *latency* (par défaut).

vmware.hv.datastore.size[url,uuid,datastore,<mode>]

 L'espace du datastore VMware en octets ou en pourcentage du total.
 Valeur de retour : *Integer* - pour les octets ; *Float* - pour le pourcentage.

Paramètres :

- **url** - l'URL du service VMware ;
- **uuid** - l'identifiant global unique de l'hyperviseur VMware ;
- **datastore** - l'UUID ou le nom du datastore ;
- **mode** - valeurs possibles : *total* (par défaut), *free*, *pfree* (pourcentage libre), *uncommitted*.

vmware.hv.datastore.write[url,uuid,datastore,<mode>]

 Le temps moyen d'une opération d'écriture vers le datastore (en millisecondes).
 Valeur de retour : *Integer* [2].

Paramètres :

- **url** - l'URL du service VMware ;
- **uuid** - l'identifiant global unique de l'hyperviseur VMware ;
- **datastore** - l'UUID ou le nom du datastore ;
- **mode** - *latency* (par défaut).

vmware.hv.discoveryurl

 La découverte des hyperviseurs VMware.
 Valeur de retour : objet JSON.

Paramètres :

- **url** - l'URL du service VMware.

vmware.hv.diskinfo.get[url,uuid]

 Les données de disque de l'hyperviseur VMware.
 Valeur de retour: objet JSON.

Paramètres:

- **url** - l'URL du service VMware;
- **uuid** - l'identifiant unique global de l'hyperviseur VMware.

vmware.hv.fullname[url,uuid]

 Le nom complet du produit, y compris les informations de version.
 Valeur de retour : *String*.

Paramètres :

- **url** - l'URL du service VMware ;
- **uuid** - l'identifiant global unique de l'hyperviseur VMware.

vmware.hv.hw.cpu.freq[url,uuid]

 La fréquence du processeur de l'hyperviseur VMware (Hz).
 Valeur de retour : *Integer*.

Paramètres :

- **url** - l'URL du service VMware ;
- **uuid** - l'identifiant global unique de l'hyperviseur VMware.

vmware.hv.hw.cpu.model[url,uuid]

 Le modèle du processeur de l'hyperviseur VMware.
 Valeur de retour : *String*.

Paramètres :

- **url** - l'URL du service VMware ;
- **uuid** - l'identifiant global unique de l'hyperviseur VMware.

vmware.hv.hw.cpu.num[url,uuid]

 Le nombre de cœurs de processeur sur l'hyperviseur VMware.
 Valeur de retour : *Integer*.

Paramètres :

- **url** - l'URL du service VMware ;
- **uuid** - l'identifiant global unique de l'hyperviseur VMware.

vmware.hv.hw.cpu.threads[url,uuid]

 Le nombre de threads de processeur sur l'hyperviseur VMware.
 Valeur de retour : *Integer*.

Paramètres :

- **url** - l'URL du service VMware ;
- **uuid** - l'identifiant global unique de l'hyperviseur VMware.

vmware.hv.hw.memory[url,uuid]

 La taille totale de la mémoire de l'hyperviseur VMware (en octets).
 Valeur de retour : *Integer*.

Paramètres :

- **url** - l'URL du service VMware ;
- **uuid** - l'identifiant global unique de l'hyperviseur VMware.

vmware.hv.hw.model[url,uuid]

 Le modèle de l'hyperviseur VMware.
 Valeur de retour : *String*.

Paramètres :

- **url** - l'URL du service VMware ;
- **uuid** - l'identifiant global unique de l'hyperviseur VMware.

vmware.hv.hw.sensors.get[url,uuid]

 La valeur des capteurs matériels de l'hyperviseur VMware.
 Valeur de retour: objet JSON.

Paramètres:

- **url** - l'URL du service VMware;
- **uuid** - l'identifiant unique global de l'hyperviseur VMware.

vmware.hv.hw.serialnumber[url,uuid]

 Le numéro de série de l'hyperviseur VMware.
 Valeur de retour : *String*.

Paramètres :

- **url** - l'URL du service VMware ;
- **uuid** - l'identifiant global unique de l'hyperviseur VMware.

Cet élément fonctionne avec l'API vSphere 6.7 et les versions plus récentes.

vmware.hv.hw.uuid[url,uuid]

 L'UUID du BIOS de l'hyperviseur VMware.
 Valeur de retour : *String*.

Paramètres :

- **url** - l'URL du service VMware ;
- **uuid** - l'identifiant global unique de l'hyperviseur VMware.

Cet élément fonctionne avec l'API vSphere 6.7 et les versions plus récentes.

vmware.hv.hw.vendor[url,uuid]

 Le nom du fournisseur de l'hyperviseur VMware.
 Valeur de retour : *String*.

Paramètres :

- **url** - l'URL du service VMware ;
- **uuid** - l'identifiant global unique de l'hyperviseur VMware.

Cet élément fonctionne avec l'API vSphere 6.7 et les versions plus récentes.

vmware.hv.maintenance[url,uuid]

 Le statut de maintenance de l'hyperviseur VMware.
 Valeur de retour : *0* - pas en maintenance ; *1* - en maintenance.

Paramètres :

- **url** - l'URL du service VMware ;
- **uuid** - l'identifiant global unique de l'hyperviseur VMware.

vmware.hv.memory.size.ballooned[url,uuid]

 La taille de la mémoire gonflée du hyperviseur VMware (en octets).
 Valeur de retour : *Integer*.

Paramètres :

- **url** - l'URL du service VMware ;
- **uuid** - l'identifiant global unique de l'hyperviseur VMware.

vmware.hv.memory.used[url,uuid]

 La taille de la mémoire utilisée par l'hyperviseur VMware (en octets).
 Valeur de retour : *Integer*.

Paramètres :

- **url** - l'URL du service VMware ;
- **uuid** - l'identifiant global unique de l'hyperviseur VMware.

vmware.hv.net.if.discovery[url,uuid]

 La découverte des interfaces réseau de l'hyperviseur VMware.
 Valeur de retour : *objet JSON*.

Paramètres :

- **url** - l'URL du service VMware ;
- **uuid** - l'identifiant global unique de l'hyperviseur VMware.

vmware.hv.network.in[url,uuid,<mode>]

 Les statistiques d'entrée réseau de l'hyperviseur VMware (octets par seconde).
 Valeur de retour : *Integer* [2].

Paramètres :

- **url** - l'URL du service VMware ;
- **uuid** - l'identifiant global unique de l'hyperviseur VMware ;
- **mode** - *bps* (par défaut), *packets*, *dropped*, *errors*, *broadcast*.

vmware.hv.network.linkspeed[url,uuid,ifname]

 La vitesse de l'interface réseau de l'hyperviseur VMware.
 Valeur de retour : *Integer*. Renvoie *0* si l'interface réseau est hors service, sinon la valeur de vitesse de l'interface.

Paramètres :

- **url** - l'URL du service VMware ;
- **uuid** - l'identifiant global unique de l'hyperviseur VMware ;
- **ifname** - le nom de l'interface.

vmware.hv.network.out[url,uuid,<mode>]

 Les statistiques de sortie réseau de l'hyperviseur VMware (octets par seconde).
 Valeur de retour : *Integer* [2].

Paramètres :

- **url** - l'URL du service VMware ;
- **uuid** - l'identifiant global unique de l'hyperviseur VMware ;
- **mode** - *bps* (par défaut), *packets*, *dropped*, *errors*, *broadcast*.

vmware.hv.perfcounter[url,uuid,path,<instance>]

 La valeur du compteur de performance de l'hyperviseur VMware.
 Valeur de retour : *Integer* [2].

Paramètres :

- **url** - l'URL du service VMware ;
- **uuid** - l'identifiant global unique de l'hyperviseur VMware ;
- **path** - le chemin du compteur de performance [1] ;
- **instance** - l'instance du compteur de performance. Utilisez une instance vide pour les valeurs agrégées (par défaut).

vmware.hv.property[url,uuid,prop]

 La propriété de l'hyperviseur VMware.
 Valeur de retour : *String*.

Paramètres :

- **url** - l'URL du service VMware ;
- **uuid** - l'identifiant global unique de l'hyperviseur VMware ;
- **prop** - le chemin de la propriété.

vmware.hv.power[url,uuid,<max>]

 La consommation électrique de l'hyperviseur VMware (W).
 Valeur de retour : *Integer*.

Paramètres :

- **url** - l'URL du service VMware ;
- **uuid** - l'identifiant global unique de l'hyperviseur VMware ;
- **max** - la consommation électrique maximale autorisée.

vmware.hv.sensor.health.state[url,uuid]

 Le capteur de synthèse de l'état de santé de l'hyperviseur VMware.
 Valeur de retour : *Integer* : *0* - gris ; *1* - vert ; *2* - jaune ; *3* - rouge.

Paramètres :

- **url** - l'URL du service VMware ;
- **uuid** - l'identifiant global unique de l'hyperviseur VMware.

Notez que l'élément peut ne pas fonctionner dans VMware vSphere 6.5 et versions ultérieures, car VMware a abandonné le capteur *VMware Rollup Health State*.

vmware.hv.sensors.get[url,uuid]

 Les capteurs d'état du fournisseur matériel du hyperviseur VMware.
 Valeur de retour: objet JSON.

Paramètres:

- **url** - l'URL du service VMware;
- **uuid** - l'identifiant unique global de l'hyperviseur VMware.

vmware.hv.status[url,uuid]

 Le statut de l'hyperviseur VMware.
 Valeur de retour : *Integer* : *0* - gris ; *1* - vert ; *2* - jaune ; *3* - rouge.

Paramètres :

- **url** - l'URL du service VMware ;
- **uuid** - l'identifiant global unique de l'hyperviseur VMware.

Cet élément utilise la propriété de statut global du système hôte.

vmware.hv.tags.get[url,uuid]

 Le tableau des balises de l'hyperviseur VMware.
 Valeur de retour: objet JSON.

Paramètres:

- **url** - l'URL du service VMware;
- **uuid** - l'identifiant unique global de l'hyperviseur VMware.

Cet élément fonctionne avec vSphere 6.5 et les versions ultérieures.

vmware.hv.uptime[url,uuid]

 Le temps de fonctionnement de l'hyperviseur VMware (en secondes).
 Valeur de retour : *Integer*.

Paramètres :

- **url** - l'URL du service VMware ;
- **uuid** - l'identifiant global unique de l'hyperviseur VMware.

Cet élément utilise la propriété d'état global du système hôte.

vmware.hv.version[url,uuid]

 La version de l'hyperviseur VMware.
 Valeur de retour : *String*.

Paramètres :

- **url** - l'URL du service VMware ;
- **uuid** - l'identifiant global unique de l'hyperviseur VMware.

vmware.hv.vm.num[url,uuid]

 Le nombre de machines virtuelles sur l'hyperviseur VMware.
 Valeur de retour : *Integer*.

Paramètres :

- **url** - l'URL du service VMware ;
- **uuid** - l'identifiant global unique de l'hyperviseur VMware.

vmware.rp.cpu.usage[url,rpid]

 L'utilisation du processeur en hertz pendant l'intervalle sur le pool de ressources VMware.
 Valeur de retour : *Integer*.

Paramètres :

- **url** - l'URL du service VMware ;
- **rpid** - l'ID du pool de ressources VMware.

vmware.rp.memory[url,rpid,<mode>]

 Les métriques de mémoire du pool de ressources VMware.
 Valeur de retour : *Integer*.

Paramètres :

- **url** - l'URL du service VMware ;
- **rpid** - l'ID du pool de ressources VMware ;
- **mode** - valeurs possibles :
consumed (par défaut) - la quantité de mémoire physique de l'hôte consommée pour sauvegarder les pages de mémoire physique invitées
 ballooned - la quantité de mémoire physique invitée récupérée de la machine virtuelle par le pilote de ballooning dans l'invité
overhead - la mémoire physique de l'hôte consommée par les structures de données ESXi pour exécuter les machines virtuelles

vmware.alarms.geturl

 Les données des alarmes du centre virtuel VMware.
 Valeur de retour: objet JSON.

Paramètres:

- **url** - l'URL du service VMware.

vmware.vm.alarms.get[url,uuid]

 Les données des alarmes de la machine virtuelle VMware.
 Valeur de retour: objet JSON.

Paramètres:

- **url** - l'URL du service VMware;
- **uuid** - l'identifiant unique global de la machine virtuelle VMware.

vmware.vm.attribute[url,uuid,name]

 La valeur de l'attribut personnalisé de la machine virtuelle VMware.
 Valeur de retour : *String*.

Paramètres :

- **url** - l'URL du service VMware ;
- **uuid** - l'identifiant global unique de la machine virtuelle VMware ;
- **name** - le nom de l'attribut personnalisé.

vmware.vm.cluster.name[url,uuid]

 Le nom de la machine virtuelle VMware.
 Valeur de retour : *String*.

Paramètres :

- **url** - l'URL du service VMware ;
- **uuid** - l'identifiant global unique de la machine virtuelle VMware.

vmware.vm.consolidationneeded[url,uuid]

 Le disque de la machine virtuelle VMware nécessite une consolidation.
 Valeur de retour : *String* : *true* - une consolidation est nécessaire ; *false* - une consolidation n'est pas nécessaire.

Paramètres :

- **url** - l'URL du service VMware ;
- **uuid** - l'identifiant global unique de la machine virtuelle VMware.

vmware.vm.cpu.latency[url,uuid]

 Le pourcentage de temps pendant lequel la machine virtuelle ne peut pas s'exécuter parce qu'elle est en concurrence pour l'accès au(x) CPU physique(s).
 Valeur de retour : *Float*.

Paramètres :

- **url** - l'URL du service VMware ;

- **uuid** - l'identifiant global unique de la machine virtuelle VMware.

vmware.vm.cpu.num[url,uuid]

 Le nombre de processeurs sur la machine virtuelle VMware.
 Valeur de retour : *Integer*.

Paramètres :

- **url** - l'URL du service VMware ;
- **uuid** - l'identifiant global unique de la machine virtuelle VMware.

vmware.vm.cpu.readiness[url,uuid,<instance>]

 Le pourcentage de temps pendant lequel la machine virtuelle était prête, mais n'a pas pu être planifiée pour s'exécuter sur le CPU physique.
 Valeur de retour : *Float*.

Paramètres :

- **url** - l'URL du service VMware ;
- **uuid** - l'identifiant global unique de la machine virtuelle VMware ;
- **instance** - l'instance du CPU.

vmware.vm.cpu.ready[url,uuid]

 Le temps (en millisecondes) pendant lequel la machine virtuelle était prête, mais n'a pas pu être planifiée pour s'exécuter sur le CPU physique. Le temps de disponibilité du CPU dépend du nombre de machines virtuelles sur l'hôte et de leurs charges CPU (%).
 Valeur de retour : *Integer* [2].

Paramètres :

- **url** - l'URL du service VMware ;
- **uuid** - l'identifiant global unique de la machine virtuelle VMware.

vmware.vm.cpu.swapwait[url,uuid,<instance>]

 Le pourcentage du temps CPU passé à attendre le swap-in.
 Valeur de retour : *Float*.

Paramètres :

- **url** - l'URL du service VMware ;
- **uuid** - l'identifiant global unique de la machine virtuelle VMware ;
- **instance** - l'instance CPU.

vmware.vm.cpu.usage[url,uuid]

 L'utilisation du processeur de la machine virtuelle VMware (Hz).
 Valeur de retour : *Integer*.

Paramètres :

- **url** - l'URL du service VMware ;
- **uuid** - l'identifiant global unique de la machine virtuelle VMware.

vmware.vm.cpu.usage.perf[url,uuid]

 L'utilisation du processeur de la machine virtuelle VMware en pourcentage pendant l'intervalle.
 Valeur de retour : *Float*.

Paramètres :

- **url** - l'URL du service VMware ;
- **uuid** - l'identifiant global unique de la machine virtuelle VMware.

vmware.vm.datacenter.name[url,uuid]

 Le nom du datacenter de la machine virtuelle VMware.
 Valeur de retour : *String*.

Paramètres :

- **url** - l'URL du service VMware ;
- **uuid** - l'identifiant global unique de la machine virtuelle VMware.

vmware.vm.discoveryurl

 La découverte des machines virtuelles VMware.
 Valeur de retour : objet JSON.

Paramètres :

- **url** - l'URL du service VMware.

vmware.vm.guest.memory.size.swapped[url,uuid]

 La quantité de mémoire physique invitée échangée vers l'espace de swap (KB).
 Valeur de retour : *Integer*.

Paramètres :

- **url** - l'URL du service VMware ;
- **uuid** - l'identifiant global unique de la machine virtuelle VMware.

vmware.vm.guest.osuptime[url,uuid]

 Le temps total écoulé depuis le dernier démarrage du système d'exploitation (en secondes).
 Valeur de retour : *Integer*.

Paramètres :

- **url** - l'URL du service VMware ;
- **uuid** - l'identifiant global unique de la machine virtuelle VMware.

vmware.vm.hv.maintenance[url,uuid]

 Le statut de maintenance de l'hyperviseur de la machine virtuelle VMware.
 Valeur de retour : *0* - pas en maintenance ; *1* - en maintenance.

Paramètres :

- **url** - l'URL du service VMware ;
- **uuid** - l'identifiant global unique de la machine virtuelle VMware.

vmware.vm.hv.name[url,uuid]

 Le nom de l'hyperviseur de la machine virtuelle VMware.
 Valeur de retour : *String*.

Paramètres :

- **url** - l'URL du service VMware ;
- **uuid** - l'identifiant global unique de la machine virtuelle VMware.

vmware.vm.memory.size[url,uuid]

 La taille totale de la mémoire de la machine virtuelle VMware (octets).
 Valeur de retour : *Integer*.

Paramètres :

- **url** - l'URL du service VMware ;
- **uuid** - l'identifiant global unique de la machine virtuelle VMware.

vmware.vm.memory.size.ballooned[url,uuid]

 La taille de la mémoire gonflée de la machine virtuelle VMware (en octets).
 Valeur de retour : *Integer*.

Paramètres :

- **url** - l'URL du service VMware ;
- **uuid** - l'identifiant global unique de la machine virtuelle VMware.

vmware.vm.memory.size.compressed[url,uuid]

 La taille de la mémoire compressée de la machine virtuelle VMware (octets).
 Valeur de retour : *Integer*.

Paramètres :

- **url** - l'URL du service VMware ;
- **uuid** - l'identifiant global unique de la machine virtuelle VMware.

vmware.vm.memory.size.consumed[url,uuid]

 La quantité de mémoire physique de l'hôte consommée pour la sauvegarde des pages de mémoire physique de l'invité (KB).
 Valeur de retour : *Integer*.

Paramètres :

- **url** - l'URL du service VMware ;
- **uuid** - l'identifiant global unique de la machine virtuelle VMware.

vmware.vm.memory.size.private[url,uuid]

 La taille de la mémoire privée de la machine virtuelle VMware (octets).
 Valeur de retour : *Integer*.

Paramètres :

- **url** - l'URL du service VMware ;
- **uuid** - l'identifiant global unique de la machine virtuelle VMware.

vmware.vm.memory.size.shared[url,uuid]

 La taille de la mémoire partagée de la machine virtuelle VMware (octets).
 Valeur de retour : *Integer*.

Paramètres :

- **url** - l'URL du service VMware ;
- **uuid** - l'identifiant global unique de la machine virtuelle VMware.

vmware.vm.memory.size.swapped[url,uuid]

 La taille de la mémoire échangée de la machine virtuelle VMware (en octets).
 Valeur de retour : *Integer*.

Paramètres :

- **url** - l'URL du service VMware ;
- **uuid** - l'identifiant global unique de la machine virtuelle VMware.

vmware.vm.memory.size.usage.guest[url,uuid]

 L'utilisation de la mémoire de l'invité de la machine virtuelle VMware (en octets).
 Valeur de retour : *Integer*.

Paramètres :

- **url** - l'URL du service VMware ;
- **uuid** - l'identifiant global unique de la machine virtuelle VMware.

vmware.vm.memory.size.usage.host[url,uuid]

 L'utilisation de la mémoire de l'hôte de la machine virtuelle VMware (en octets).
 Valeur de retour : *Integer*.

Paramètres :

- **url** - l'URL du service VMware ;
- **uuid** - l'identifiant global unique de la machine virtuelle VMware.

vmware.vm.memory.usage[url,uuid]

 Le pourcentage de la mémoire physique de l'hôte qui a été consommé.
 Valeur de retour : *Float*.

Paramètres :

- **url** - l'URL du service VMware ;
- **uuid** - l'identifiant global unique de la machine virtuelle VMware.

vmware.vm.net.if.discovery[url,uuid]

 La découverte des interfaces réseau des machines virtuelles VMware.
 Valeur de retour : objet JSON.

Paramètres :

- **url** - l'URL du service VMware ;
- **uuid** - l'identifiant global unique de la machine virtuelle VMware.

vmware.vm.net.if.in[url,uuid,instance,<mode>]

 Les statistiques d'entrée de l'interface réseau de la machine virtuelle VMware (octets/paquets par seconde).
 Valeur de retour : *Integer* [2].

Paramètres :

- **url** - l'URL du service VMware ;
- **uuid** - l'identifiant global unique de la machine virtuelle VMware ;
- **instance** - l'instance de l'interface réseau ;
- **mode** - *bps* (par défaut) ou *pps* - octets ou paquets par seconde.

vmware.vm.net.if.out[url,uuid,instance,<mode>]

 Les statistiques de sortie de l'interface réseau de la machine virtuelle VMware (octets/paquets par seconde).
 Valeur de retour : *Integer* [2].

Paramètres :

- **url** - l'URL du service VMware ;
- **uuid** - l'identifiant global unique de la machine virtuelle VMware ;
- **instance** - l'instance de l'interface réseau ;

- **mode** - *bps* (par défaut) ou *pps* - octets ou paquets par seconde.

vmware.vm.net.if.usage[url,uuid,<instance>]

 L'utilisation du réseau de la machine virtuelle VMware (débits d'émission et de réception combinés) pendant l'intervalle (KBps).
 Valeur de retour : *Integer*.

Paramètres :

- **url** - l'URL du service VMware ;
- **uuid** - l'identifiant global unique de la machine virtuelle VMware ;
- **instance** - l'instance de l'interface réseau.

vmware.vm.perfcounter[url,uuid,path,<instance>]

 La valeur du compteur de performance de la machine virtuelle VMware.
 Valeur de retour : *Integer* [2].

Paramètres :

- **url** - l'URL du service VMware ;
- **uuid** - l'identifiant global unique de la machine virtuelle VMware ;
- **path** - le chemin du compteur de performance [1] ;
- **instance** - l'instance du compteur de performance. Utilisez une instance vide pour les valeurs agrégées (par défaut).

vmware.vm.powerstate[url,uuid]

 L'état d'alimentation de la machine virtuelle VMware.
 Valeur de retour : *0* - poweredOff ; *1* - poweredOn ; *2* - suspended.

Paramètres :

- **url** - l'URL du service VMware ;
- **uuid** - l'identifiant global unique de la machine virtuelle VMware.

vmware.vm.property[url,uuid,prop]

 La propriété de la machine virtuelle VMware.
 Valeur de retour : *String*.

Paramètres :

- **url** - l'URL du service VMware ;
- **uuid** - l'identifiant global unique de la machine virtuelle VMware ;
- **prop** - le chemin de la propriété.

vmware.vm.snapshot.get[url,uuid]

 L'état du snapshot de la machine virtuelle VMware.
 Valeur de retour: objet JSON.

Paramètres:

- **url** - l'URL du service VMware;
- **uuid** - l'identifiant unique global de la machine virtuelle VMware.

vmware.vm.state[url,uuid]

 L'état de la machine virtuelle VMware.
 Valeur de retour : *String* : *notRunning*, *resetting*, *running*, *shuttingDown*, *standby* ou *unknown*.

Paramètres :

- **url** - l'URL du service VMware ;
- **uuid** - l'identifiant global unique de la machine virtuelle VMware.

vmware.vm.storage.committed[url,uuid]

 L'espace de stockage validé de la machine virtuelle VMware (octets).
 Valeur de retour : *Integer*.

Paramètres :

- **url** - l'URL du service VMware ;
- **uuid** - l'identifiant global unique de la machine virtuelle VMware.

vmware.vm.storage.readoio[url,uuid,instance]

 Le nombre moyen de requêtes de lecture en attente vers le disque virtuel pendant l'intervalle de collecte.
 Valeur de retour : *Integer*.

Paramètres :

- **url** - l'URL du service VMware ;

- **uuid** - l'identifiant global unique de la machine virtuelle VMware ;
- **instance** - l'instance du périphérique disque.

vmware.vm.storage.totalreadlatency[url,uuid,instance]

 Le temps moyen nécessaire pour effectuer une lecture depuis le disque virtuel (en millisecondes).
 Valeur de retour : *Integer*.

Paramètres :

- **url** - l'URL du service VMware ;
- **uuid** - l'identifiant global unique de la machine virtuelle VMware ;
- **instance** - l'instance du périphérique disque.

vmware.vm.storage.totalwritelatency[url,uuid,instance]

 Le temps moyen qu'une écriture sur le disque virtuel prend (en millisecondes).
 Valeur de retour : *Integer*.

Paramètres :

- **url** - l'URL du service VMware ;
- **uuid** - l'identifiant global unique de la machine virtuelle VMware ;
- **instance** - l'instance du périphérique de disque.

vmware.vm.storage.uncommitted[url,uuid]

 L'espace de stockage non validé de la machine virtuelle VMware (en octets).
 Valeur de retour : *Integer*.

Paramètres :

- **url** - l'URL du service VMware ;
- **uuid** - l'identifiant global unique de la machine virtuelle VMware.

vmware.vm.storage.unshared[url,uuid]

 L'espace de stockage non partagé de la machine virtuelle VMware (en octets).
 Valeur de retour : *Integer*.

Paramètres :

- **url** - l'URL du service VMware ;
- **uuid** - l'identifiant global unique de la machine virtuelle VMware.

vmware.vm.storage.writeoio[url,uuid,instance]

 Le nombre moyen de requêtes d'écriture en attente vers le disque virtuel pendant l'intervalle de collecte.
 Valeur de retour : *Integer*.

Paramètres :

- **url** - l'URL du service VMware ;
- **uuid** - l'identifiant global unique de la machine virtuelle VMware ;
- **instance** - l'instance du périphérique de disque.

vmware.vm.tags.get[url,uuid]

 Le tableau des balises de la machine virtuelle VMware.
 Valeur de retour: objet JSON.

Paramètres:

- **url** - l'URL du service VMware;
- **uuid** - l'identifiant unique global de la machine virtuelle VMware.

Cet élément fonctionne avec vSphere 6.5 et les versions ultérieures.

vmware.vm.tools[url,uuid,<mode>]

 L'état des outils invités de la machine virtuelle VMware.
 Valeur de retour : *String*.
 En mode *status* : *guestToolsExecutingScripts* - VMware Tools démarre ; *guestToolsNotRunning* - VMware Tools n'est pas en cours d'exécution ; *guestToolsRunning* - VMware Tools est en cours d'exécution.
 En mode *version* : version.

Paramètres :

- **url** - l'URL du service VMware ;
- **uuid** - l'identifiant global unique de la machine virtuelle VMware ;
- **mode** - *version*, *status* (par défaut).

vmware.vm.uptime[url,uuid]

 Le temps de fonctionnement de la machine virtuelle VMware (en secondes).
 Valeur de retour : *Integer*.

Paramètres :

- **url** - l'URL du service VMware ;
- **uuid** - l'identifiant global unique de la machine virtuelle VMware.

vmware.vm.vfs.dev.discovery[url,uuid]

 La découverte des périphériques de disque des machines virtuelles VMware.
 Valeur de retour : objet JSON.

Paramètres :

- **url** - l'URL du service VMware ;
- **uuid** - l'identifiant global unique de la machine virtuelle VMware.

vmware.vm.vfs.dev.read[url,uuid,instance,<mode>]

 Les statistiques de lecture du périphérique de disque de la machine virtuelle VMware (octets/opérations par seconde).
 Valeur de retour : *Integer* [2].

Paramètres :

- **url** - l'URL du service VMware ;
- **uuid** - l'identifiant global unique de la machine virtuelle VMware ;
- **instance** - l'instance du périphérique de disque ;
- **mode** - *bps* (par défaut) ou *ops* - octets ou opérations par seconde.

vmware.vm.vfs.dev.write[url,uuid,instance,<mode>]

 Les statistiques d'écriture du périphérique de disque de la machine virtuelle VMware (octets/opérations par seconde).
 Valeur de retour : *Integer* [2].

Paramètres :

- **url** - l'URL du service VMware ;
- **uuid** - l'identifiant global unique de la machine virtuelle VMware ;
- **instance** - l'instance du périphérique de disque ;
- **mode** - *bps* (par défaut) ou *ops* - octets ou opérations par seconde.

vmware.vm.vfs.fs.discovery[url,uuid]

 La découverte des systèmes de fichiers de la machine virtuelle VMware.
 Valeur de retour : objet JSON.

Paramètres :

- **url** - l'URL du service VMware ;
- **uuid** - l'identifiant global unique de la machine virtuelle VMware.

VMware Tools doit être installé sur la machine virtuelle invitée pour que cet élément fonctionne.

vmware.vm.vfs.fs.size[url,uuid,fsname,<mode>]

 Les statistiques du système de fichiers de la machine virtuelle VMware (octets/pourcentages).
 Valeur de retour : *Integer* - pour les octets ; *Float* - pour le pourcentage.

Paramètres :

- **url** - l'URL du service VMware ;
- **uuid** - l'identifiant global unique de la machine virtuelle VMware ;
- **fsname** - le nom du système de fichiers ;
- **mode** - *total*, *free*, *used*, *pfree* ou *pused*.

VMware Tools doit être installé sur la machine virtuelle invitée pour que cet élément fonctionne.

Notes de bas de page

[1] Voir Création de noms de compteurs de performance personnalisés pour VMware.

[2] La valeur de ces éléments est obtenue à partir des compteurs de performance VMware et le paramètre VMwarePerfFrequency est utilisé pour actualiser leurs données dans le cache VMware de Zabbix :

- vmware.cl.perfcounter
- vmware.hv.datastore.read
- vmware.hv.datastore.write

- vmware.hv.network.in
- vmware.hv.network.out
- vmware.hv.perfcounter
- vmware.vm.cpu.ready
- vmware.vm.net.if.in
- vmware.vm.net.if.out
- vmware.vm.perfcounter
- vmware.vm.vfs.dev.read
- vmware.vm.vfs.dev.write

Plus d'informations

Voir Surveillance des machines virtuelles pour des informations détaillées sur la configuration de Zabbix pour surveiller les environnements VMware.

Champs de découverte des machines virtuelles

Le tableau suivant répertorie les champs renvoyés par les clés de découverte liées aux machines virtuelles.

Clé d'élément		
Description	**Champ**	**Contenu récupéré**
vmware.cluster.discovery		
Effectue la découverte des clusters.	{#CLUSTER.ID}	Identifiant du cluster.
	{#CLUSTER.NAME}	Nom du cluster.

	"resource_pool"	Un tableau contenant les données du pool de ressources, notamment l'ID du groupe de ressources, le tableau des balises, le chemin du pool de ressources et le nombre de machines virtuelles.
		
		Structure du tableau :
		`[{`
		`"rpid":"resource group id",`
		`"tags":[{}],`
		`"rpath":"resource group path",`
		`"vm_count":0`
		`}]`
		
		Pour la structure du tableau `"tags"`, voir le champ "tags".

	"tags"	Un tableau contenant des balises avec le nom de la balise, la description et la catégorie.
		
		Structure du tableau :
		`[{`
		`"tag":"tag name",`
		`"tag_description":"tag description",`
		`"category":"tag category"`
		`}]`

vmware.datastore.discovery		
Effectue la découverte des datastores.	{#DATASTORE}	Nom du datastore.

	{#DATASTORE.EXTENT}	Un tableau contenant l'ID de partition de l'extent du datastore et le nom de l'instance.
		
		Structure du tableau :
		`[{`
		`"partitionid":1,`
		`"instance":"name"`
		`}]`

	{#DATASTORE.TYPE}	Type de datastore. Exemples de valeurs : `VMFS`, `NFS`, `vsan`, etc.

Clé d'élément		
	{#DATASTORE.UUID}	Identifiant du datastore.

	"tags"	Un tableau contenant des balises avec le nom de la balise, la description et la catégorie.
		
		Structure du tableau :
		`[{`
		`"tag":"tag name",`
		`"tag_description":"tag description",`
		`"category":"tag category"`
		`}]`

vmware.dc.discovery		
Effectue la découverte des datacenters.	{#DATACENTER}	Nom du datacenter.
	{#DATACENTERID}	Identifiant du datacenter.

	"tags"	Un tableau contenant des balises avec le nom de la balise, la description et la catégorie.
		
		Structure du tableau :
		`[{`
		`"tag":"tag name",`
		`"tag_description":"tag description",`
		`"category":"tag category"`
		`}]`

vmware.dvswitch.discovery		
Effectue la découverte des commutateurs distribués vSphere.	{#DVS.NAME}	Nom du commutateur.
	{#DVS.UUID}	Identifiant du commutateur.
vmware.hv.discovery		
Effectue la découverte des hyperviseurs.	{#HV.UUID}	Identifiant unique de l'hyperviseur.
	{#HV.ID}	Identifiant de l'hyperviseur (nom de l'objet géré HostSystem).
	{#HV.NAME}	Nom de l'hyperviseur.
	{#HV.NETNAME}	Nom d'hôte réseau de l'hyperviseur.
	{#HV.IP}	Adresse IP de l'hyperviseur, peut être vide. Dans le cas d'une configuration HA avec plusieurs interfaces réseau, la priorité de sélection suivante est appliquée pour l'interface : - privilégier l'IP qui partage le sous-réseau IP avec l'IP de vCenter ; - privilégier l'IP du sous-réseau IP avec la passerelle par défaut ; - privilégier l'IP de l'interface avec l'ID le plus faible.
	{#CLUSTER.NAME}	Nom du cluster, peut être vide.
	{#DATACENTER.NAME}	Nom du datacenter.
	{#PARENT.NAME}	Nom du conteneur qui stocke l'hyperviseur.
	{#PARENT.TYPE}	Type de conteneur dans lequel l'hyperviseur est stocké. Les valeurs peuvent être `Datacenter`, `Folder`, `ClusterComputeResource`, `VMware`, où "VMware" désigne un type de conteneur inconnu.

	"resource_pool"	Un tableau contenant les données du pool de ressources, notamment l'ID du groupe de ressources, le tableau des balises, le chemin du pool de ressources et le nombre de machines virtuelles.
		
		Structure du tableau :
		`[{`
		`"rpid":"resource group id",`
		`"tags":[{}],`
		`"rpath":"resource group path",`
		`"vm_count":0`
		`}]`
		
		Pour la structure du tableau `"tags"`, voir le champ "tags".

Clé d'élément		

	"tags"	Un tableau contenant des balises avec le nom de la balise, la description et la catégorie.
		
		Structure du tableau :
		`[{`
		`"tag":"tag name",`
		`"tag_description":"tag description",`
		`"category":"tag category"`
		`}]`

vmware.hv.datastore.discovery		
Effectue la découverte des datastores de l'hyperviseur. Notez que plusieurs hyperviseurs peuvent utiliser le même datastore.	{#DATASTORE}	Nom du datastore.
	{#DATASTORE.TYPE}	Type de datastore. Exemples de valeurs : `VMFS`, `NFS`, `vsan`, etc.
	{#DATASTORE.UUID}	Identifiant du datastore.
	{#MULTIPATH.COUNT}	Nombre enregistré de chemins du datastore.
	{#MULTIPATH.PARTITION.COUNT}	Nombre de partitions de disque disponibles.

	"datastore_extent"	Un tableau contenant le nom de l'instance de l'extent du datastore et l'ID de partition.
		
		Structure du tableau :
		`[{`
		`"partitionid":1,`
		`"instance":"name"`
		`}]`

	"tags"	Un tableau contenant des balises avec le nom de la balise, la description et la catégorie.
		
		Structure du tableau :
		`[{`
		`"tag":"tag name",`
		`"tag_description":"tag description",`
		`"category":"tag category"`
		`}]`

vmware.hv.net.if.discovery		
Effectue la découverte des interfaces réseau de l'hyperviseur.	{#IFNAME}	Nom de l'interface.
	{#IFDRIVER}	Pilote de l'interface.
	{#IFDUPLEX}	Paramètres duplex de l'interface.
	{#IFSPEED}	Vitesse de l'interface.
	{#IFMAC}	Adresse MAC de l'interface.
vmware.vm.discovery		
Effectue la découverte des machines virtuelles.	{#VM.UUID}	Identifiant unique de la machine virtuelle.
	{#VM.ID}	Identifiant de la machine virtuelle (nom de l'objet géré VirtualMachine).
	{#VM.NAME}	Nom de la machine virtuelle.
	{#HV.NAME}	Nom de l'hyperviseur.
	{#HV.UUID}	Identifiant unique de l'hyperviseur.
	{#HV.ID}	Identifiant de l'hyperviseur (nom de l'objet géré HostSystem).
	{#CLUSTER.NAME}	Nom du cluster, peut être vide.
	{#DATACENTER.NAME}	Nom du datacenter.
	{#DATASTORE.NAME}	Nom du datastore.
	{#DATASTORE.UUID}	Identifiant du datastore.

Clé d'élément		
	{#VM.IP}	Adresse IP de la machine virtuelle, peut être vide.
	{#VM.DNS}	Nom DNS de la machine virtuelle, peut être vide.
	{#VM.GUESTFAMILY}	Famille du système d'exploitation invité de la machine virtuelle, peut être vide.
	{#VM.GUESTFULLNAME}	Nom complet du système d'exploitation invité de la machine virtuelle, peut être vide.
	{#VM.FOLDER}	Chaîne des dossiers parents de la machine virtuelle, peut être utilisée comme valeur pour des groupes imbriqués ; les noms des dossiers sont combinés avec "/". Peut être vide.
	{#VM.TOOLS.STATUS}	État des outils de la machine virtuelle VMware.
	{#VM.POWERSTATE}	État d'alimentation de la machine virtuelle VMware (`poweredOff`, `poweredOn`, ou `suspended`).
	{#VM.RPOOL.ID}	Identifiant du pool de ressources.
	{#VM.RPOOL.PATH}	Chemin complet du pool de ressources, à l'exclusion du nom racine "Resources". Les noms des dossiers sont combinés avec "/".
	{#VM.SNAPSHOT.COUNT}	Nombre de snapshots de la machine virtuelle.

	"tags"	Un tableau contenant des balises avec le nom de la balise, la description et la catégorie.
		
		Structure du tableau :
		`[{`
		`"tag":"tag name",`
		`"tag_description":"tag description",`
		`"category":"tag category"`
		`}]`

	"vm_customattribute"	Un tableau des attributs personnalisés de la machine virtuelle (s'ils sont définis).
		
		Structure du tableau :
		`[{`
		`"name":"custom field name",`
		`"value":"custom field value"`
		`}]`

	"net_if"	Un tableau des interfaces réseau de la machine virtuelle.
		
		Structure du tableau :
		`[{`
		`"ifname": "interface name",`
		`"ifdesc": "interface description",`
		`"ifmac": "00:00:00:00:00:00",`
		`"ifconnected": true,`
		`"iftype": "interface type",`
		`"ifbackingdevice": "interface backing device",`
		`"ifdvswitch_uuid": "interface switch uuid",`
		`"ifdvswitch_portgroup": "interface switch port group",`
		`"ifdvswitch_port": "interface switch port",`
		`"ifip": ["interface ip addresses"]`
		`}]`
		
		Pour la description des données renvoyées, voir la clé d'élément "vmware.vm.net.if.discovery".

vmware.vm.net.if.discovery		
Effectue la découverte des interfaces réseau de la machine virtuelle.	{#IFNAME}	Nom de l'interface réseau.
	{#IFDESC}	Description de l'interface.
	{#IFMAC}	Adresse MAC de l'interface.
	{#IFCONNECTED}	État de connexion de l'interface (`false` - déconnectée ; `true` - connectée).
	{#IFTYPE}	Type de l'interface.

Clé d'élément		
	{#IFBACKINGDEVICE}	Nom du périphérique de support.
	{#IFDVSWITCH.UUID}	Identifiant unique du Distributed Switch vSphere.
	{#IFDVSWITCH.PORTGROUP}	Groupe de ports distribué.
	{#IFDVSWITCH.PORT}	Port du Distributed Switch vSphere.
	"ifip"	Un tableau des adresses de l'interface.
vmware.vm.vfs.dev.discovery		
Effectue la découverte des périphériques de disque de la machine virtuelle.	{#DISKNAME}	Nom du périphérique de disque.
vmware.vm.vfs.fs.discovery		
Effectue la découverte du système de fichiers de la machine virtuelle.	{#FSNAME}	Nom du système de fichiers.

Exemples JSON pour les éléments VMware

Vue d'ensemble Cette section fournit des informations supplémentaires sur les objets JSON renvoyés par divers éléments VMware.

vmware.*.alarms.get Les éléments **vmware.alarms.get[]**, **vmware.cluster.alarms.get[]**, **vmware.datastore.alarms.get[]**, **vmware.dc.alarms.get[]**, **vmware.hv.alarms.get[]**, **vmware.vm.alarms.get[]** renvoient des objets JSON avec la structure suivante (les valeurs sont fournies à titre d'exemple) :


```
{
    "alarms": [
        {
            "name": "Host connection and power state",
            "system_name": "alarm.HostConnectionStateAlarm",
            "description": "Default alarm to monitor host connection and power state",
            "enabled": true,
            "key": "alarm-1.host-2013",
            "time": "2022-06-27T05:27:38.759976Z",
            "overall_status": "red",
            "acknowledged": false,
            "entity": {
                "id": "host-2013",
                "uuid": "4201b707-f737-7cf8-881f-49215d1aad2a",
                "type": "HostSystem",
                "name": "esx-riga-01.prod.local"
            }
        },
        {
            "name": "Host memory usage",
            "system_name": "alarm.HostMemoryUsageAlarm",
            "description": "Default alarm to monitor host memory usage",
            "enabled": true,
            "key": "alarm-4.host-1004",
            "time": "2022-05-16T13:32:42.47863Z",
            "overall_status": "yellow",
            "acknowledged": false,
            "entity": {
                "id": "host-1004",
                "uuid": "5315c818-g848-8dg9-992g-50326e2bbe3a",
                "type": "HostSystem",
                "name": "esx-riga-02.prod.local",
            }
        },
```

```
        {
            // other alarms
        }
    ]
}
```

vmware.*.tags.get Les éléments **vmware.cluster.tags.get[]**, **vmware.datastore.tags.get[]**, **vmware.dc.tags.get[]**, **vmware.hv.tags.get[]**, **vmware.vm.tags.get[]** renvoient des objets JSON ayant la structure suivante (les valeurs sont fournies à titre d'exemple) :

```
{
    "tags": [
        {
            "name": "Windows",
        "description": "balise pour la catégorie type d'OS",
        "category": "type d'OS"
    },
    {
            "name": "SQL Server",
        "description": "balise pour la catégorie nom de l'application",
        "category": "nom de l'application"
        },
    {
            // autres balises
        }
    ]
}
```

vmware.hv.diskinfo.get The item **vmware.hv.diskinfo.get[]** returns JSON objects with the following structure (values are provided as an example):

```
[
  {
    "instance": "mpx.vmhba32:C0:T0:L0",
    "hv_uuid": "8002299e-d7b9-8728-d224-76004bbb6100",
    "datastore_uuid": "",
    "operational_state": [
      "ok"
    ],
    "lun_type": "disk",
    "queue_depth": 1,
    "model": "USB DISK",
    "vendor": "SMI Corp",
    "revision": "1100",
    "serial_number": "CCYYMMDDHHmmSS9S62CK",
    "vsan": {}
  },
  {
    // other instances
  }
]
```

vmware.dvswitch.fetchports.get L'élément **vmware.dvswitch.fetchports.get[]** renvoie des objets JSON ayant la structure suivante (les valeurs sont fournies à titre d'exemple) :

```
{
    "FetchDVPortsResponse":
    {
        "returnval": [
            {
                "key": "0",
                "dvsUuid": "50 36 6a 24 25 c0 10 9e-05 4a f6 ea 4e 3d 09 88",
                "portgroupKey": "dvportgroup-2023",
```

```
            "proxyHost":
            {
                "@type": "HostSystem",
                "#text": "host-2021"
            },
            "connectee":
            {
                "connectedEntity":
                {
                    "@type": "HostSystem",
                    "#text": "host-2021"
                },
                "nicKey": "vmnic0",
                "type": "pnic"
            },
            "conflict": "false",
            "state":
            {
                "runtimeInfo":
                {
                    "linkUp": "true",
                    "blocked": "false",
                    "vlanIds":
                    {
                        "start": "0",
                        "end": "4094"
                    },
                    "trunkingMode": "true",
                    "linkPeer": "vmnic0",
                    "macAddress": "00:00:00:00:00:00",
                    "statusDetail": null,
                    "vmDirectPathGen2Active": "false",
                    "vmDirectPathGen2InactiveReasonOther": "portNptIncompatibleConnectee"
                },
                "stats":
                {
                    "packetsInMulticast": "2385470",
                    "packetsOutMulticast": "45",
                    "bytesInMulticast": "309250248",
                    "bytesOutMulticast": "5890",
                    "packetsInUnicast": "155601537",
                    "packetsOutUnicast": "113008658",
                    "bytesInUnicast": "121609489384",
                    "bytesOutUnicast": "47240279759",
                    "packetsInBroadcast": "1040420",
                    "packetsOutBroadcast": "7051",
                    "bytesInBroadcast": "77339771",
                    "bytesOutBroadcast": "430392",
                    "packetsInDropped": "0",
                    "packetsOutDropped": "0",
                    "packetsInException": "0",
                    "packetsOutException": "0"
                }
            },
            "connectionCookie": "1702765133",
            "lastStatusChange": "2022-03-25T14:01:11Z",
            "hostLocalPort": "false"
        },
        {
            //autres clés
        }
    ]
```

```
    }
}
```

vmware.hv.hw.sensors.get L'élément **vmware.hv.hw.sensors.get[]** renvoie des objets JSON ayant la structure suivante (les valeurs sont fournies à titre d'exemple) :

```
{
    "val":
        {
            "@type": "HostHardwareStatusInfo",
            "storageStatusInfo": [
                {
                    "name": "Intel Corporation HD Graphics 630 #2",
                "status":
                        {
                            "label": "Unknown",
                    "summary": "Cannot report on the current status of the physical element",
                    "key": "Unknown"
                }
            },
            {
                    "name": "Intel Corporation 200 Series/Z370 Chipset Family USB 3.0 xHCI Controller #20"
                "status":
                        {
                     "label": "Unknown",
                     "summary": "Cannot report on the current status of the physical element",
                     "key": "Unknown"
                 }
            },
            {
                    // autres capteurs matériels hv
                }
        ]
        }
}
```

vmware.hv.sensors.get L'élément **vmware.hv.sensors.get[]** renvoie des objets JSON ayant la structure suivante (les valeurs sont fournies à titre d'exemple) :

```
{
    "val":
        {
            "@type": "ArrayOfHostNumericSensorInfo", "HostNumericSensorInfo": [
                {
                    "@type": "HostNumericSensorInfo",
                    "name": "System Board 1 PwrMeter Output --- Normal",
                    "healthState":
                        {
                            "label": "Green",
                            "summary": "Sensor is operating under normal conditions",
                    "key": "green"
                },
                    "currentReading": "10500",
                "unitModifier": "-2",
                "baseUnits": "Watts",
                "sensorType": "other"
            },
                {
                    "@type": "HostNumericSensorInfo",
                    "name": "Power Supply 1 PS 1 Output --- Normal",
                    "healthState":
                        {
                            "label": "Green",
```

```
                    "summary": "Sensor is operating under normal conditions",
                    "key": "green"
                },
                    "currentReading": "10000",
            "unitModifier": "-2",
            "baseUnits": "Watts",
            "sensorType": "power"
        },
            {
                    // autres capteurs hv
                }
        ]
        }
}
```

vmware.vm.snapshot.get Si des snapshots existent, l'élément **vmware.snapshot.get[]** renvoie un objet JSON avec la structure suivante (les valeurs sont fournies à titre d'exemple) :

```
{
  "snapshot": [
    {
      "name": "VM Snapshot 4%2f1%2f2022, 9:16:39 AM",
      "description": "Descr 1",
      "createtime": "2022-04-01T06:16:51.761Z",
      "size": 5755795171,
      "uniquesize": 5755795171
    },
    {
      "name": "VM Snapshot 4%2f1%2f2022, 9:18:21 AM",
      "description": "Descr 2",
      "createtime": "2022-04-01T06:18:29.164999Z",
      "size": 118650595,
      "uniquesize": 118650595
    },
    {
      "name": "VM Snapshot 4%2f1%2f2022, 9:37:29 AM",
      "description": "Descr 3",
      "createtime": "2022-04-01T06:37:53.534999Z",
      "size": 62935016,
      "uniquesize": 62935016
    }
  ],
  "count": 3,
  "latestdate": "2022-04-01T06:37:53.534999Z",
  "latestage": 22729203,
  "oldestdate": "2022-04-01T06:16:51.761Z",
  "oldestage": 22730465,
  "size": 5937380782,
  "uniquesize": 5937380782
}
```

Si aucun snapshot n'existe, l'élément **vmware.snapshot.get[]** renvoie un objet JSON avec des valeurs vides :

```
{
  "snapshot": [],
  "count": 0,
  "latestdate": null,
  "latestage": 0,
  "oldestdate": null,
  "oldestage": 0,
  "size": 0,
  "uniquesize": 0
}
```

Exemple de configuration de surveillance VMware

Vue d'ensemble

L'exemple suivant décrit comment configurer Zabbix pour surveiller des machines virtuelles VMware. Cela implique :

- la création d'un hôte représentant votre environnement VMware ;
- la création d'une règle de découverte de bas niveau qui découvre les machines virtuelles dans votre environnement VMware ;
- la création d'un prototype d'hôte, sur la base duquel Zabbix générera de véritables hôtes pour les machines virtuelles découvertes par la règle de découverte de bas niveau.

Prérequis

> **Note:**
> Cet exemple ne couvre pas la configuration de VMware. Il est supposé que VMware est déjà configuré.

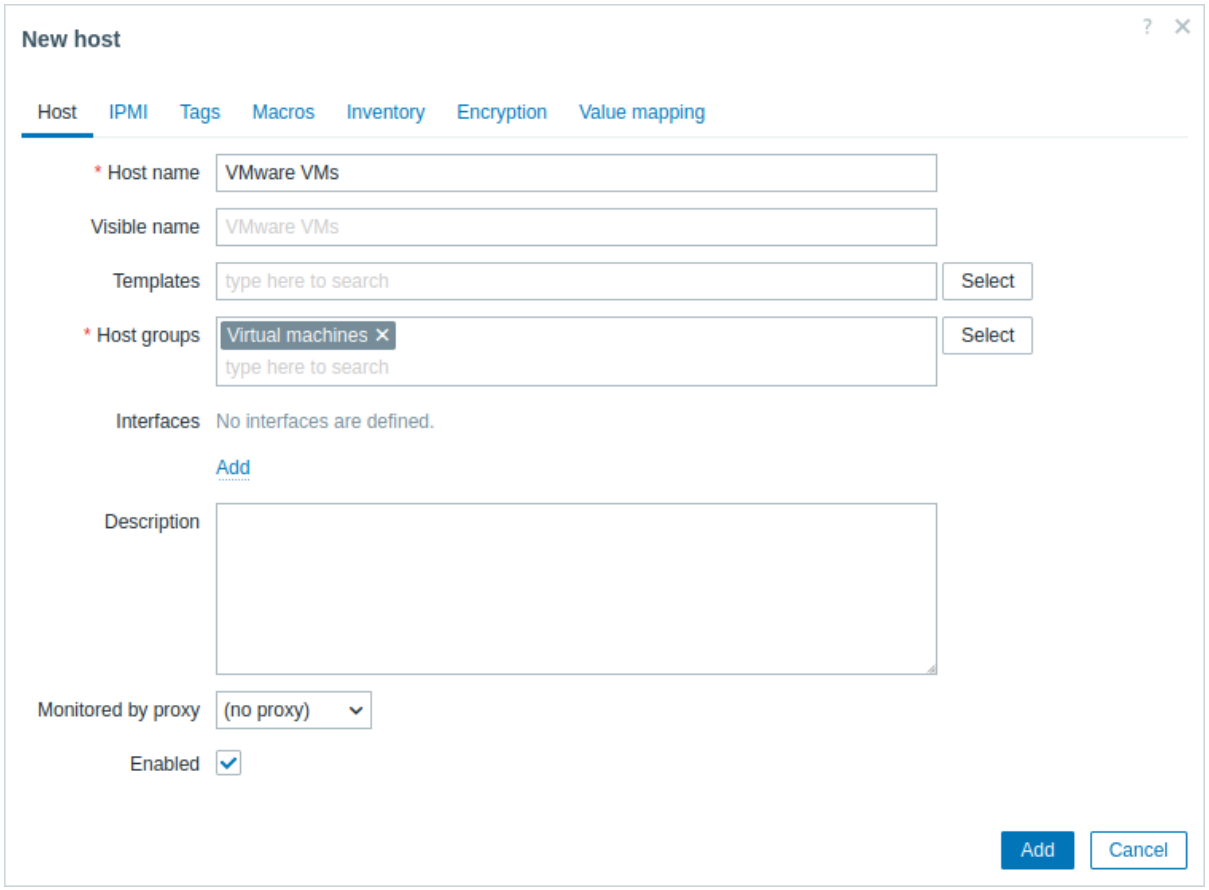
Avant de continuer, définissez le paramètre `StartVMwareCollectors` dans le fichier de configuration du serveur Zabbix sur 2 ou plus (la valeur par défaut est 0).

Créer un hôte

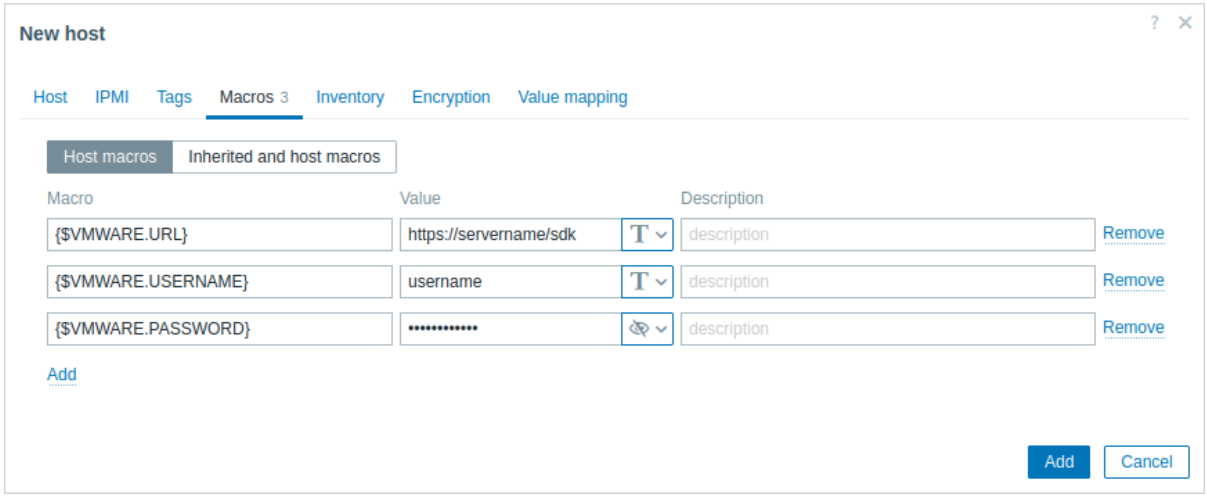
1. Accédez à *Collecte de données* > *Hôtes*.

2. Créez un hôte :

- Dans le champ *Nom de l'hôte*, saisissez un nom d'hôte (par exemple, "VMware VMs").
- Dans le champ *Groupes d'hôtes*, saisissez ou sélectionnez un groupe d'hôtes (par exemple, "Machines virtuelles").



- Dans l'onglet *Macros*, définissez les macros d'hôte suivantes :
  - {$VMWARE.URL} - URL SDK du service VMware (hyperviseur ESXi) (https://servername/sdk)
  - {$VMWARE.USERNAME} - nom d'utilisateur du service VMware
  - {$VMWARE.PASSWORD} - mot de passe de l'utilisateur {$VMWARE.USERNAME} du service VMware



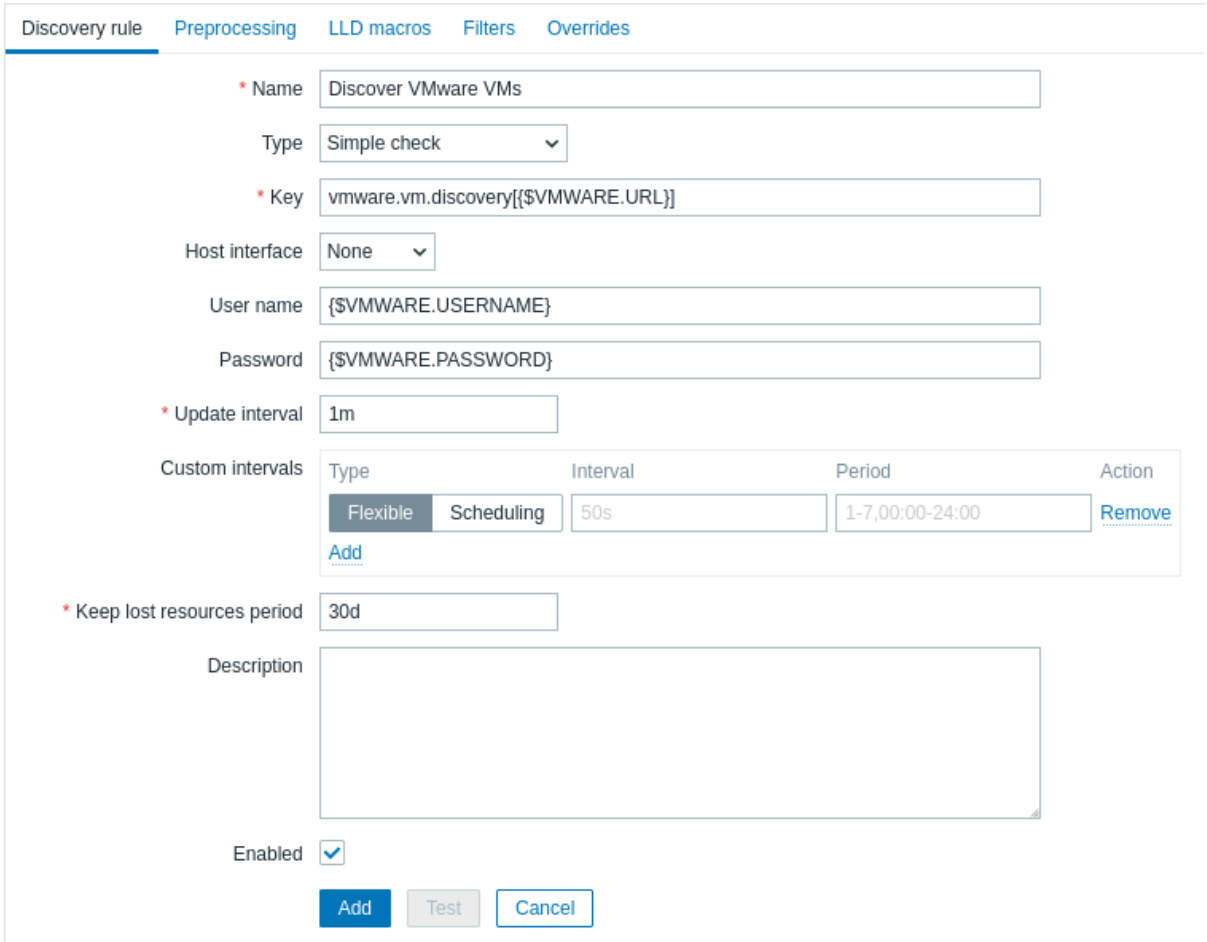
3. Cliquez sur le bouton *Ajouter* pour créer l'hôte. Cet hôte représentera votre environnement VMware.

Créer une règle de découverte de bas niveau

1. Cliquez sur *Découverte* pour l'hôte créé afin d'accéder à la liste des règles de découverte de bas niveau pour cet hôte.

2. Créez une règle de découverte de bas niveau :

- Dans le champ *Nom*, saisissez un nom de règle de découverte de bas niveau (par exemple, « Découvrir les VM VMware »).
- Dans le champ *Type*, sélectionnez « Simple check ».
- Dans le champ *Clé*, saisissez la clé d'élément intégrée pour découvrir les machines virtuelles VMware : `vmware.vm.discovery[{$VMW`
- Dans les champs *Nom d'utilisateur* et *Mot de passe*, saisissez les macros correspondantes précédemment configurées sur l'hôte.



3. Cliquez sur le bouton *Ajouter* pour créer la règle de découverte de bas niveau. Cette règle de découverte détectera les machines virtuelles dans votre environnement VMware.

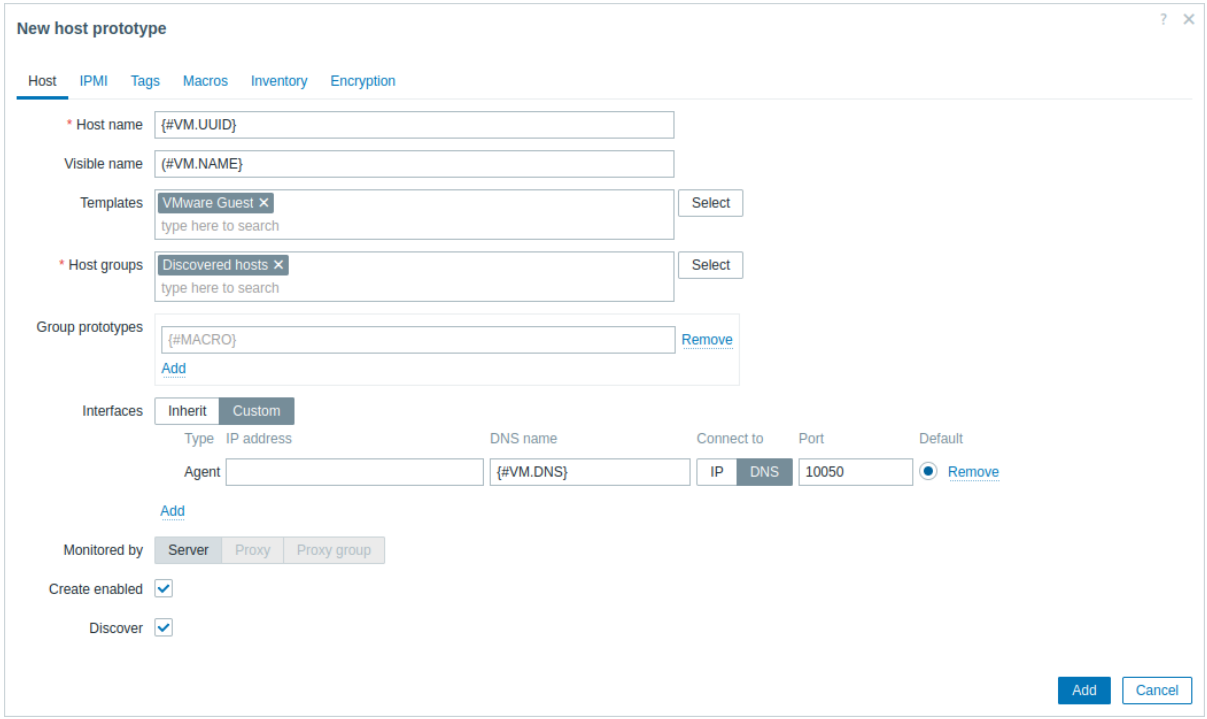
Créer un prototype d'hôte

1. Dans la liste des règles de découverte de bas niveau, cliquez sur *Prototypes d'hôte* pour la règle de découverte de bas niveau

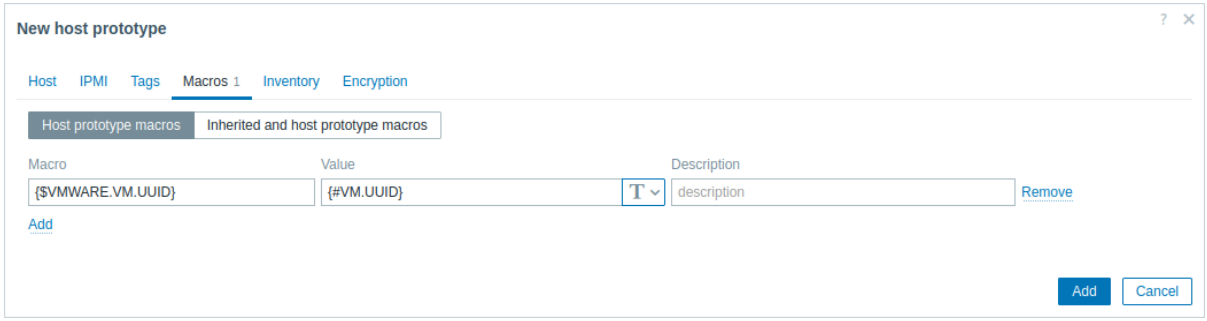
créée précédemment.

2. Créez un prototype d'hôte. Comme les prototypes d'hôte servent de modèles pour créer des hôtes via des règles de découverte de bas niveau, la plupart des champs contiendront des macros de découverte de bas niveau. Cela garantit que les hôtes sont créés avec des propriétés basées sur le contenu récupéré par la règle de découverte de bas niveau créée précédemment.

- Dans le champ *Nom d'hôte*, saisissez la macro {#VM.UUID}.
- Dans le champ *Nom visible*, saisissez la macro {#VM.NAME}.
- Dans le champ *Modèles*, saisissez ou sélectionnez le modèle "VMware Guest". Ce modèle contient des éléments VMware et des règles de découverte pour surveiller l'état d'alimentation d'une machine virtuelle, l'utilisation du processeur, l'utilisation de la mémoire, les périphériques réseau, etc.
- Dans le champ *Groupes d'hôtes*, saisissez ou sélectionnez un groupe d'hôtes (par exemple, "Discovered hosts").
- Dans le champ *Interfaces*, ajoutez une interface d'hôte personnalisée. Ensuite, saisissez la macro {#VM.DNS} dans le champ *Nom DNS*, ou saisissez la macro {#VM.IP} dans le champ *Adresse IP*. Sinon, si les machines virtuelles de votre environnement VMware possèdent plusieurs interfaces, passez à la section *Configuration avancée de l'interface d'hôte*. La configuration d'une interface d'hôte personnalisée est nécessaire au bon fonctionnement du modèle *VMware Guest*.



- Dans l'onglet *Macros*, définissez la macro {$VMWARE.VM.UUID} avec la valeur {#VM.UUID}. Cela est nécessaire au bon fonctionnement du modèle *VMware Guest*, qui utilise cette macro comme macro utilisateur au niveau de l'hôte dans les paramètres des éléments (par exemple, *vmware.vm.net.if.discovery[{$VMWARE.URL}, {$VMWARE.VM.UUID}]*).

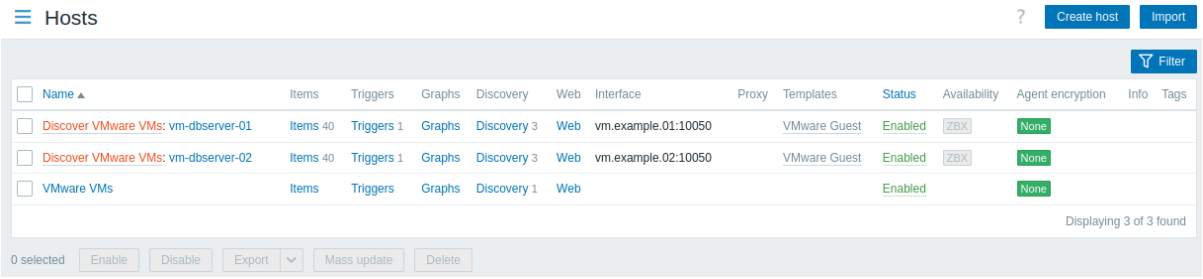


3. Cliquez sur le bouton *Ajouter* pour créer le prototype d'hôte. Ce prototype d'hôte sera utilisé pour créer des hôtes pour les machines virtuelles découvertes par la règle de découverte de bas niveau créée précédemment.

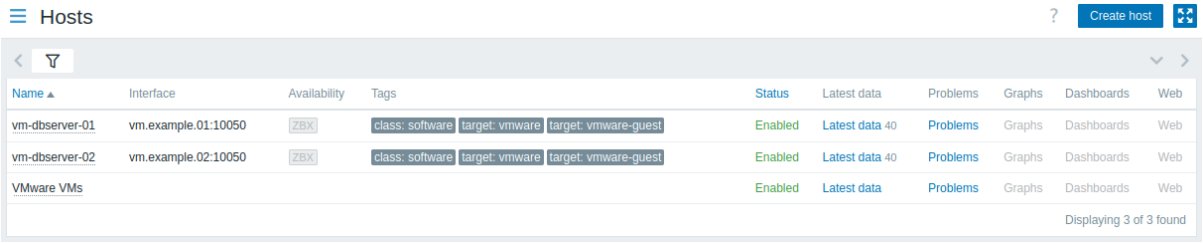
Afficher les hôtes et les métriques

Après la création du prototype d'hôte, la règle de découverte bas niveau créera des hôtes pour les machines virtuelles VMware découvertes, et Zabbix commencera à les surveiller. Notez que la découverte et la création des hôtes peuvent également être exécutées manuellement, si nécessaire.

Pour afficher les hôtes créés, accédez à la section de menu *Collecte de données > Hôtes*.



Pour afficher les métriques collectées, accédez à la section de menu *Surveillance > Hôtes* et cliquez sur *Dernières données* pour l'un des hôtes.



Configuration avancée de l'interface d'hôte

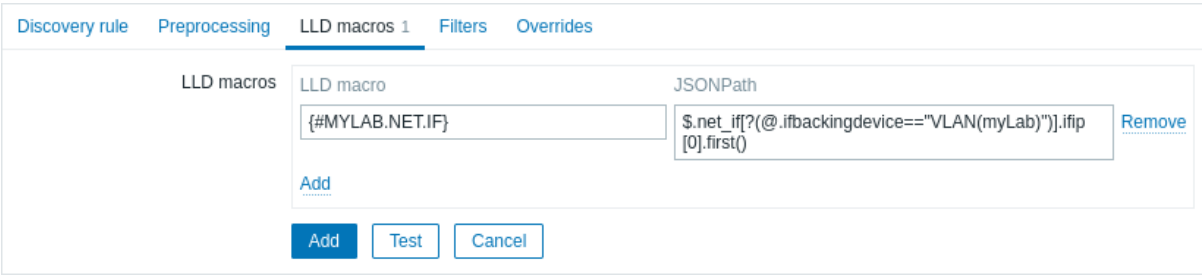
La clé d'élément `vmware.vm.discovery[{$VMWARE.URL}]`, configurée dans la section *Créer une règle de découverte de bas niveau*, renvoie les données des interfaces réseau dans le champ "net_if" :

```
"net_if": [
      {
        "ifname": "5000",
        "ifdesc": "Network adapter 1",
        "ifmac": "00:11:22:33:44:55",
        "ifconnected": true,
        "iftype": "VirtualVmxnet3",
        "ifbackingdevice": "VLAN(myLab)",
        "ifdvswitch_uuid": "",
        "ifdvswitch_portgroup": "",
        "ifdvswitch_port": "",
        "ifip": [
          "127.0.0.1",
          "::1"
        ]
      },
      {
        "ifname": "5001",
        "ifdesc": "Network adapter 2",
        "ifmac": "00:11:22:33:44:55",
        "ifconnected": false,
        "iftype": "VirtualVmxnet3",
        "ifbackingdevice": "VLAN(myLab2)",
        "ifdvswitch_uuid": "",
        "ifdvswitch_portgroup": "",
        "ifdvswitch_port": "",
        "ifip": []
      }
    ]
```

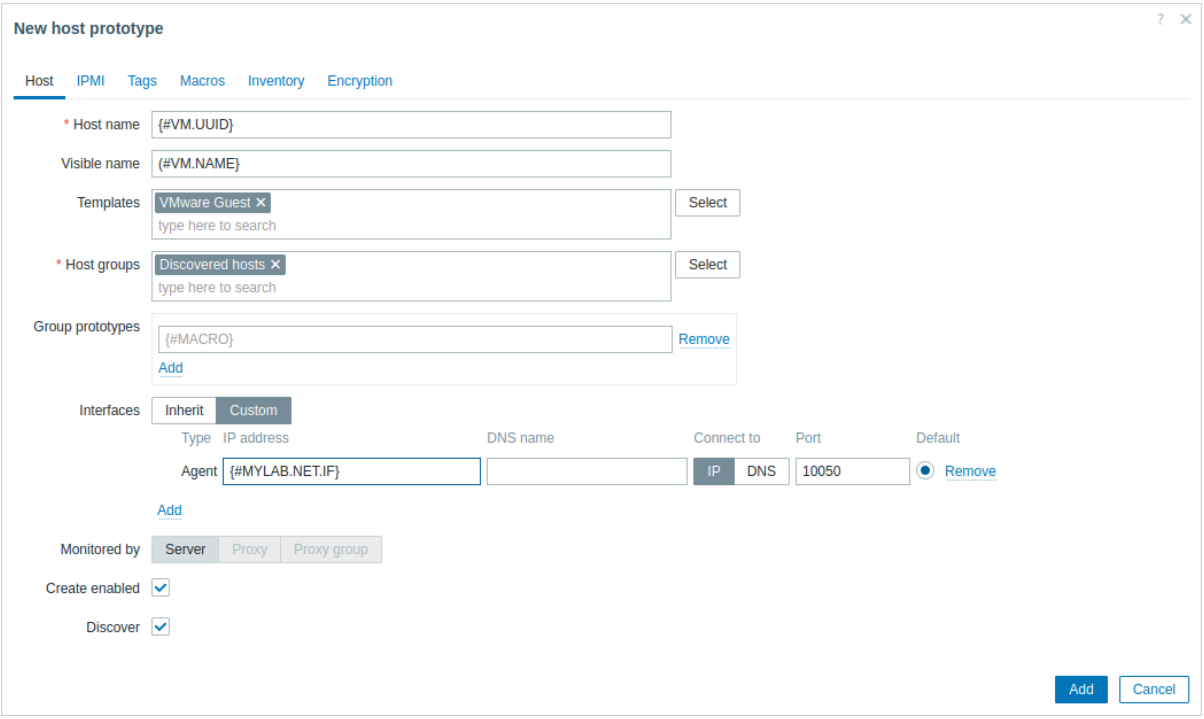
Ces données peuvent être utilisées pour configurer une interface d'hôte personnalisée.

1. Lors de la création d'une règle de découverte de bas niveau, configurez également une macro de découverte de bas niveau. Dans l'onglet *Macros LLD*, créez une macro LLD personnalisée avec une valeur JSONPath. Par exemple :

- `{#MYLAB.NET.IF}` - `$.net_if[?(@.ifbackingdevice=="VLAN(myLab)")].ifip[0].first()`



2. Lors de la création d'un prototype d'hôte, ajoutez une interface d'hôte personnalisée et saisissez la macro LLD dans le champ *Nom DNS* ou *Adresse IP*.



Surveillance web

Aperçu Avec Zabbix, vous pouvez vérifier plusieurs aspects de la disponibilité des sites web.

> **Attention:**
> Pour effectuer la surveillance web, le serveur Zabbix doit être initialement configuré avec la prise en charge de cURL (libcurl).

Pour activer la surveillance web, vous devez définir des scénarios web. Un scénario web se compose d'une ou plusieurs requêtes HTTP ou "étapes". Les étapes sont exécutées périodiquement par le serveur Zabbix dans un ordre prédéfini. Si un hôte est surveillé par un proxy, les étapes sont exécutées par le proxy.

Les scénarios web sont associés aux hôtes/modèles de la même manière que les éléments, déclencheurs, etc. Cela signifie que les scénarios web peuvent également être créés au niveau d'un modèle, puis appliqués à plusieurs hôtes en une seule opération.

Les informations suivantes sont collectées dans tout scénario web :

- vitesse moyenne de téléchargement par seconde pour toutes les étapes de l'ensemble du scénario
- numéro de l'étape ayant échoué
- dernier message d'erreur

Les informations suivantes sont collectées dans toute étape d'un scénario web :

- vitesse de téléchargement par seconde
- temps de réponse
- code de réponse

Pour plus de détails, voir éléments de surveillance web.

Les données collectées lors de l'exécution des scénarios web sont conservées dans la base de données. Ces données sont automatiquement utilisées pour les graphiques, les déclencheurs et les notifications.

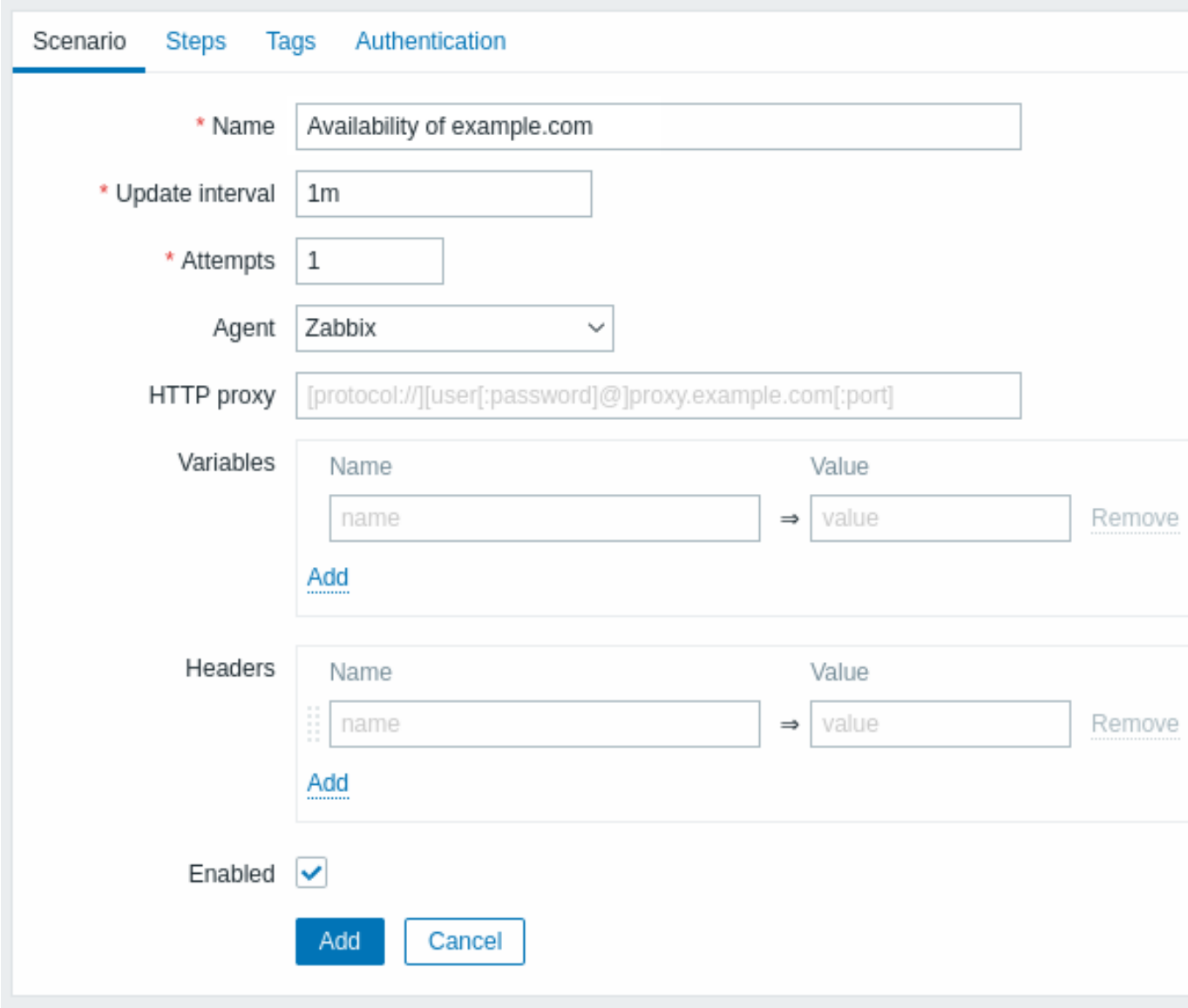
Zabbix peut également vérifier si une page HTML récupérée contient une chaîne prédéfinie. Il peut exécuter une connexion simulée et suivre un chemin de clics de souris simulés sur la page.

La surveillance web de Zabbix prend en charge HTTP et HTTPS. Lors de l'exécution d'un scénario web, Zabbix suivra éventuellement les redirections (voir l'option *Suivre les redirections* ci-dessous). Le nombre maximal de redirections est fixé en dur à 10 (à l'aide de l'option cURL CURLOPT_MAXREDIRS). Tous les cookies sont conservés pendant l'exécution d'un scénario unique.

Configuration d'un scénario web Pour configurer un scénario web :

1. Accédez à : *Collecte de données* > *Hôtes* (ou *Modèles*).
2. Cliquez sur *Web* dans la ligne de l'hôte/du modèle.
3. Cliquez sur *Créer un scénario web* à droite (ou sur le nom du scénario pour modifier un scénario existant).
4. Saisissez les paramètres du scénario dans le formulaire.

L'onglet **Scénario** permet de configurer les paramètres généraux d'un scénario web.



Tous les champs obligatoires sont signalés par un astérisque rouge.

Paramètres du scénario :

Paramètre	Description
Nom	Nom unique du scénario. Les macros utilisateur sont prises en charge. *Notez* que si des macros utilisateur sont utilisées, ces macros resteront non résolues dans les noms des éléments de supervision web.

Paramètre	Description
Intervalle de mise à jour	Fréquence d'exécution du scénario. Les suffixes temporels sont pris en charge, par exemple : 30s, 1m, 2h, 1d. Les macros utilisateur sont prises en charge. *Notez* que si une macro utilisateur est utilisée et que sa valeur est modifiée (par exemple, 5m → 30s), la prochaine vérification sera exécutée selon la valeur précédente (plus tard dans le futur avec les valeurs de l'exemple). Les nouveaux scénarios web seront vérifiés dans les 60 secondes suivant leur création.
Tentatives	Nombre de tentatives d'exécution des étapes du scénario web. En cas de problèmes réseau (délai d'attente, absence de connectivité, etc.), Zabbix peut répéter l'exécution d'une étape plusieurs fois. La valeur définie s'applique de la même manière à chaque étape du scénario. Jusqu'à 10 tentatives peuvent être spécifiées ; la valeur par défaut est 1. *Note* : Zabbix ne répétera pas une étape en raison d'un code de réponse incorrect ou de l'absence d'une chaîne requise.
Agent	Sélectionnez un agent client. Zabbix simulera le navigateur sélectionné. Cette option est utile lorsqu'un site web renvoie un contenu différent selon le navigateur utilisé. Des macros utilisateur peuvent être utilisées dans ce champ.
Proxy HTTP	Vous pouvez spécifier un proxy HTTP à utiliser, au format `[protocol://][username[:password]@]proxy.example.com[:port]`. Cette option définit l'option cURL CURLOPT_PROXY. Le préfixe facultatif `protocol://` peut être utilisé pour spécifier d'autres protocoles de proxy (la prise en charge du préfixe de protocole a été ajoutée dans cURL 7.21.7). Si aucun protocole n'est spécifié, le proxy sera traité comme un proxy HTTP. Par défaut, le port 1080 sera utilisé. Si le proxy est spécifié, il remplacera les variables d'environnement liées au proxy, telles que http_proxy et HTTPS_PROXY. S'il n'est pas spécifié, il ne remplacera pas les variables d'environnement liées au proxy. La valeur saisie est transmise « telle quelle », sans aucune vérification de cohérence. Vous pouvez également saisir l'adresse d'un proxy SOCKS. Si vous spécifiez un protocole incorrect, la connexion échouera et l'élément deviendra non pris en charge. *Notez* que seule l'authentification simple est prise en charge avec un proxy HTTP. Des macros utilisateur peuvent être utilisées dans ce champ.
Variables	Variables pouvant être utilisées dans les étapes du scénario (URL, variables post). Elles utilisent le format suivant : **{macro1}**=value1 **{macro2}**=value2 **{macro3}**=regex:<regular expression> **{macro4}**=jsonpath:<jsonpath> **{macro5}**=xmlxpath:<xmlxpath> **{macro6}**={{macro}.function()} (voir fonctions de macro) Par exemple : {username}=Alexei {password}=kj3h5kJ34bd {hostid}=regex:hostid is ([0-9]+) {url}=jsonpath:$.host_url {status}=xmlxpath://host/response/status {newvar}={{myvar}.btoa()} Les macros peuvent ensuite être référencées dans les étapes sous la forme {username}, {password}, {hostid}, etc. Zabbix les remplacera automatiquement par leurs valeurs réelles. Notez que les variables avec `regex:` nécessitent une étape pour obtenir la valeur de l'expression régulière ; la valeur extraite ne peut donc être appliquée qu'à l'étape suivante. Si la partie valeur commence par `regex:`, la partie qui suit est traitée comme une expression régulière qui recherche une correspondance dans la page web et, si elle en trouve une, l'enregistre dans la variable. Au moins un sous-groupe doit être présent afin que la valeur correspondante puisse être extraite. Les macros utilisateur et les macros `{HOST.*}` sont prises en charge. Les variables sont automatiquement encodées au format URL lorsqu'elles sont utilisées dans des champs de requête ou dans les données de formulaire des variables post, mais doivent être encodées manuellement au format URL lorsqu'elles sont utilisées dans des données post brutes ou directement dans l'URL.

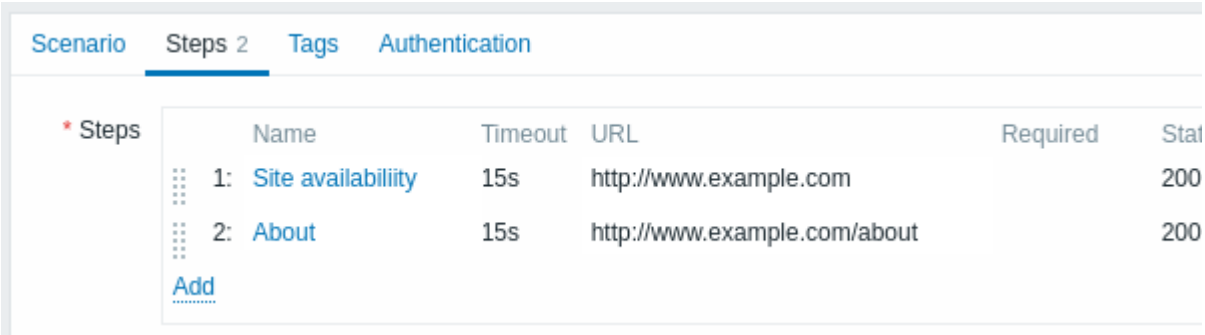
Paramètre	Description
En-têtes	Les en-têtes HTTP sont utilisés lors de l'exécution d'une requête. Des en-têtes par défaut et personnalisés peuvent être utilisés. Les en-têtes seront attribués à l'aide des paramètres par défaut correspondant au type d'agent sélectionné dans une liste déroulante au niveau du scénario et seront appliqués à toutes les étapes, sauf s'ils sont définis de manière personnalisée au niveau d'une étape. **Notez que la définition de l'en-tête au niveau d'une étape supprime automatiquement tous les en-têtes définis précédemment, à l'exception d'un en-tête par défaut attribué en sélectionnant « User-Agent » dans une liste déroulante au niveau du scénario.** Cependant, même l'en-tête par défaut « User-Agent » peut être remplacé en le spécifiant au niveau d'une étape. Pour désactiver l'en-tête au niveau du scénario, celui-ci doit être nommé et défini sans valeur au niveau d'une étape. Les en-têtes doivent être répertoriés selon la même syntaxe que celle utilisée dans le protocole HTTP, avec la possibilité d'utiliser certaines fonctionnalités supplémentaires prises en charge par l'option cURL CURLOPT_HTTPHEADER. Par exemple : Accept-Charset=utf-8 Accept-Language=en-US Content-Type=application/xml; charset=utf-8 Les macros utilisateur et les macros `{HOST.*}` sont prises en charge.
Activé	Le scénario est actif si cette case est cochée ; sinon, il est désactivé.

Notez que lors de la modification d'un scénario existant, deux boutons supplémentaires sont disponibles dans le formulaire :

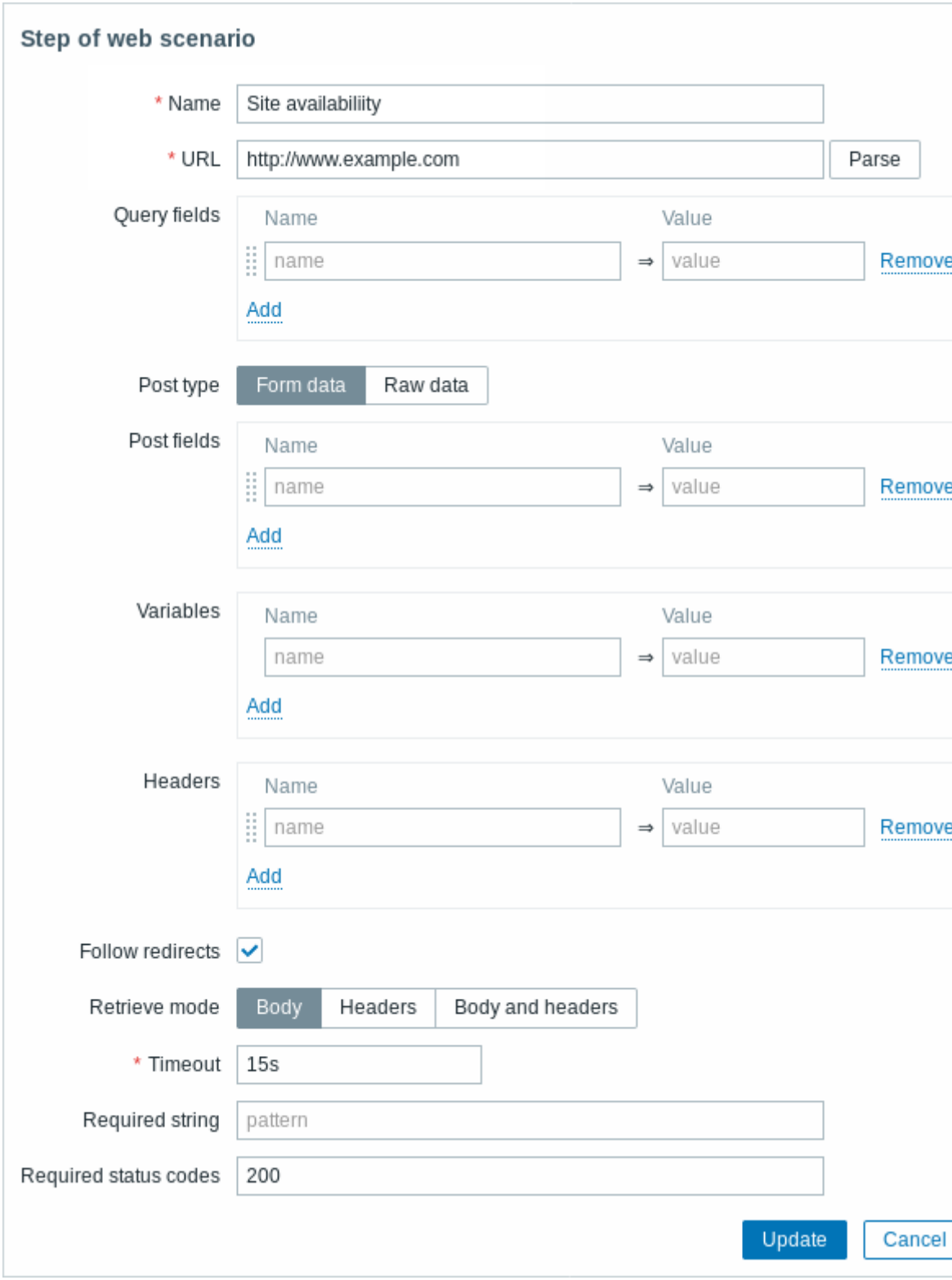
Clone	Créer un autre scénario à partir des propriétés du scénario existant.
Clear history and trends	Supprimer l'historique et les données de tendance du scénario. Le serveur exécutera le scénario immédiatement après la suppression des données.

Note:
Si le champ *Proxy HTTP* est laissé vide, une autre méthode pour utiliser un proxy HTTP consiste à définir les variables d'environnement liées au proxy.
Pour les vérifications HTTP, définissez la variable d'environnement `http_proxy` pour l'utilisateur du serveur Zabbix. Par exemple : `http_proxy=http://proxy_ip:proxy_port`.
Pour les vérifications HTTPS, définissez la variable d'environnement `HTTPS_PROXY`. Par exemple : `HTTPS_PROXY=http://proxy_ip:proxy_port`. Pour plus de détails, exécutez la commande shell suivante : `# man curl`.

L'onglet **Étapes** permet de configurer les étapes du scénario web. Pour ajouter une étape à un scénario web, cliquez sur *Ajouter* dans le bloc *Étapes*.



Note:
Les macros utilisateur secrètes ne doivent pas être utilisées dans les URL, car elles seront remplacées par "******".



Configuration des étapes

Paramètres de l'étape :

Paramètre	Description
Nom	Nom unique de l'étape. Les macros utilisateur sont prises en charge. *Notez* que si des macros utilisateur sont utilisées, ces macros resteront non résolues dans les noms des éléments de supervision Web.

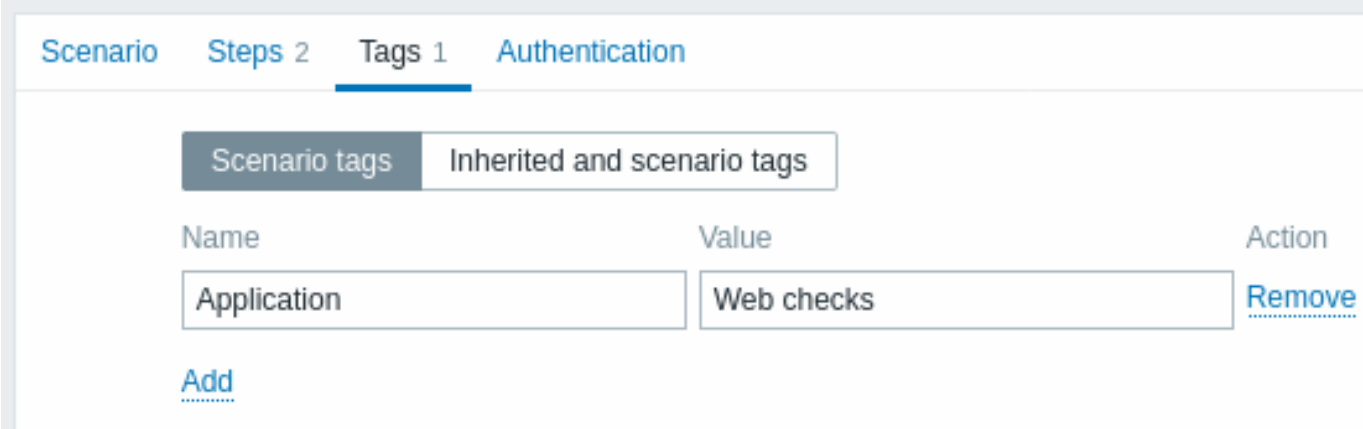
Paramètre	Description
URL	URL à laquelle se connecter et depuis laquelle récupérer les données. Par exemple : https://www.example.com http://www.example.com/download Les noms de domaine peuvent être spécifiés avec des caractères Unicode. Ils sont automatiquement convertis en punycode ASCII lors de l'exécution de l'étape du scénario Web. Le bouton *Analyser* peut être utilisé pour séparer de l'URL les champs de requête facultatifs (tels que `?key=value&limit=10`) et déplacer les attributs et les valeurs dans *Champs de requête* afin d'effectuer automatiquement l'encodage URL. Des variables peuvent être utilisées dans l'URL, avec la syntaxe {macro}. Les variables peuvent être encodées manuellement dans l'URL avec la syntaxe {{macro}.urlencode()}. Les macros utilisateur et les macros `{HOST.*}` sont prises en charge. Pour des raisons de sécurité, n'incluez pas de noms d'utilisateur ou de mots de passe réels dans les URL. Utilisez plutôt des macros utilisateur afin d'éviter d'exposer les identifiants. Limité à 2048 caractères.
Champs de requête	Variables HTTP GET pour l'URL. Spécifiées sous forme de paires attribut-valeur. Les valeurs sont automatiquement encodées dans l'URL. Les valeurs provenant de variables de scénario, de macros utilisateur ou de macros `{HOST.*}` sont résolues, puis automatiquement encodées dans l'URL. L'utilisation de la syntaxe {{macro}.urlencode()} entraîne un double encodage URL. Les macros utilisateur et les macros `{HOST.*}` sont prises en charge.
Post	Variables HTTP POST. En mode **Données de formulaire**, elles sont spécifiées sous forme de paires attribut-valeur. Les valeurs sont automatiquement encodées dans l'URL. Les valeurs provenant de variables de scénario, de macros utilisateur ou de macros `{HOST.*}` sont résolues, puis automatiquement encodées dans l'URL. En mode **Données brutes**, les attributs et les valeurs sont affichés sur une seule ligne et concaténés avec le symbole **&**. Les valeurs brutes peuvent être encodées ou décodées manuellement dans l'URL avec la syntaxe {{macro}.urlencode()} ou {{macro}.urldecode()}. Par exemple : id=2345&userid={user} Si {user} est défini comme variable du scénario Web, il sera remplacé par sa valeur lors de l'exécution de l'étape. Si vous souhaitez encoder la variable dans l'URL, remplacez {user} par {{user}.urlencode()}. Les macros utilisateur et les macros `{HOST.*}` sont prises en charge.
Variables	Variables au niveau de l'étape pouvant être utilisées pour les fonctions GET et POST. Spécifiées sous forme de paires attribut-valeur. Les variables au niveau de l'étape remplacent les variables au niveau du scénario ou les variables de l'étape précédente. Toutefois, la valeur d'une variable au niveau de l'étape n'affecte que l'étape suivante, et non l'étape actuelle. Elles utilisent le format suivant : **{macro}**=value **{macro}**=regex:<regular expression> Pour plus d'informations, consultez la description des variables au niveau du scénario. Les variables sont automatiquement encodées dans l'URL lorsqu'elles sont utilisées dans les champs de requête ou les données de formulaire des variables POST, mais doivent être encodées manuellement dans l'URL lorsqu'elles sont utilisées dans des données POST brutes ou directement dans l'URL.
En-têtes	En-têtes HTTP personnalisés qui seront envoyés lors de l'exécution d'une requête. Spécifiés sous forme de paires attribut-valeur. Un en-tête défini au niveau de l'étape sera utilisé pour cette étape particulière. **Notez que la définition de l'en-tête au niveau de l'étape supprime automatiquement tous les en-têtes définis précédemment, à l'exception d'un en-tête par défaut attribué en sélectionnant « User-Agent » dans une liste déroulante au niveau du scénario.** Toutefois, même l'en-tête par défaut « User-Agent » peut être remplacé en le spécifiant au niveau de l'étape. Par exemple, attribuer un nom à un en-tête sans définir de valeur supprimera l'en-tête par défaut défini au niveau du scénario. Les macros utilisateur et les macros `{HOST.*}` sont prises en charge. Cette option définit l'option cURL `CURLOPT_HTTPHEADER`.
Suivre les redirections	Cochez la case pour suivre les redirections HTTP. Cette option définit l'option cURL CURLOPT_FOLLOWLOCATION.

Paramètre	Description
Mode de récupération	Sélectionnez le mode de récupération : **Corps** - récupérer uniquement le corps de la réponse HTTP **En-têtes** - récupérer uniquement les en-têtes de la réponse HTTP **Corps et en-têtes** - récupérer le corps et les en-têtes de la réponse HTTP
Délai d'expiration	Zabbix ne consacrera pas plus que la durée définie au traitement de l'URL (d'une seconde à 1 heure maximum). En réalité, ce paramètre définit la durée maximale pour établir la connexion à l'URL et la durée maximale pour effectuer une requête HTTP. Par conséquent, Zabbix ne consacrera pas plus de **2 x Délai d'expiration** secondes à l'étape. Les suffixes de durée sont pris en charge, par exemple 30s, 1m, 1h. Les macros utilisateur sont prises en charge.
Chaîne requise	Modèle d'expression régulière requis. L'étape échouera si le contenu récupéré (HTML) ne correspond pas au modèle requis. Si le champ est vide, aucune vérification de la chaîne requise n'est effectuée. Par exemple : Homepage of Zabbix Welcome.*admin *Note* : il n'est pas possible de référencer dans ce champ les expressions régulières créées dans l'interface Zabbix. Les macros utilisateur et les macros `{HOST.*}` sont prises en charge.
Codes d'état requis	Liste des codes d'état HTTP attendus. Si Zabbix reçoit un code qui ne figure pas dans la liste, l'étape échouera. Si le champ est vide, aucune vérification des codes d'état n'est effectuée. Par exemple : 200,201,210-299 Les macros utilisateur sont prises en charge.

Note:
Les modifications apportées aux étapes d'un scénario Web ne seront enregistrées que lorsque l'ensemble du scénario sera enregistré.

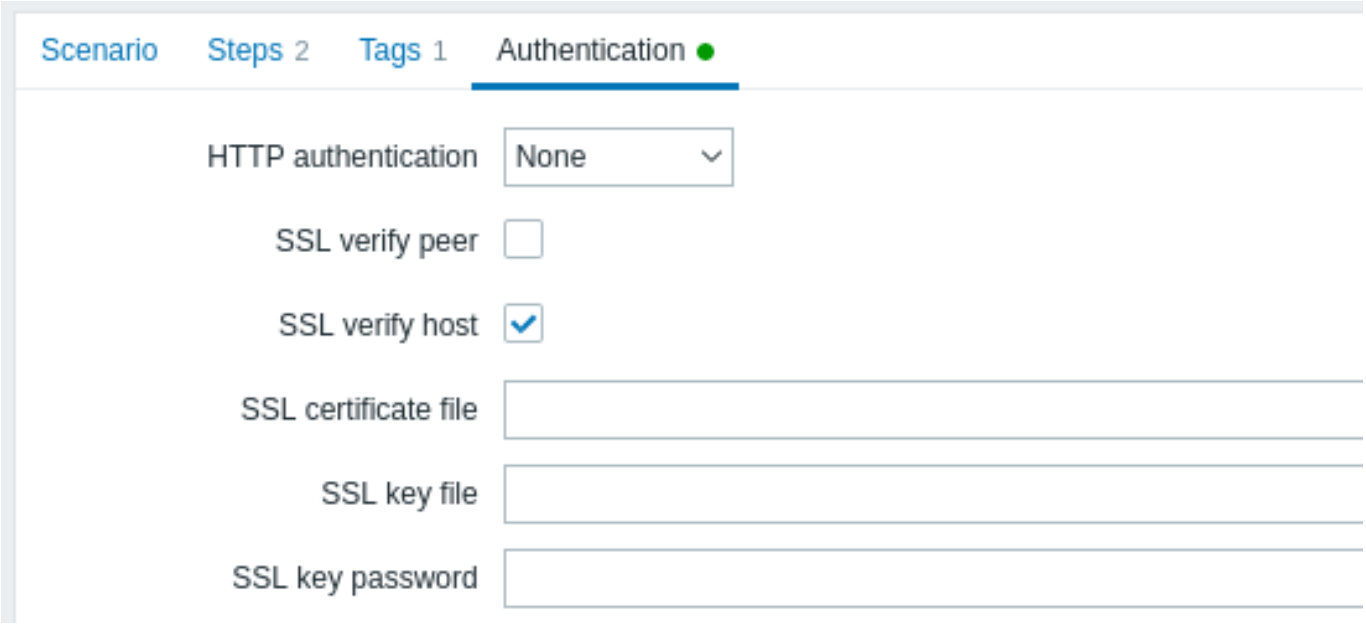
Consultez également un exemple concret montrant comment configurer les étapes de supervision Web.

Configuration des tags L'onglet **Tags** vous permet de définir des tags au niveau du scénario.



Les tags permettent de filtrer les scénarios web et les éléments de supervision web.

Configuration de l'authentification L'onglet **Authentification** permet de configurer les options d'authentification du scénario. Un point vert à côté du nom de l'onglet indique qu'un type d'authentification HTTP est activé.



Paramètres d'authentification :

Paramètre	Description
Authentification HTTP	Sélectionnez l'option d'authentification : **Aucune** - aucune authentification n'est utilisée ; **Basic** - l'authentification Basic est utilisée ; **NTLM** - l'authentification NTLM (Windows NT LAN Manager) est utilisée ; **Kerberos** - l'authentification Kerberos est utilisée (voir également : Configuration de Kerberos avec Zabbix) ; **Digest** - l'authentification Digest est utilisée.
Utilisateur	Saisissez le nom d'utilisateur (255 caractères maximum). Ce champ est disponible si *Authentification HTTP* est définie sur Basic, NTLM, Kerberos ou Digest. Les macros utilisateur sont prises en charge.
Mot de passe	Saisissez le mot de passe de l'utilisateur (255 caractères maximum). Ce champ est disponible si *Authentification HTTP* est définie sur Basic, NTLM, Kerberos ou Digest. Les macros utilisateur sont prises en charge.
Vérifier le pair SSL	Cochez la case pour vérifier le certificat SSL du serveur web. Le certificat du serveur est vérifié par rapport aux certificats de l'autorité de certification (CA) de confiance situés dans l'emplacement système des CA. Vous pouvez remplacer l'emplacement des fichiers CA à l'aide du paramètre de configuration du serveur Zabbix ou du proxy SSLCALocation. Cette option définit l'option cURL CURLOPT_SSL_VERIFYPEER.
Vérifier l'hôte SSL	Cochez la case pour vérifier que le champ *Common Name* ou le champ *Subject Alternate Name* du certificat du serveur web correspond. Cette option définit l'option cURL CURLOPT_SSL_VERIFYHOST.
Fichier de certificat SSL	Nom du fichier de certificat SSL utilisé pour l'authentification du client. Le fichier de certificat doit être au format PEM[1]. Si le fichier de certificat contient également la clé privée, laissez le champ *Fichier de clé SSL* vide. Si la clé est chiffrée, indiquez le mot de passe dans le champ *Mot de passe de la clé SSL*. Le répertoire contenant ce fichier est spécifié par le paramètre de configuration du serveur Zabbix ou du proxy SSLCertLocation. Les macros `HOST.*` et les macros utilisateur peuvent être utilisées dans ce champ. Cette option définit l'option cURL CURLOPT_SSLCERT.
Fichier de clé SSL	Nom du fichier de clé privée SSL utilisé pour l'authentification du client. Le fichier de clé privée doit être au format PEM[1]. Le répertoire contenant ce fichier est spécifié par le paramètre de configuration du serveur Zabbix ou du proxy SSLKeyLocation. Les macros `HOST.*` et les macros utilisateur peuvent être utilisées dans ce champ. Cette option définit l'option cURL CURLOPT_SSLKEY.
Mot de passe de la clé SSL	Mot de passe du fichier de clé privée SSL. Les macros utilisateur peuvent être utilisées dans ce champ. Cette option définit l'option cURL CURLOPT_KEYPASSWD.

Attention:
[1] Zabbix prend uniquement en charge les fichiers de certificat et de clé privée au format PEM. Si les données de votre certificat et de votre clé privée se trouvent dans un fichier au format PKCS #12 (généralement avec l'extension *.p12 ou *.pfx), vous pouvez générer le fichier PEM à l'aide des commandes suivantes :

```
openssl pkcs12 -in ssl-cert.p12 -clcerts -nokeys -out ssl-cert.pem
openssl pkcs12 -in ssl-cert.p12 -nocerts -nodes  -out ssl-cert.key
```

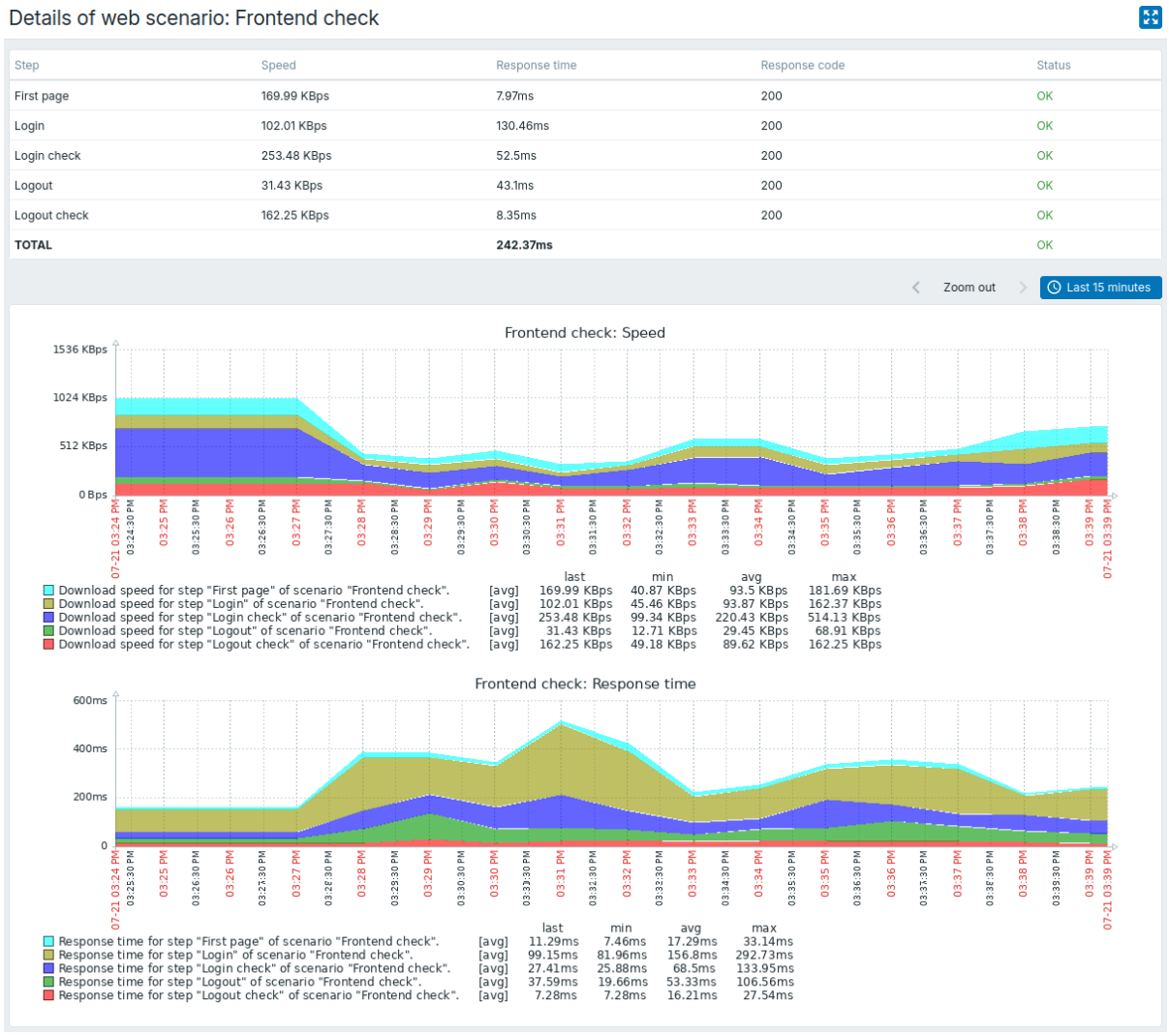
Note:
Le serveur Zabbix prend en compte les modifications apportées aux certificats sans redémarrage.

Note:
Si le certificat client et la clé privée se trouvent dans un seul fichier, indiquez simplement ce fichier dans le champ « Fichier de certificat SSL » et laissez le champ « Fichier de clé SSL » vide. Le certificat et la clé doivent toujours être au format PEM. Il est facile de combiner le certificat et la clé :

```
cat client.crt client.key > client.pem
```

Affichage Pour afficher les scénarios web configurés pour un hôte :

1. Accédez à *Monitoring* > *Hosts*.
2. Repérez l'hôte dans la liste et cliquez sur le lien hypertexte *Web* dans la dernière colonne.
3. Cliquez sur le nom du scénario pour obtenir des informations détaillées.



Une vue d'ensemble des scénarios web peut également être affichée dans *Dashboards* à l'aide du widget de surveillance web.

Les résultats récents de l'exécution du scénario web sont disponibles dans la section *Monitoring* > *Latest data*.

Surveillance étendue Il est parfois nécessaire d'enregistrer le contenu HTML de la page reçue.
Cela est particulièrement utile si une étape de scénario web échoue.

Le niveau de débogage 5 (trace) sert à cet effet.
Ce niveau peut être défini dans les fichiers de configuration serveur et proxy ou à l'aide d'une option de contrôle à l'exécution (`-R log_level_increase="http poller,N"`, où N est le numéro du processus).
Les exemples suivants montrent comment la surveillance étendue peut être démarrée, à condition que le niveau de débogage 4 soit déjà défini :

```
# Augmenter le niveau de journalisation de tous les http pollers :
zabbix_server -R log_level_increase="http poller"

# Augmenter le niveau de journalisation du deuxième http poller :
zabbix_server -R log_level_increase="http poller,2"
```

Si la surveillance web étendue n'est pas requise, elle peut être arrêtée à l'aide de l'option `-R log_level_decrease`.

Éléments de surveillance web

Aperçu

Certains nouveaux éléments sont automatiquement ajoutés pour la supervision lors de la création de scénarios Web.

Tous les éléments héritent des tags du scénario Web.

Éléments du scénario

Dès qu'un scénario est créé, Zabbix ajoute automatiquement les éléments suivants pour la surveillance.

Élément	Description
Vitesse de téléchargement pour le scénario <Scenario>	Cet élément collectera des informations sur la vitesse de téléchargement (octets par seconde) de l'ensemble du scénario, c'est-à-dire la moyenne de toutes les étapes. Clé d'élément : web.test.in[Scenario,,bps] Type : *Numeric(float)*
Étape échouée du scénario <Scenario>	Cet élément affichera le numéro de l'étape qui a échoué dans le scénario. Si toutes les étapes sont exécutées avec succès, 0 est renvoyé. Clé d'élément : web.test.fail[Scenario] Type : *Numeric(unsigned)*
Dernier message d'erreur du scénario <Scenario>	Cet élément renvoie le texte du dernier message d'erreur du scénario. Une nouvelle valeur n'est stockée que si le scénario comporte une étape en échec. Si toutes les étapes sont correctes, aucune nouvelle valeur n'est collectée. Clé d'élément : web.test.error[Scenario] Type : *Character*

Le nom réel du scénario sera utilisé à la place de « Scenario ».

Note:
Si le nom du scénario contient des macros utilisateur, ces macros resteront non résolues dans les noms des éléments de supervision web.

 Si le nom du scénario commence par un guillemet double ou contient une virgule ou un crochet, il sera correctement mis entre guillemets dans les clés d'élément. Dans les autres cas, aucune mise entre guillemets supplémentaire ne sera effectuée.

Note:
Les éléments de supervision web sont ajoutés avec un historique de 30 jours et une période de conservation des tendances de 90 jours.

Ces éléments peuvent être utilisés pour créer des déclencheurs et définir des conditions de notification.

Exemple 1

Pour créer un déclencheur "Échec du scenario Web", vous pouvez définir une expression de déclencheur :

```
last(/host/web.test.fail[Scenario])<>0
```

Assurez-vous de remplacer 'Scenario' par le vrai nom de votre scénario.

Exemple 2

Pour créer un déclencheur ”Échec du scenario Web” avec une description de problème utile dans le nom du déclencheur, vous pouvez définir un déclencheur avec le nom suivant :

```
Echec du scenario Web "Scenario": {ITEM.VALUE}
```

Et l’expression de déclencheur :

```
length(last(/host/web.test.error[Scenario]))>0 and last(/host/web.test.fail[Scenario])>0
```

Assurez-vous de remplacer ’Scenario’ par le vrai nom de votre scénario.

Exemple 3

Pour créer un déclencheur ”Application Web lente”, vous pouvez définir une expression de déclencheur :

```
last(/host/web.test.in[Scenario,,bps])<10000
```

Assurez-vous de remplacer ’Scenario’ par le vrai nom de votre scénario.

Éléments d’étape de scénario

Dès qu’une étape est créée, Zabbix ajoute automatiquement les éléments suivants à superviser.

Élément	Description
Vitesse de téléchargement de l’étape <Etape> du scénario <Scenario>	Cet élément collectera des informations sur la vitesse de téléchargement (octets par seconde) de l’étape. Clé de l’élément :web.test.in[Scenario,Step,bps] Type : *Numeric(float)*
Temps de réponse de l’étape <Etape> du scénario <Scenario>	et élément collectera des informations sur le temps de réponse de l’étape en secondes. Le temps de réponse est compté à partir du début de la demande jusqu’à ce que toutes les informations aient été transférées. Clé de l’élément : web.test.time[Scenario,Step,resp] Type : *Numeric(float)*
* Code de réponse pour l’étape <Etape> du scénario <Scenario>*	Cet élément va collecter les codes de réponse de l’étape. Clé de l’élément : web.test.rspcode[Scenario,Step] Type : *Numeric(unsigned)*

Les noms de scénario et d’étape seront utilisés à la place de ”Scenario” et ”Step” respectivement.

> **Note:**
> Les éléments de supervision Web sont ajoutés avec un historique de 30 jour et une période de conservation des tendances de 90 jours

> **Note:**
> Si le nom du scénario commence par des guillemets doubles ou contient une virgule ou un crochet, il sera correctement déspécialisé dans les clés d’élément. Dans les autres cas, aucune déspécialisation supplémentaire ne sera effectuée.

Ces éléments peuvent être utilisés pour créer des déclencheurs et définir des conditions de notification. Par exemple, pour créer un déclencheur ”La connexion à l’interface graphique de Zabbix est trop lente”, vous pouvez définir une expression de déclencheur :

```
last(/zabbix/web.test.time[ZABBIX GUI,Login,resp])>3
```

Scénario réel

Vue d’ensemble

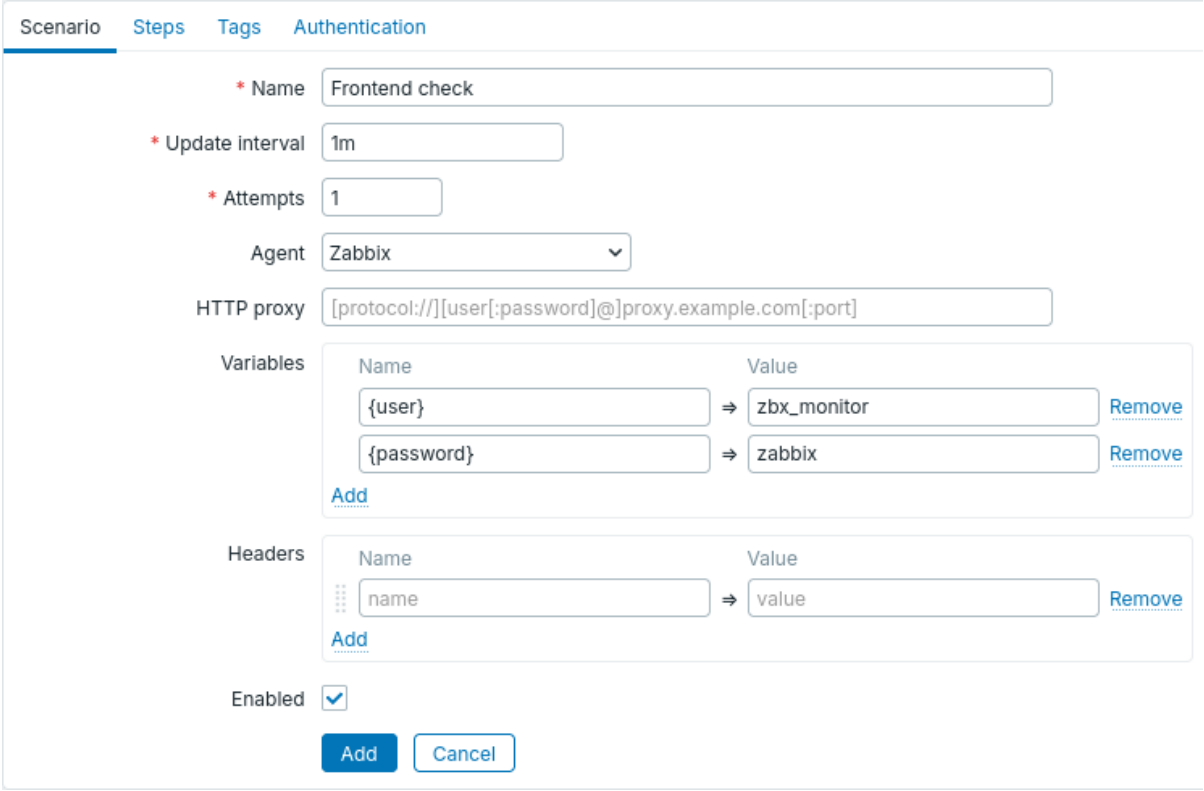
Cette section présente un exemple concret, étape par étape, de la manière dont la surveillance web peut être utilisée.

La surveillance web de Zabbix sera utilisée pour surveiller l’interface Zabbix. L’objectif est de déterminer si elle est disponible, si elle fournit le bon contenu et à quelle vitesse elle fonctionne. Pour y parvenir, plusieurs étapes sont nécessaires, notamment la vérification de la disponibilité de la première page, la connexion avec un nom d’utilisateur et un mot de passe, la vérification de la réussite de la connexion, la déconnexion et la confirmation de la déconnexion.

Scénario

Ajouter un nouveau scénario web

1. Accédez à *Collecte de données* > *Hôtes*.
2. Choisissez un hôte et cliquez sur *Web* dans la ligne de cet hôte.
3. Cliquez sur *Créer un scénario web*.



4. Dans le formulaire du nouveau scénario, dans l'onglet *Scénario*, renseignez les champs suivants:

    - *Nom*: Frontend check.
    - *Intervalle de mise à jour*: 1m.
    - *Tentatives*: 1.
    - *Agent*: Zabbix.
 Dans la section *Variables*, ajoutez les paires nom-valeur suivantes:
    - `{user}`: vos identifiants utilisateur Zabbix existants;
    - `{password}`: vos identifiants utilisateur Zabbix existants.

Note:
Pour des raisons de sécurité, il est recommandé de créer un utilisateur distinct avec des permissions minimales à utiliser à des fins de supervision.

En option, passez à l'onglet *Tags* et ajoutez des tags de scénario web. Vous pouvez utiliser les tags de scénario web pour identifier rapidement les éléments et déclencheurs associés ou rechercher dans les données collectées. Par exemple, des tags appropriés pour ce tutoriel sont `component: web-scenario` et/ou `target: frontend`.

Configurer les étapes du scénario web

Passez à l'onglet *Steps* et définissez les étapes du scénario.
 Après avoir configuré chaque étape, cliquez sur le bouton *Add*.

Attention:
Notez que l'interface Zabbix utilise une redirection JavaScript lors de la connexion, donc la connexion doit avoir lieu en premier, et les fonctionnalités accessibles après authentification ne peuvent être vérifiées que dans les étapes suivantes. De plus, l'étape de connexion doit utiliser l'URL complète du fichier **index.php**.

Champs communs à toutes les étapes

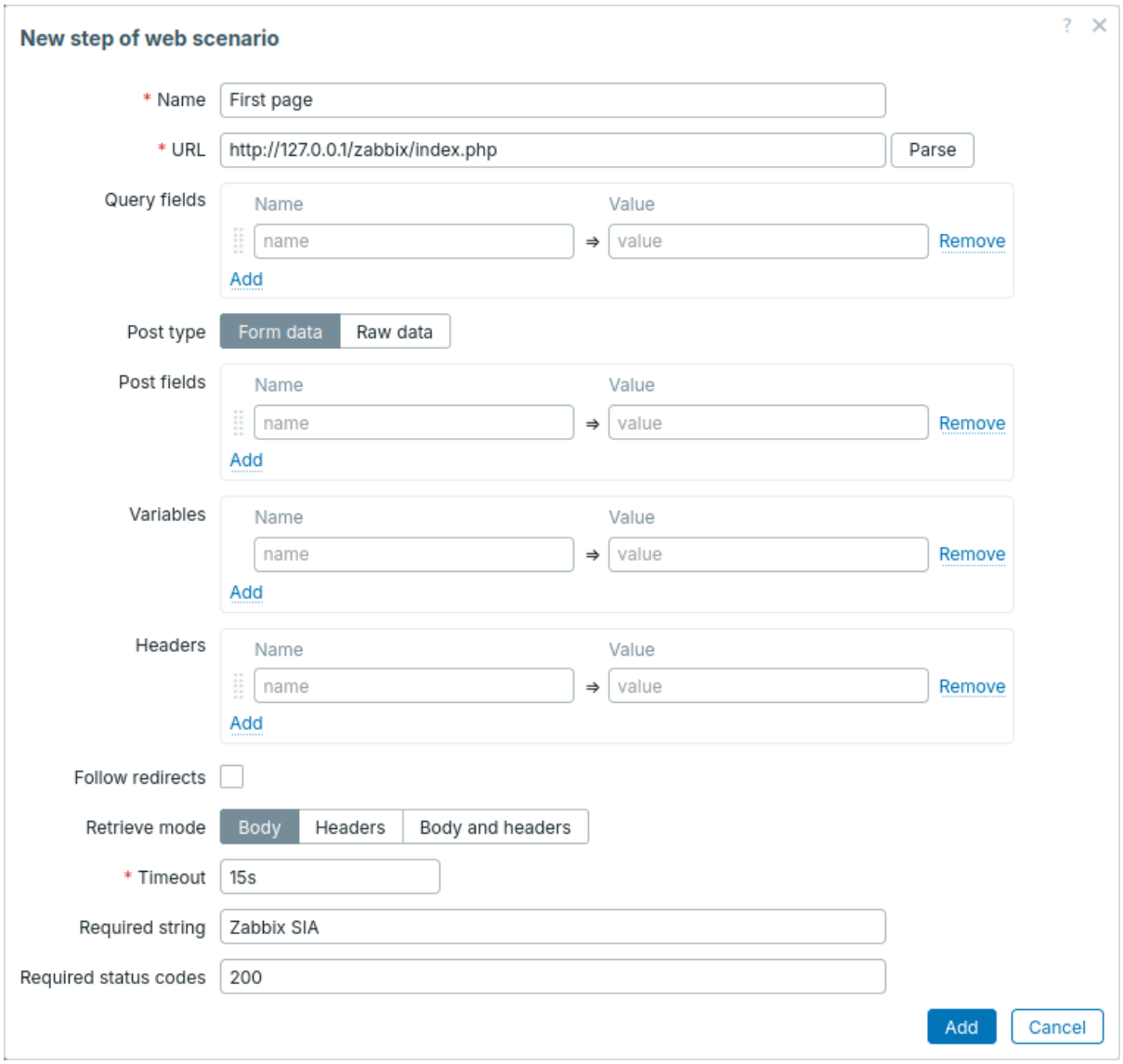
En plus des champs spécifiques à chaque étape, renseignez les champs communs suivants pour chaque étape décrite ci-dessous :

- *URL* : l'URL de l'interface Zabbix (`http://<server_ip_or_name>/zabbix/index.php` (pour Apache) ou `http://<server_ip_or` (pour NGINX)). Pour l'étape 3 du scénario Web, utilisez une URL différente, comme décrit ci-dessous.
- *Timeout* : 15s.
- *Required status codes* : 200.

Étape 1 du scénario web

Vérifiez que la première page répond correctement, renvoie le code d'état HTTP 200 et contient le texte *Zabbix SIA*.
 Renseignez les champs suivants :

- *Nom* : Première page.
- *Délai d'attente* : 15s.
- *Chaîne requise* : Zabbix SIA.
- Les champs communs pour toutes les étapes.

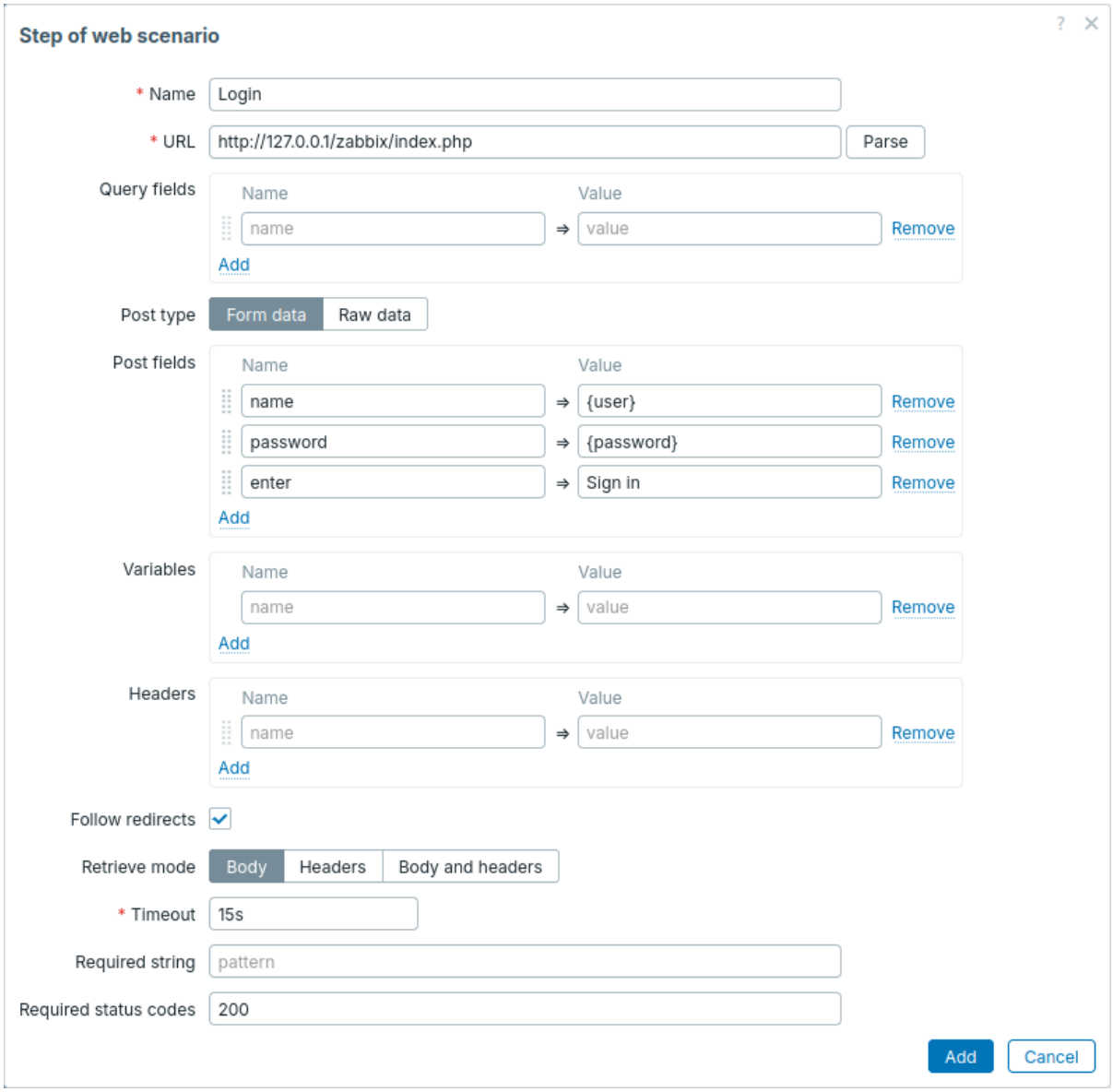


Étape 2 du scénario Web

Connectez-vous à l'interface Zabbix à l'aide des macros (variables) définies dans l'onglet *Scenario* : *{user}* et *{password}*.
 Renseignez les champs suivants :

- *Name* : Login.
- Dans la section *Post fields*, ajoutez les paires nom-valeur suivantes :
  - name : `{user}`;
  - password : `{password}`;
  - enter : Sign in.
- Les champs communs à toutes les étapes.

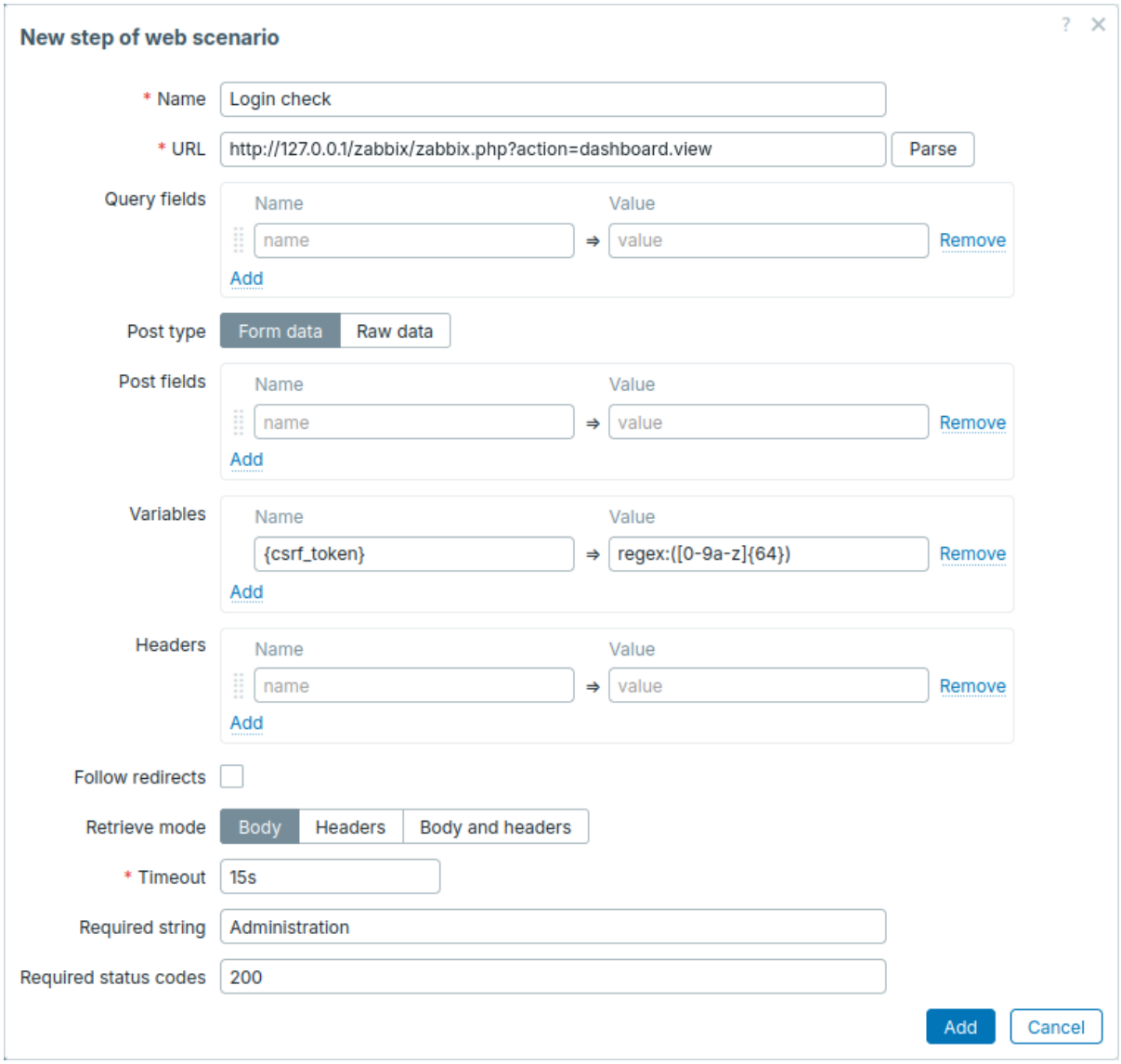
Cochez la case *Follow redirects*.



Étape 3 du scénario web

Après la connexion, vérifiez la réussite en recherchant une chaîne visible uniquement lorsque vous êtes connecté, par exemple, *Administration*.
 Renseignez les champs suivants :

- *Nom* : Vérification de la connexion.
- *URL* : l'URL vers laquelle l'utilisateur est redirigé après la connexion (par exemple, http://127.0.0.1/zabbix/zabbix.php?action=dashboard
- Dans la section *Variables*, ajoutez la paire nom-valeur `{csrf_token}` : `regex:([0-9a-z]{64})`. Cette variable capturera la valeur du jeton CSRF attribué afin de la réutiliser dans l'étape 4 du scénario web.
- *Chaîne requise* : Administration.
- Les champs communs à toutes les étapes.



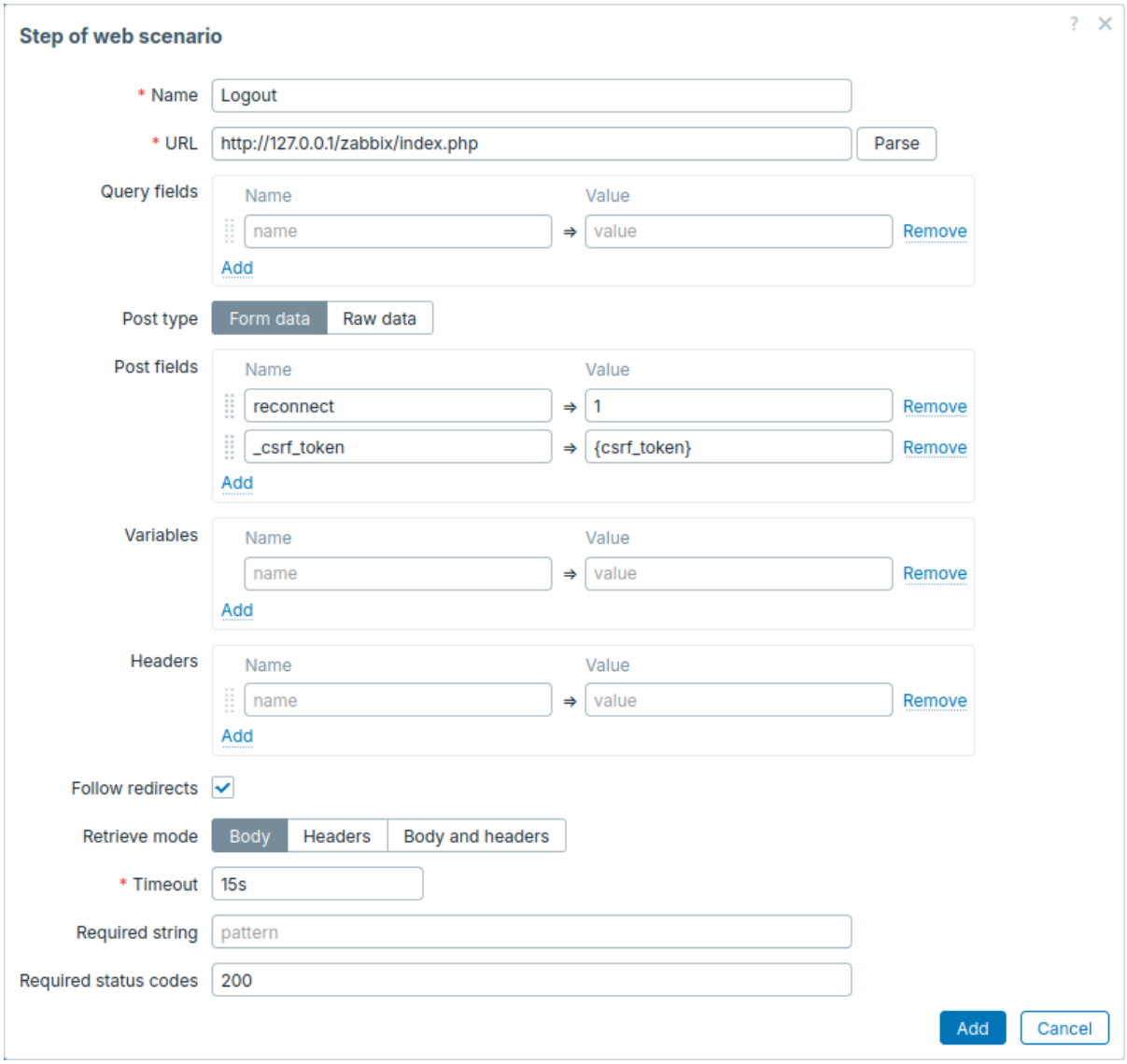
Étape 4 du scénario web

Une fois l'accessibilité de l'interface et la connexion vérifiées, ajoutez une étape de déconnexion ; sinon, la base de données Zabbix sera encombrée par de nombreux enregistrements de session ouverts.
 Renseignez les champs suivants :

- *Nom* : Logout.
- Dans la section *Champs POST*, ajoutez les paires nom-valeur suivantes :
  - reconnect : 1 ;
  - _csrf_token : `{csrf_token}`.
- Les champs communs à toutes les étapes.

Cochez la case *Suivre les redirections*.

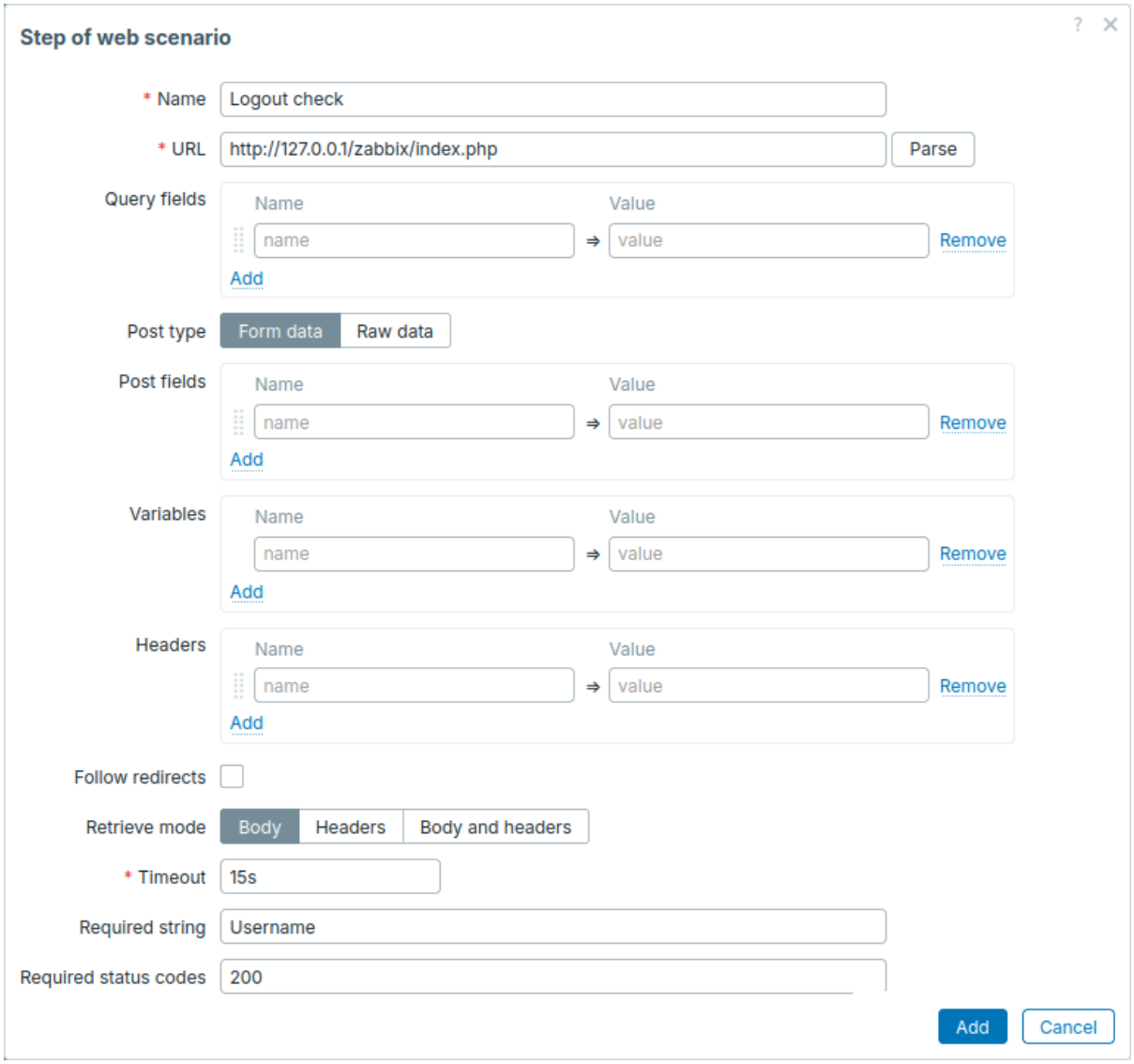
Cette étape utilise la variable `{csrf_token}` obtenue à l'étape 3 du scénario web.



Étape 5 du scénario web

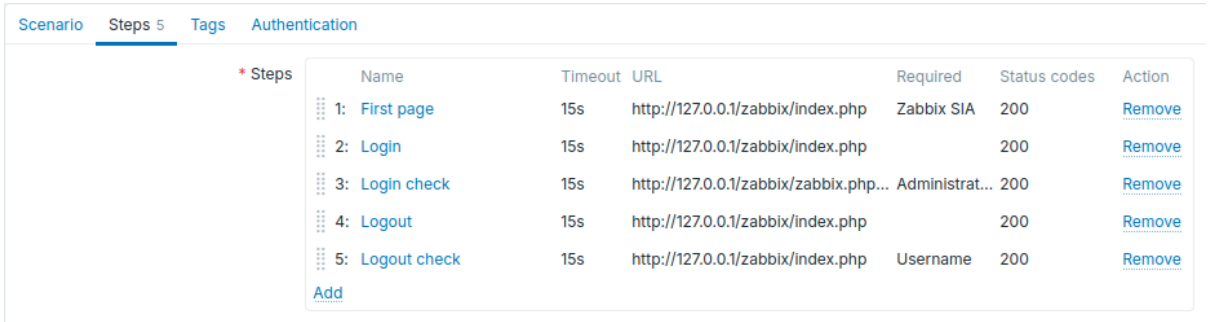
Pour confirmer la déconnexion, vérifiez la chaîne *Username*.
 Renseignez les champs suivants :

- *Nom* : Vérification de la déconnexion.
- *Chaîne requise* : Username.
- Les champs communs à toutes les étapes.



Configuration complète des étapes

Une configuration complète des étapes d'un scénario web devrait ressembler à ceci :

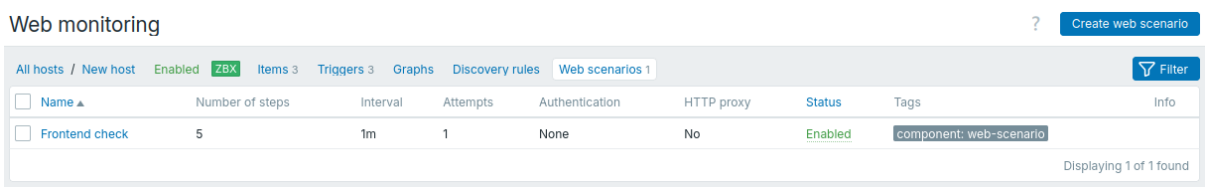


Vérifier les résultats

Enregistrez le scénario de surveillance web terminé.

Lorsque le scénario web est créé, de nouveaux éléments de surveillance web sont automatiquement ajoutés pour la surveillance.

Le scénario sera ajouté à l'hôte. Pour afficher les informations du scénario web, accédez à *Monitoring* > *Hosts*, localisez l'hôte dans la liste et cliquez sur le lien hypertexte *Web* dans la dernière colonne.



Cliquez sur le nom du scénario pour voir des statistiques détaillées :



Surveillance des services

Vue d'ensemble La supervision des services est une supervision au niveau métier qui permet d'obtenir une vue d'ensemble de l'arborescence complète des services de l'infrastructure informatique, d'identifier les points faibles de l'infrastructure, de calculer le SLA de différents services informatiques et de consulter d'autres informations à un niveau supérieur. La supervision des services se concentre sur la disponibilité globale d'un service plutôt que sur les détails de bas niveau, tels que le manque d'espace disque, une charge processeur élevée, etc. La supervision des services fournit également une fonctionnalité permettant de trouver la cause première d'un problème lorsqu'un service ne fonctionne pas comme prévu.

La supervision des services permet de créer une représentation hiérarchique des données supervisées.

Une structure de services très simple peut se présenter comme suit :

```
Service
|
|-Postes de travail
| |
| |-Poste1
| |
| |-Poste2
|
|-Serveurs
```

Chaque nœud de la structure possède un attribut d'état. L'état est calculé et propagé vers les niveaux supérieurs selon l'algorithme sélectionné. L'état des nœuds individuels est affecté par l'état des problèmes associés. L'association des problèmes est effectuée à l'aide de balises.

Zabbix peut envoyer des notifications ou exécuter automatiquement un script sur le serveur Zabbix lorsqu'un changement d'état d'un service est détecté. Il est possible de définir des règles flexibles pour déterminer si un service parent doit passer à l'« état Problème » en fonction de l'état de ses services enfants. Les données relatives aux problèmes des services peuvent ensuite être utilisées pour calculer le SLA et envoyer des rapports SLA selon un ensemble flexible de conditions.

La supervision des services est configurée dans le menu Services, qui comprend les sections suivantes :

- Services

La section Services vous permet de créer une hiérarchie de votre infrastructure supervisée en ajoutant des services parents, puis des services enfants à ces services parents.

En plus de permettre la configuration de l'arborescence des services, cette section fournit une vue d'ensemble de l'infrastructure complète et permet d'identifier rapidement les problèmes ayant entraîné un changement d'état d'un service.

- SLA

Dans cette section, vous pouvez définir des accords de niveau de service et définir des objectifs de niveau de service pour des services spécifiques.

- Rapport SLA

Dans cette section, vous pouvez consulter les rapports SLA.

Actions de service

Vous pouvez également configurer des actions de service.

Les actions de service sont facultatives et permettent de :

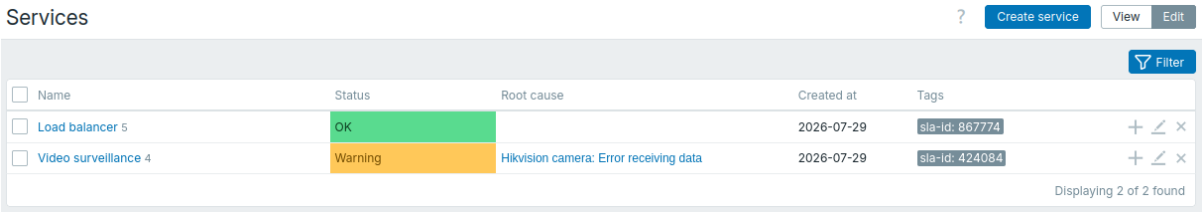
- envoyer une notification lorsqu'un service est indisponible
- exécuter une commande à distance sur le serveur Zabbix lors d'un changement d'état d'un service
- envoyer une notification de rétablissement lorsqu'un service est de nouveau disponible.

Voir aussi :

- Exemple de configuration de la supervision SLA
- Remarques concernant la mise à niveau des services depuis des versions de Zabbix antérieures à la version 6.0

Arborescence des services

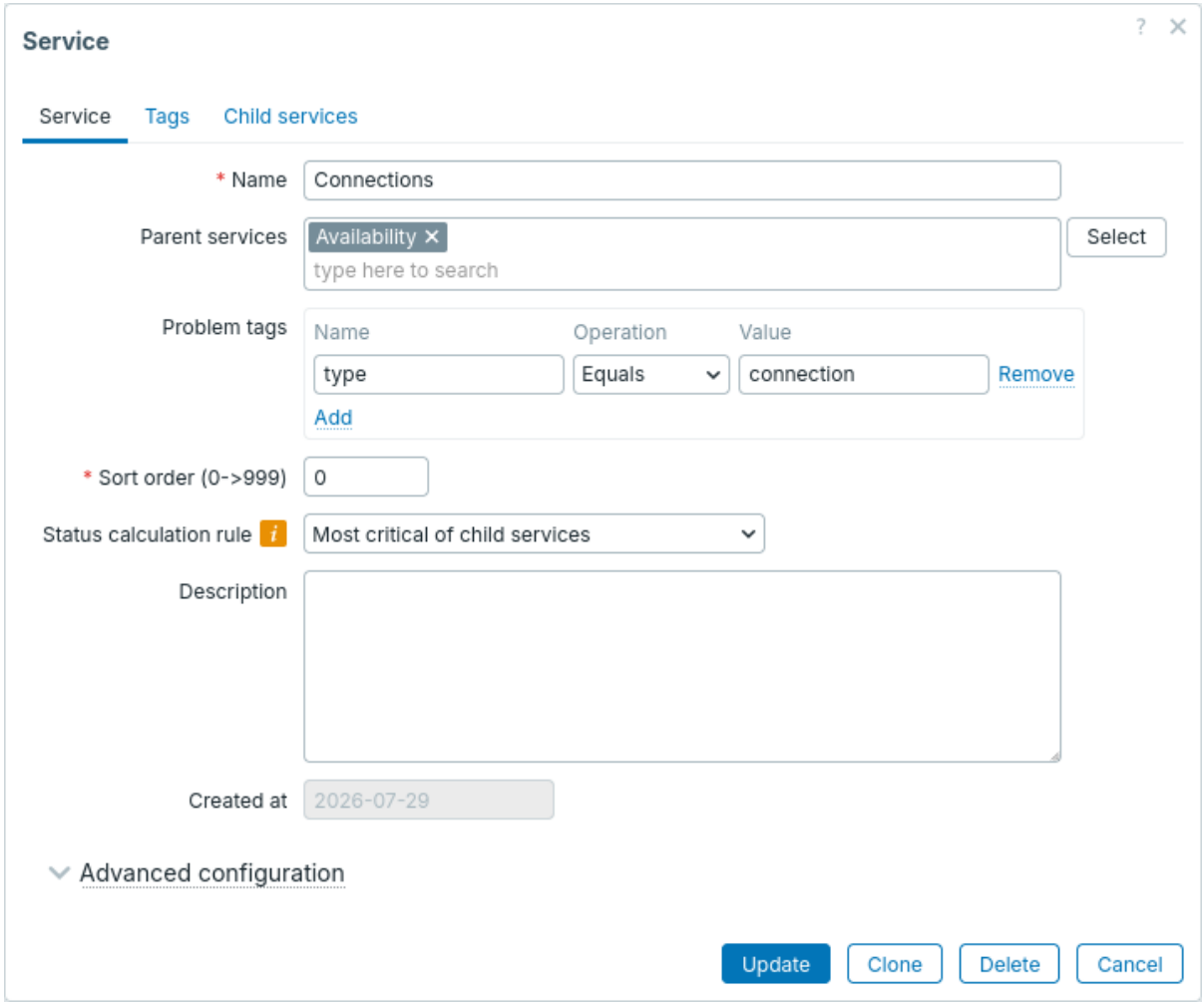
L'arborescence des services est configurée dans la section de menu *Services* > *Services*. Dans le coin supérieur droit, passez du mode view au mode édition.



Pour configurer un nouveau service, cliquez sur le bouton *Create service* dans le coin supérieur droit.

Pour ajouter rapidement un service enfant, vous pouvez également cliquer sur l'icône plus à côté du service parent. Cela ouvrira le même formulaire de configuration du service, mais le paramètre *Parent services* sera prérempli.

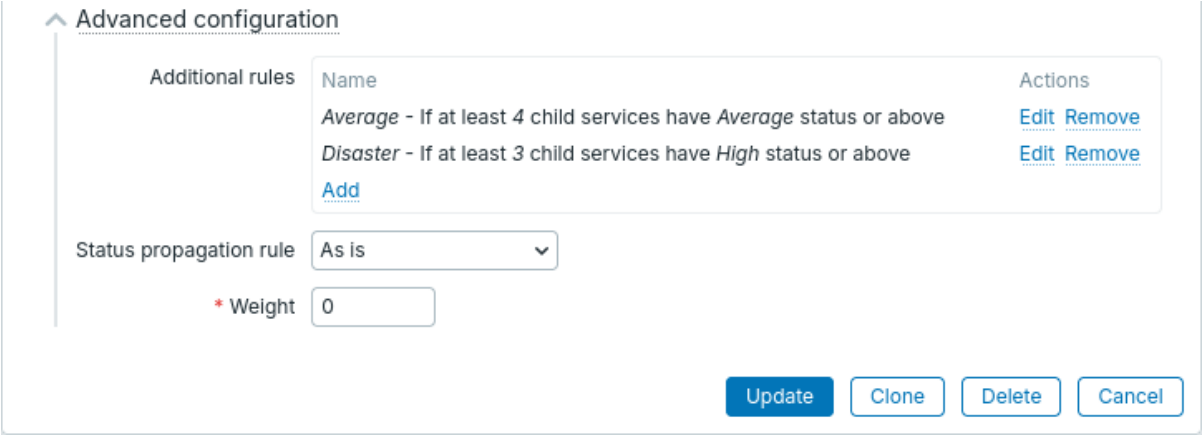
Configuration du service Dans l'onglet **Service**, spécifiez les paramètres de service requis :



Tous les champs de saisie obligatoires sont marqués d'un astérisque rouge.

Parameter	Description
Name	Nom du service.
Parent services	Services parents auxquels le service appartient. Laissez ce champ vide si vous ajoutez le service de plus haut niveau. Un service peut avoir plusieurs services parents. Dans ce cas, il sera affiché dans l'arborescence des services sous chacun des services parents.
Problem tags	Spécifiez des tags pour associer les données de problème au service. Plusieurs conditions peuvent être définies. La correspondance du nom du tag est toujours sensible à la casse. Deux opérateurs sont disponibles pour chaque condition : **Equals** - inclure les noms et valeurs de tag spécifiés (sensible à la casse) **Contains** - inclure les noms de tag spécifiés lorsque les valeurs de tag contiennent la chaîne saisie (correspondance de sous-chaîne, sensible à la casse)
Sort order	Ordre de tri pour l'affichage, le plus bas vient en premier.
Status calculation rule	Règle de calcul de l'état du service : **Most critical if all children have problems** - le problème le plus critique dans les services enfants est utilisé pour colorer l'état du service, si tous les services enfants ont des problèmes **Most critical of child services** - le problème le plus critique dans les services enfants est utilisé pour colorer l'état du service **Set status to OK** - ne pas calculer l'état du service D'autres règles de calcul de l'état peuvent être configurées dans les options de configuration avancée.
Description	Description du service.
Created at	Moment de création du service ; affiché lors de la modification d'un service existant.
Advanced configuration	Cliquez sur l'en-tête *Advanced configuration* pour afficher les options de configuration avancée.

Configuration avancée



Parameter	Description
Additional rules	Cliquez sur *Add* pour configurer des règles supplémentaires de calcul du statut.
Set status to	Définissez le statut du service sur *OK* (par défaut), *Not classified*, *Information*, *Warning*, *Average*, *High* ou *Disaster* en cas de correspondance avec une condition.
Condition	Sélectionnez la condition pour les services enfants directs: **si au moins (N) services enfants ont le statut (Status) ou supérieur** **si au moins (N%) des services enfants ont le statut (Status) ou supérieur** **si moins de (N) services enfants ont le statut (Status) ou inférieur** **si moins de (N%) des services enfants ont le statut (Status) ou inférieur** **si le poids des services enfants avec le statut (Status) ou supérieur est au moins (W)** **si le poids des services enfants avec le statut (Status) ou supérieur est au moins (N%)** **si le poids des services enfants avec le statut (Status) ou inférieur est inférieur à (W)** **si le poids des services enfants avec le statut (Status) ou inférieur est inférieur à (N%)** Si plusieurs conditions sont spécifiées et que la situation correspond à plus d'une condition, la gravité la plus élevée sera appliquée.
N (W)	Définissez la valeur de N ou W (1-100000), ou N% (1-100) dans la condition.
Status	Sélectionnez la valeur de *Status* dans la condition: *OK* (par défaut), *Not classified*, *Information*, *Warning*, *Average*, *High* ou *Disaster*.
Status propagation rule	Règle de propagation du statut du service vers le service parent: **As is** - le statut est propagé sans modification **Increase by** - vous pouvez augmenter le statut propagé de 1 à 5 niveaux de gravité **Decrease by** - vous pouvez diminuer le statut propagé de 1 à 5 niveaux de gravité **Ignore this service** - le statut n'est pas propagé au service parent du tout **Fixed status** - le statut est propagé de manière statique, c'est-à-dire toujours identique
Weight	Poids du service (entier compris entre 0 (par défaut) et 1000000).

Note:
Les règles supplémentaires de calcul du statut ne peuvent être utilisées que pour augmenter le niveau de gravité au-dessus du niveau calculé selon le paramètre principal *Status calculation rule*. Si, selon les règles supplémentaires, le statut doit être Warning, mais que, selon *Status calculation rule*, le statut est Disaster, le service aura le statut Disaster.

L'onglet *Tags* contient des tags au niveau du service. Les tags au niveau du service servent à identifier un service. Les tags de ce type ne sont pas utilisés pour associer des problèmes au service (pour cela, utilisez les *tags de problème* du premier onglet).

L'onglet *Services enfants* vous permet de spécifier les services dépendants. Cliquez sur *Ajouter* pour ajouter un service à partir de la liste des services existants. Si vous souhaitez ajouter un nouveau service enfant, enregistrez d'abord ce service, puis cliquez sur une icône plus à côté du service que vous venez de créer.

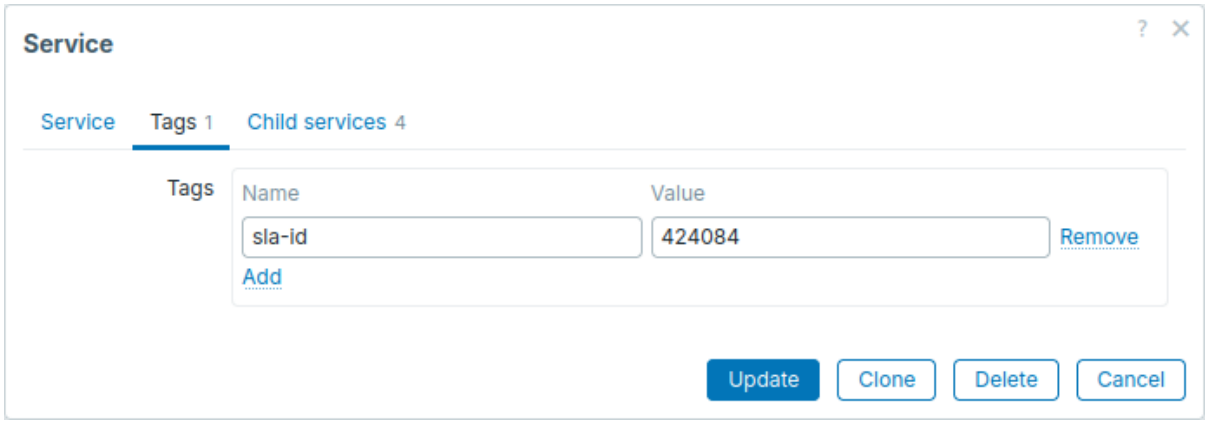
Balises Il existe deux types de balises dans les services :

- Balises de service
- Balises de problème

Balises de service

Les balises de service sont utilisées pour associer des services à des actions de service et à des SLA. Ces balises sont spécifiées dans l'onglet de configuration du service *Balises*. Pour le mappage des SLA, la logique *OR* est utilisée : un service sera associé à

un SLA s'il possède au moins une balise correspondante. Dans les actions de service, les règles de mappage sont configurables et peuvent utiliser la logique *AND*, *OR* ou *AND/OR*.

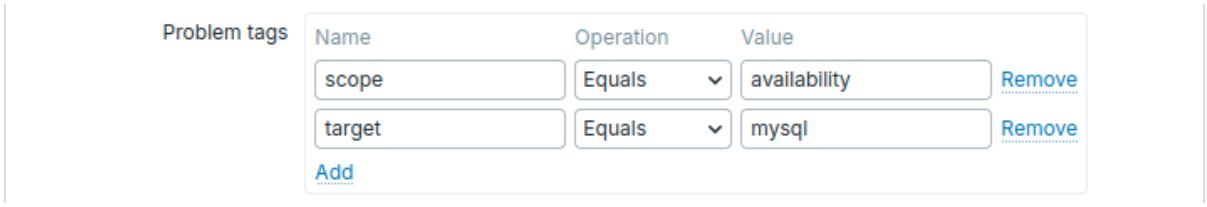


Balises de problème

Les balises de problème sont utilisées pour faire correspondre les problèmes et les services. Ces balises sont spécifiées dans l'onglet de configuration du service principal.

Seuls les services enfants du niveau hiérarchique le plus bas peuvent avoir des balises de problème définies et être directement corrélés aux problèmes. Si les balises de problème correspondent, l'état du service changera pour prendre le même état que celui du problème. En cas de plusieurs problèmes, un service aura l'état du problème le plus grave. L'état d'un service parent est alors calculé en fonction des états des services enfants, conformément aux règles de calcul de l'état.

Si plusieurs balises sont spécifiées, la logique *ET* est utilisée: un problème doit avoir toutes les balises spécifiées dans la configuration du service pour être associé au service.

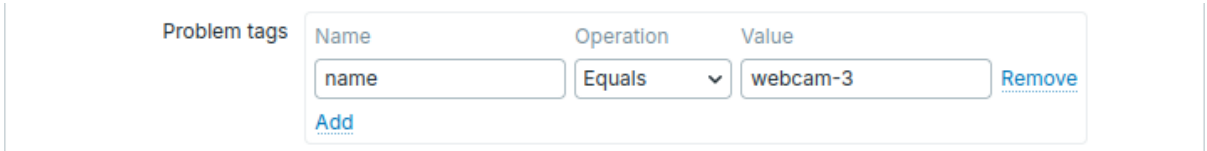


Note:
Un problème dans Zabbix hérite des balises de toute la chaîne de modèles, hôtes, éléments, scénarios web et déclencheurs. N'importe laquelle de ces balises peut être utilisée pour faire correspondre des problèmes aux services.

Exemple :

Le problème *Web camera 3 is down* a les balises `type:video-surveillance`, `floor:1` et `name:webcam-3` et le statut *Warning*

Le service **Web camera 3** a la seule balise de problème spécifiée : `name:webcam-3`



Le statut du service passera de *OK* à *Warning* lorsque ce problème sera détecté.

Si le service **Web camera 3** avait les balises de problème `name:webcam-3` et `floor:2`, son statut ne serait pas modifié lorsque le problème est détecté, car les conditions ne sont que partiellement remplies.

Modification des services existants

Note:
Les boutons décrits ci-dessous ne sont visibles que lorsque la section *Services* est en mode édi

Pour modifier un service existant, cliquez sur l'icône en forme de crayon à côté du service.

Pour cloner un service existant, cliquez sur l'icône en forme de crayon pour ouvrir sa configuration, puis cliquez sur le bouton *Clone*. Lorsqu'un service est cloné, ses liens parent sont conservés, tandis que les liens enfant ne le sont pas.

Pour supprimer un service, cliquez sur l'icône x à côté de celui-ci. Lorsque vous supprimez un service parent, ses services enfants ne seront pas supprimés et monteront d'un niveau dans l'arborescence des services (les enfants du 1er niveau auront le même niveau que le service parent supprimé).

Deux boutons sous la liste des services offrent certaines options de modification en masse :

- *Mass update* - mise à jour en masse des propriétés du service
- *Delete* - supprimer les services

Pour utiliser ces options, cochez les cases devant les services concernés, puis cliquez sur le bouton requis.

SLA

Présentation Une fois les services créés, vous pouvez commencer à surveiller si leurs performances respectent l'accord de niveau de service (SLA).

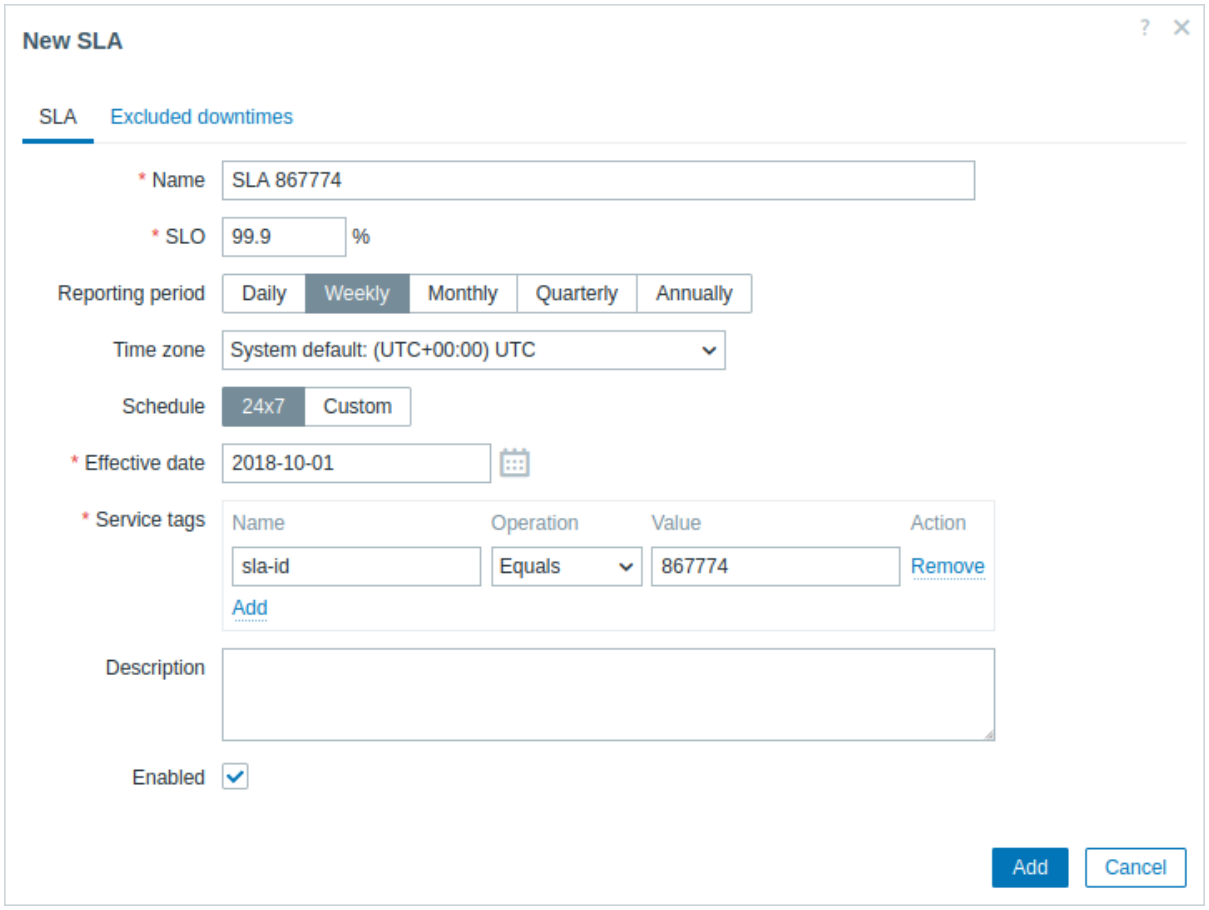
La section de menu *Services* > *SLA* permet de configurer des SLA pour différents services. Dans Zabbix, un SLA définit un objectif de niveau de service (SLO), une période de disponibilité prévue et des périodes d'indisponibilité planifiées.

Les SLA et les services sont associés à l'aide des étiquettes de service. Le même SLA peut être appliqué à plusieurs services : les performances seront mesurées séparément pour chaque service correspondant. Un même service peut être associé à plusieurs SLA : les données de chaque SLA seront affichées séparément.

Dans les rapports SLA, Zabbix fournit les données de l'indicateur de niveau de service (SLI), qui mesure la disponibilité réelle du service. Pour déterminer si un service respecte les objectifs du SLA, le SLO (disponibilité prévue en %) est comparé au SLI (disponibilité réelle en %).

Configuration Pour créer un nouveau SLA, cliquez sur le bouton *Créer un SLA*.

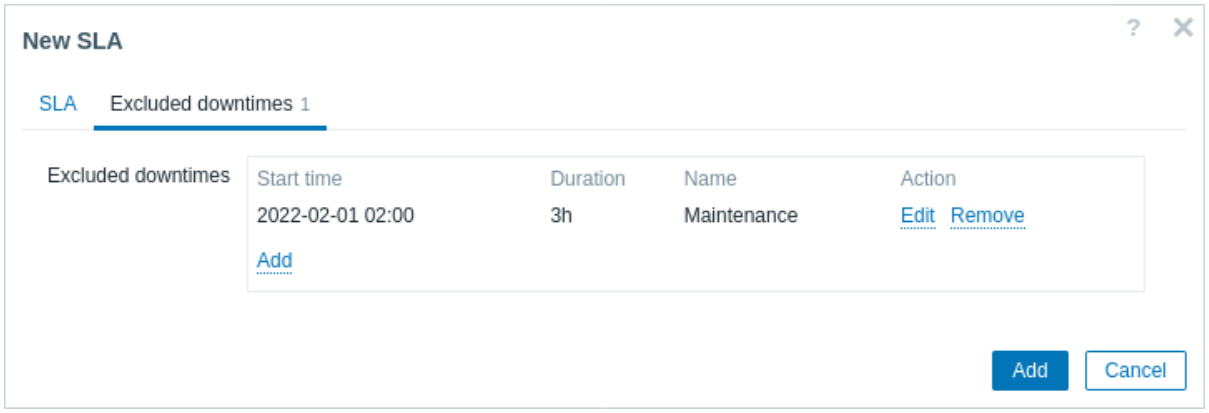
L'onglet *SLA* permet de spécifier les paramètres généraux du SLA.



Paramètre	Description
Nom	Saisissez le nom du SLA.
SLO	Saisissez l'objectif de niveau de service (SLO) en pourcentage.

Paramètre	Description
Période de rapport	Sélectionnez la période de rapport SLI utilisée pour calculer les données SLI dans le rapport SLA : **Quotidienne** - chaque jour, de 00:00:00 à 23:59:59. **Hebdomadaire** - chaque semaine, du dimanche à 00:00:00 au samedi à 23:59:59. **Mensuelle** - chaque mois, du premier jour à 00:00:00 au dernier jour à 23:59:59. **Trimestrielle** - chaque trimestre civil : T1 (janvier-mars), T2 (avril-juin), T3 (juillet-septembre), T4 (octobre-décembre). **Annuelle** - chaque année civile, du 1er janvier à 00:00:00 au 31 décembre à 23:59:59.
Fuseau horaire	Sélectionnez le fuseau horaire du SLA.
Planification	Sélectionnez la planification du SLA : 24x7 ou personnalisée.
Date d'effet	Sélectionnez la date de début du calcul du SLA.
Étiquettes de service	Ajoutez des étiquettes de service pour identifier les services auxquels ce SLA doit être appliqué. **Nom** - nom de l'étiquette de service, qui doit correspondre exactement, en respectant la casse. **Opération** - sélectionnez *Égal à* si la valeur de l'étiquette doit correspondre exactement (en respectant la casse), ou *Contient* si une partie de la valeur de l'étiquette doit correspondre (sans respecter la casse). **Valeur** - valeur de l'étiquette de service à rechercher selon l'opération sélectionnée. Le SLA est appliqué à un service si au moins une étiquette de service correspond.
Description	Ajoutez une description du SLA.
Activé	Cochez la case pour activer le calcul du SLA.

L'onglet *Temps d'arrêt exclus* permet de spécifier les temps d'arrêt exclus du calcul du SLA.



Cliquez sur *Ajouter* pour configurer les temps d'arrêt exclus, puis saisissez le nom de la période, la date de début et la durée.

Rapports SLA La performance d'un service par rapport à un SLA est visible dans le rapport SLA. Les rapports SLA peuvent être consultés :

- depuis la section *SLA* en cliquant sur le lien hypertexte du rapport SLA ;
- depuis la section *Services* en cliquant sur le nom du SLA dans l'onglet d'informations ;
- dans le widget *Rapport SLA* du tableau de bord.

Une fois qu'un SLA est configuré, l'onglet *Info* dans la section des services affichera également certaines informations sur les performances du service.

Exemple de configuration

Aperçu Cette section décrit une configuration simple pour surveiller le cluster haute disponibilité Zabbix en tant que service.

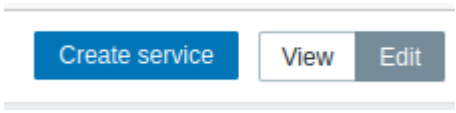
Prérequis Avant de configurer la surveillance des services, vous devez avoir configuré les hôtes :

- *Nœud HA 1* avec au moins un déclencheur et un tag (de préférence défini au niveau du déclencheur) `component:ha-node-1`
- *Nœud HA 2* avec au moins un déclencheur et un tag (de préférence défini au niveau du déclencheur) `component:ha-node-2`

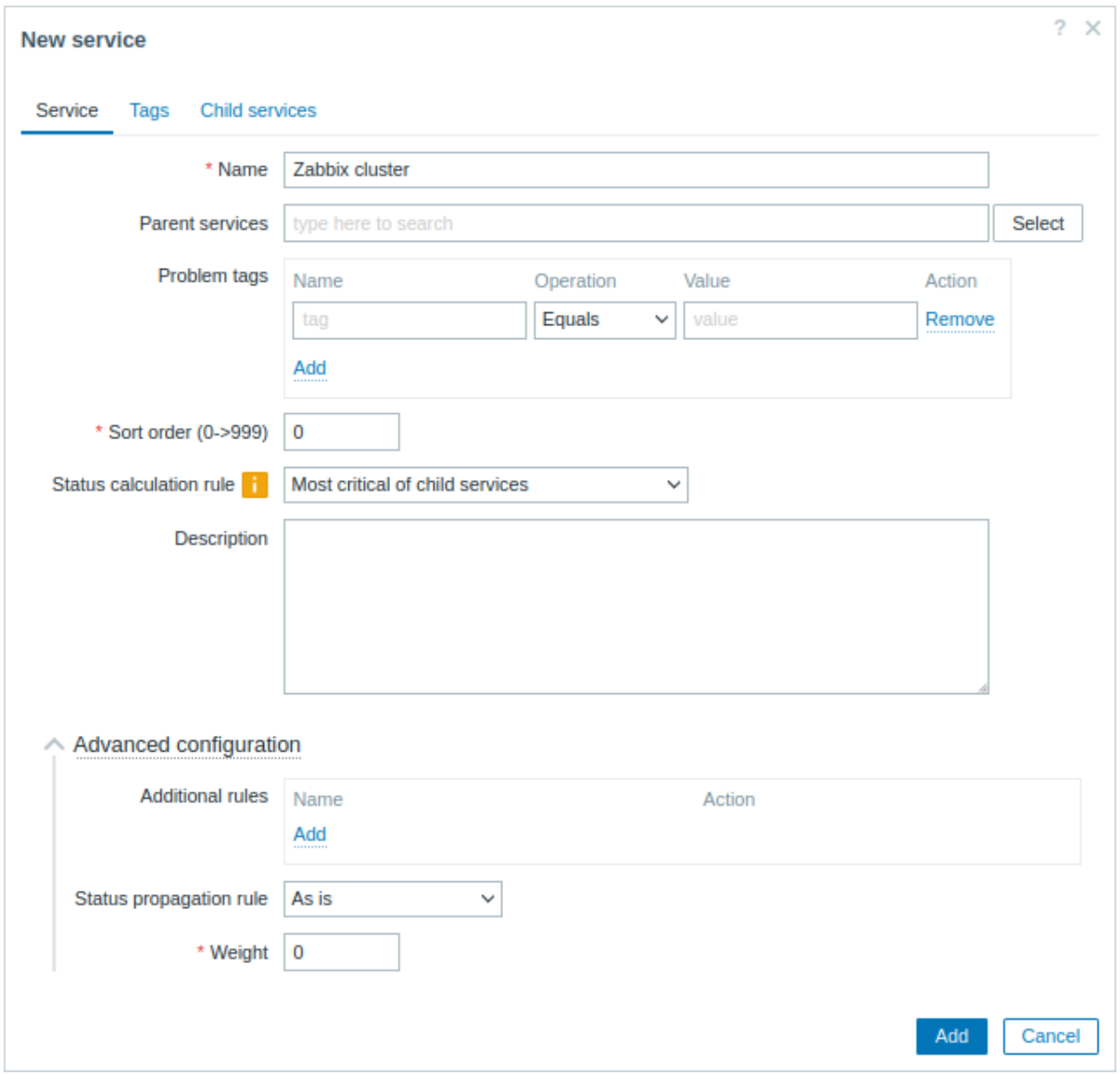
Arbre de services L'étape suivante consiste à construire l'arbre de services. Dans cet exemple, l'infrastructure est très basique et se compose de trois services: *Zabbix cluster* (parent) et deux services enfants *Zabbix server node 1* et *Zabbix server node 2*.

```
Zabbix cluster
|
|- Zabbix server node 1
|- Zabbix server node 2
```

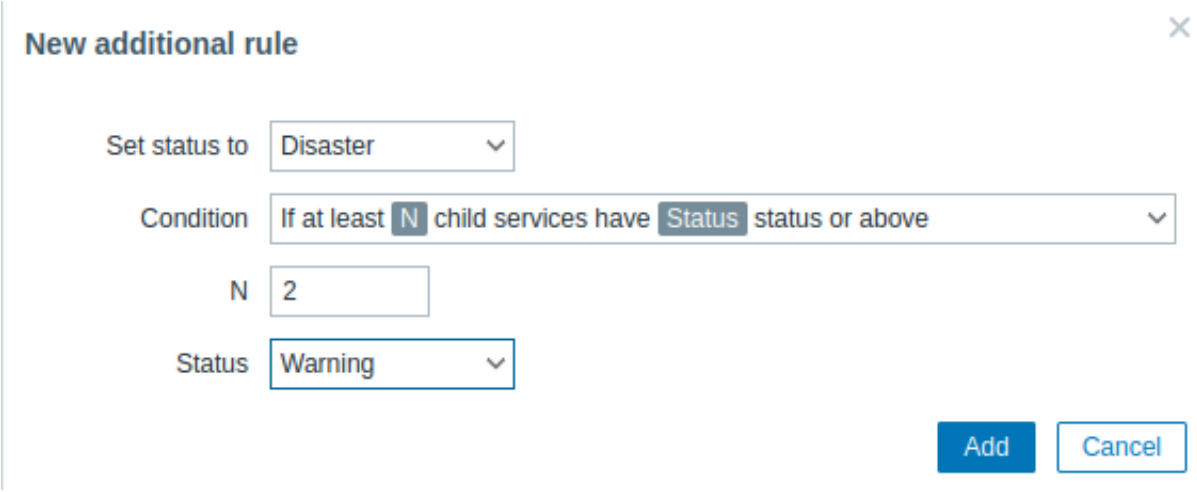
Sur la page Services, activez le mode *Edit* et cliquez sur *Create service*:



Dans la fenêtre de configuration du service, saisissez le nom *Zabbix cluster* et cliquez sur l'en-tête *Advanced configuration* pour afficher les options de configuration avancées.



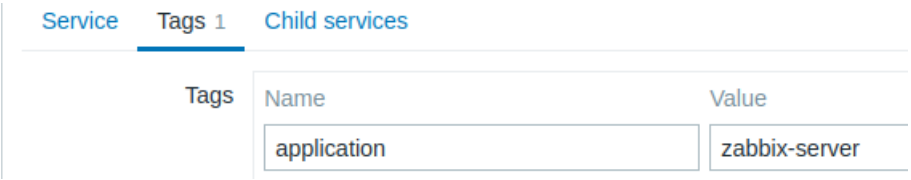
Configurez une règle supplémentaire:



Zabbix cluster aura deux services enfants - un pour chacun des nœuds HA. Si les deux nœuds HA ont des problèmes d'un statut au moins *Warning*, le statut du service parent doit être défini sur *Disaster*. Pour y parvenir, la règle supplémentaire doit être configurée comme suit:

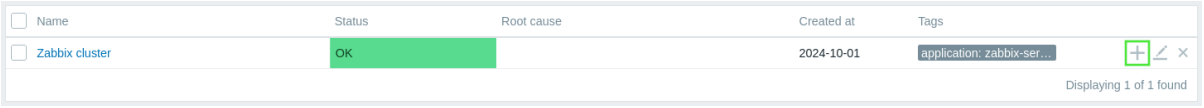
- Définir le statut sur: Disaster
- Condition: Si au moins `N` services enfants ont le statut `Status` ou supérieur
- N: 2
- Status: Warning

Passez à l'onglet *Tags* et ajoutez un tag `application:zabbix-server`. Ce tag sera utilisé plus tard pour les actions de service et les rapports SLA.



Enregistrez le nouveau service.

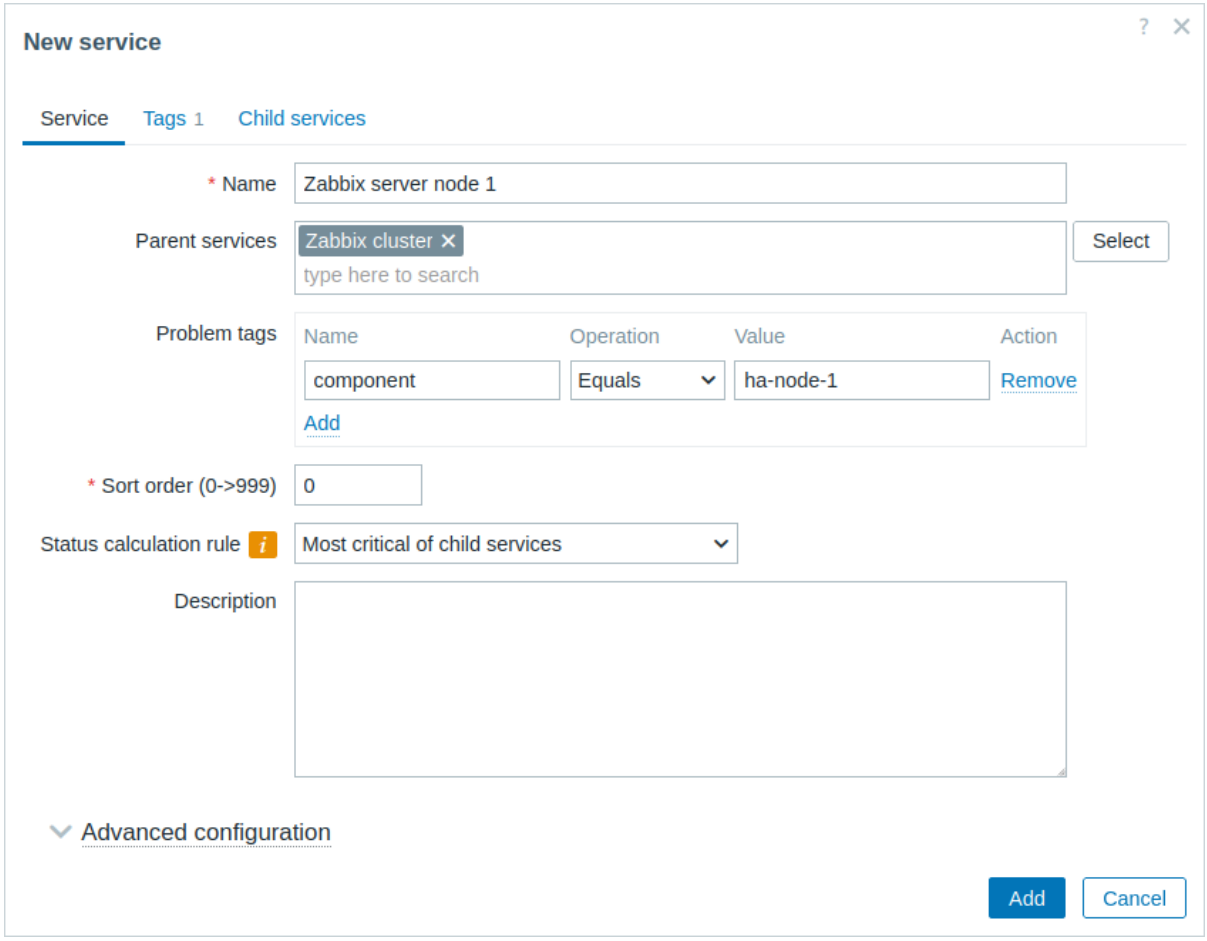
Pour ajouter un service enfant, cliquez sur l'icône plus à côté du service de cluster Zabbix (l'icône n'est visible qu'en mode *Édition*).



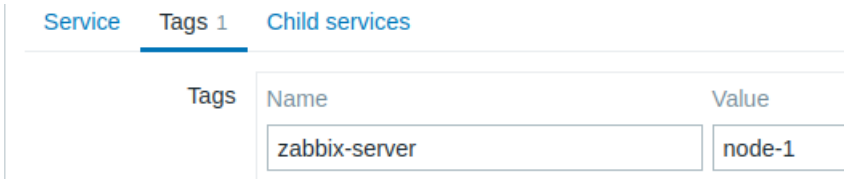
Dans la fenêtre de configuration du service, saisissez le nom *Zabbix server node 1*. Notez que le paramètre Services parents est déjà prérempli avec *Zabbix cluster*.

La disponibilité de ce service est affectée par des problèmes sur l'hôte *HA node 1*, marqués avec la balise de problème `component:ha-node-1`. Dans le paramètre Balises de problème, saisissez :

- Nom : component
- Opération : Égal à
- Valeur : ha-node-1



Passez à l'onglet *Balises* et ajoutez une balise de service : `zabbix-server:node-1`. Cette balise sera utilisée plus tard pour les actions de service et les rapports SLA.



Enregistrez le nouveau service.

Créez un autre service enfant du cluster Zabbix nommé « Zabbix server node 2 ».

Définissez les tags de problème comme suit :

- Nom : component
- Opération : Égal à
- Valeur : ha-node-2

Passez à l'onglet *Tags* et ajoutez un tag de service : `zabbix-server:node-2`.

Enregistrez le nouveau service.

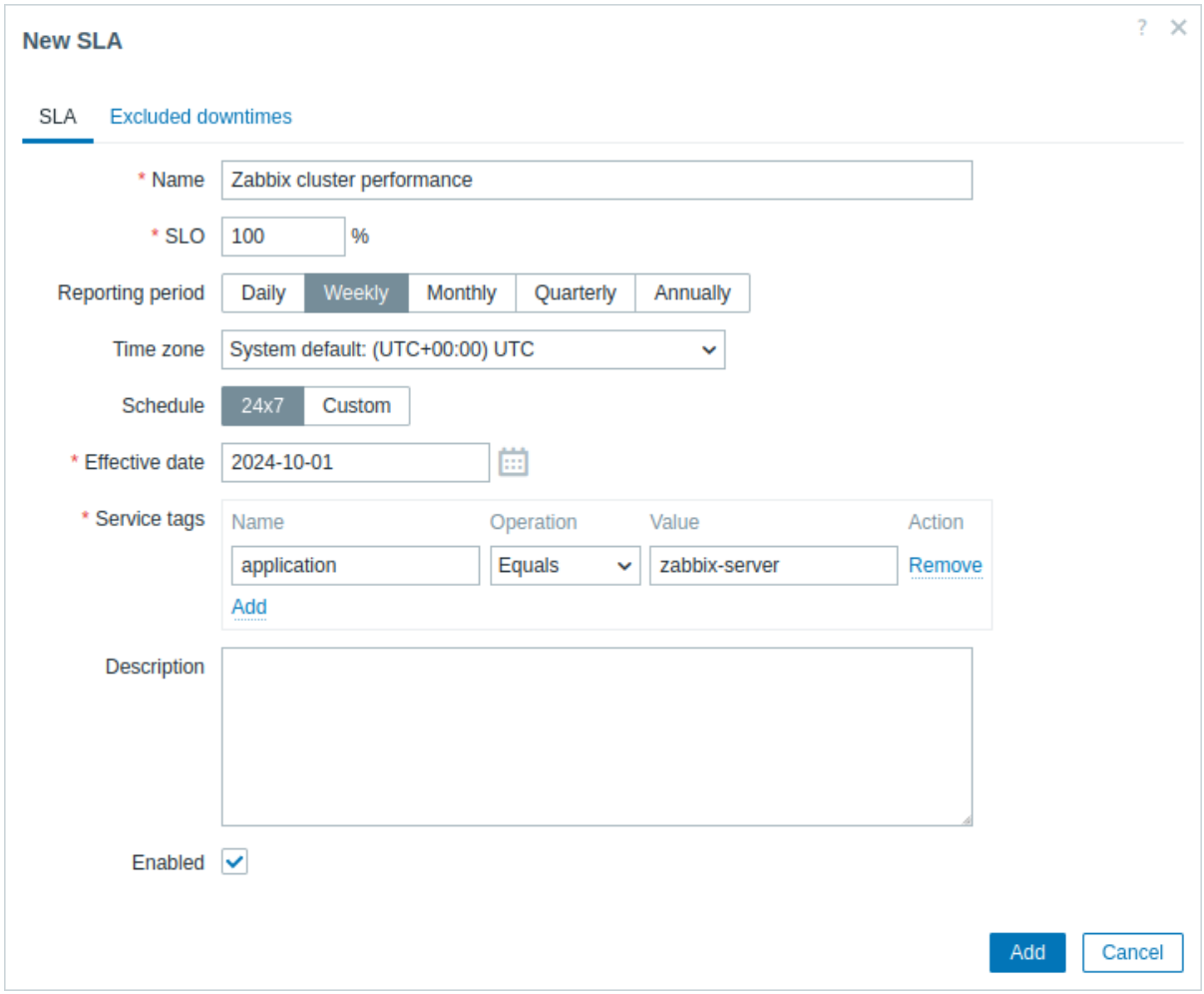
SLA Dans cet exemple, les performances attendues du cluster Zabbix sont de 100 %, à l'exclusion d'une période de maintenance d'une heure deux fois par an.

Commencez par ajouter un nouvel accord de niveau de service :

1. Accédez à *Services* > *SLA*.
2. Cliquez sur *Create SLA*.
3. Saisissez le nom *Zabbix cluster performance* et définissez le SLO sur 100 %. Le service Zabbix cluster possède le tag de service `application:zabbix-server`.
4. Pour utiliser cet SLA afin de mesurer les performances du cluster Zabbix, indiquez les valeurs suivantes dans le paramètre *Service tags* :

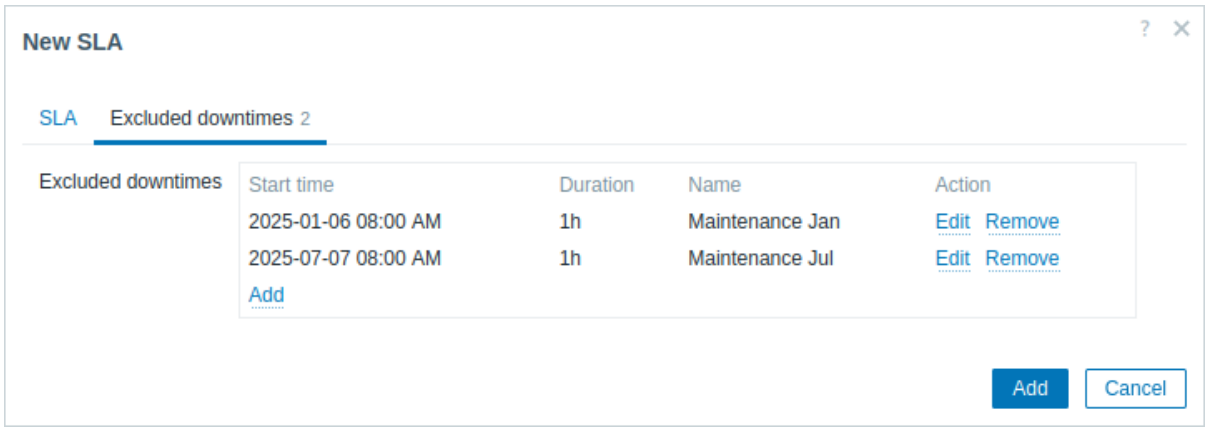
- Name : application
- Operation : Equals

- Value : zabbix-server



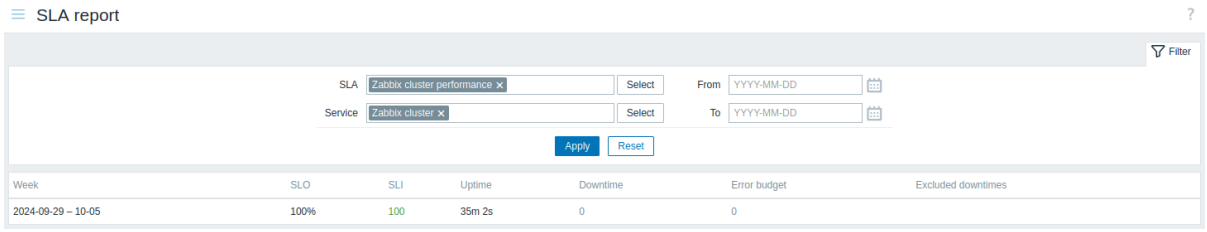
Dans un environnement réel, vous pouvez également modifier la période de rapport souhaitée, le fuseau horaire et la date de début, ou remplacer la planification 24/7 par une planification personnalisée. Pour cet exemple, les paramètres par défaut sont suffisants.

5. Accédez à l'onglet *Excluded downtimes* et ajoutez les périodes d'indisponibilité correspondant aux périodes de maintenance planifiées afin de les exclure du calcul de l'SLA.
6. Dans la section *Excluded downtimes*, cliquez sur le lien *Add*, puis saisissez le nom de la période d'indisponibilité, l'heure de début planifiée et la durée.

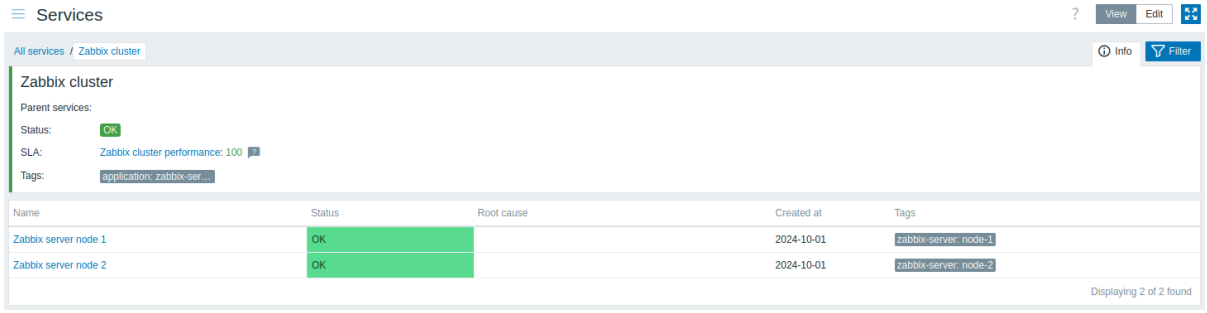


7. Cliquez sur *Add* pour enregistrer le nouvel SLA.

Accédez à la section des rapports SLA pour consulter le rapport SLA du cluster Zabbix.



Les informations relatives à l'SLA peuvent également être consultées dans la section *Services*.



Mise à niveau de la surveillance des services

Vue d'ensemble Dans Zabbix 6.0, la fonctionnalité de surveillance des services a été profondément remaniée (voir Nouveautés de Zabbix 6.0.0 pour la liste des changements).

Cette page décrit comment les services et les SLA, définis dans les versions antérieures de Zabbix, sont modifiés lors d'une mise à niveau vers Zabbix 6.0 ou une version plus récente.

Services Dans les anciennes versions de Zabbix, les services avaient deux types de dépendances: souple et forte. Après une mise à niveau, toutes les dépendances deviendront équivalentes.

Si un service "Child service" a été précédemment lié à "Parent service 1" via une dépendance forte et, en plus, à "Parent service 2" via une dépendance souple, après une mise à niveau, le "Child service" aura deux services parents, "Parent service 1" et "Parent service 2".

Le mappage entre les problèmes et les services basé sur les déclencheurs a été remplacé par un mappage basé sur les balises. Dans Zabbix 6.0 et les versions ultérieures, le formulaire de configuration du service comporte un nouveau paramètre *Problem tags*, qui permet de spécifier une ou plusieurs paires nom de balise et valeur pour la correspondance des problèmes. Les déclencheurs qui ont été liés à un service recevront une nouvelle balise `ServiceLink` : `<trigger ID>:<trigger name>` (la valeur de la balise sera tronquée à 32 caractères). Les services liés recevront la balise de problème `ServiceLink` avec la même valeur.

Règles de calcul du statut

L'« algorithme de calcul du statut » sera mis à niveau selon les règles suivantes :

- Ne pas calculer → Définir le statut sur OK
- Problème, si au moins un enfant a un problème → Le plus critique des services enfants
- Problème, si tous les enfants ont des problèmes → Le plus critique si tous les enfants ont des problèmes

SLA Auparavant, les objectifs de SLA devaient être définis séparément pour chaque service. Depuis Zabbix 6.0, le SLA est devenu une entité distincte, qui contient des informations sur le planning du service, l'objectif de niveau de service attendu (SLO) et les périodes d'indisponibilité à exclure du calcul. Une fois configuré, un SLA peut être attribué à plusieurs services via des balises de service.

Lors d'une mise à niveau :

- Les SLA identiques définis pour chaque service seront regroupés et un SLA par groupe sera créé.
- Chaque service concerné recevra une balise spéciale `SLA:<ID>` et la même balise sera spécifiée dans le paramètre *Balises de service* du SLA correspondant.
- L'heure de création du service, une nouvelle métrique dans les rapports SLA, sera définie sur 01/01/2000 00:00 pour les services existants.

Chiffrement et sécurité

Chiffrement Zabbix prend en charge les communications chiffrées entre les composants Zabbix à l'aide du protocole Transport Layer Security (TLS) v.1.2 et 1.3 (selon la bibliothèque cryptographique). Le chiffrement basé sur des certificats et le chiffrement basé sur une clé prépartagée sont pris en charge.

Le chiffrement peut être configuré pour les connexions :

- Entre le serveur Zabbix, le proxy Zabbix, l'agent Zabbix, le service web Zabbix, les utilitaires zabbix_sender et zabbix_get
- Vers la base de données Zabbix depuis l'interface Zabbix et serveur/proxy
- Entre l'interface Zabbix et le serveur Zabbix

Le chiffrement est facultatif et configurable pour des composants individuels :

- Certains proxies et agents peuvent être configurés pour utiliser un chiffrement basé sur des certificats avec le serveur, tandis que d'autres peuvent utiliser un chiffrement basé sur une clé prépartagée, et d'autres encore peuvent continuer à utiliser des communications non chiffrées (comme auparavant).
- Le serveur (proxy) peut utiliser différentes configurations de chiffrement pour différents hôtes.

Les programmes démon de Zabbix utilisent un seul port d'écoute pour les connexions entrantes chiffrées et non chiffrées. L'ajout du chiffrement ne nécessite pas d'ouvrir de nouveaux ports sur les pare-feu.

Limitations

- Les clés privées sont stockées en texte brut dans des fichiers lisibles par les composants Zabbix au démarrage.
- Les clés prépartagées sont saisies dans l'interface Zabbix et stockées en texte brut dans la base de données Zabbix.
- Le chiffrement intégré ne protège pas les communications entre le serveur web exécutant l'interface Zabbix et le navigateur web de l'utilisateur.
- Actuellement, chaque connexion chiffrée s'ouvre avec une négociation TLS complète ; aucun cache de session ni ticket n'est implémenté.
- L'ajout du chiffrement augmente le temps nécessaire aux vérifications d'éléments et aux actions, selon la latence du réseau :
  - Par exemple, si le délai des paquets est de 100 ms, l'ouverture d'une connexion TCP et l'envoi d'une requête non chiffrée prennent environ 200 ms. Avec le chiffrement, environ 1000 ms sont ajoutées pour l'établissement de la connexion TLS.
  - Il peut être nécessaire d'augmenter les délais d'attente, sinon certains éléments et actions exécutant des scripts distants sur les agents peuvent fonctionner avec des connexions non chiffrées, mais échouer par dépassement de délai avec le chiffrement.
- Le chiffrement n'est pas pris en charge par la découverte réseau. Les vérifications de l'agent Zabbix effectuées par la découverte réseau seront non chiffrées et, si l'agent Zabbix est configuré pour refuser les connexions non chiffrées, ces vérifications échoueront.

Compilation de Zabbix avec prise en charge du chiffrement Pour prendre en charge le chiffrement, Zabbix doit être compilé et lié à l'une des bibliothèques cryptographiques prises en charge :

- GnuTLS - à partir de la version 3.1.18
- OpenSSL - versions 1.0.1, 1.0.2, 1.1.0, 1.1.1, 3.0.x - 3.5.x
- LibreSSL - testé avec les versions 2.7.4 et 2.8.2 :
  - LibreSSL 2.6.x n'est pas pris en charge
  - LibreSSL est pris en charge en tant que remplacement compatible d'OpenSSL ; les nouvelles fonctions d'API spécifiques à LibreSSL `tls_*()` ne sont pas utilisées. Les composants Zabbix compilés avec LibreSSL ne pourront pas utiliser de clés prépartagées (PSK) ; seuls les certificats pourront être utilisés.

> **Note:**
> Pour en savoir plus sur la configuration de SSL pour l'interface Zabbix, consultez ces bonnes pratiques.

La bibliothèque est sélectionnée en spécifiant l'option correspondante au script « configure » :

- `--with-gnutls[=DIR]`
- `--with-openssl[=DIR]` (également utilisé pour LibreSSL)

Par exemple, pour configurer les sources du serveur et de l'agent avec *OpenSSL*, vous pouvez utiliser une commande similaire à la suivante :

```
./configure --enable-server --enable-agent --with-mysql --enable-ipv6 --with-net-snmp --with-libcurl --wit
```

Différents composants Zabbix peuvent être compilés avec différentes bibliothèques cryptographiques (par exemple, un serveur avec *OpenSSL* et un agent avec *GnuTLS*).

Attention:
Si vous prévoyez d'utiliser des clés prépartagées (PSK), envisagez d'utiliser les bibliothèques *GnuTLS* ou *OpenSSL 1.1.0* (ou une version plus récente) dans les composants Zabbix utilisant des PSK. Les bibliothèques *GnuTLS* et *OpenSSL 1.1.0* prennent en charge les suites de chiffrement PSK avec la confidentialité persistante parfaite. Les versions plus anciennes de la bibliothèque *OpenSSL* (1.0.1, 1.0.2c) prennent également en charge les PSK, mais les suites de chiffrement PSK disponibles n'offrent pas de confidentialité persistante parfaite.

Gestion du chiffrement des connexions Les connexions dans Zabbix peuvent utiliser :

- aucun chiffrement (par défaut)
- un chiffrement basé sur des certificats RSA
- un chiffrement basé sur une PSK

Deux paramètres importants permettent de spécifier le chiffrement entre les composants Zabbix :

- TLSConnect - spécifie le chiffrement à utiliser pour les connexions sortantes (sans chiffrement, PSK ou certificat)
- TLSAccept - spécifie les types de connexions autorisés pour les connexions entrantes (sans chiffrement, PSK ou certificat). Une ou plusieurs valeurs peuvent être spécifiées.

`TLSConnect` est utilisé dans les fichiers de configuration du proxy Zabbix (en mode actif, spécifie uniquement les connexions au serveur) et de l'agent Zabbix (pour les vérifications actives). Dans l'interface Zabbix, l'équivalent de `TLSConnect` est le champ *Connexions vers l'hôte* dans *Collecte de données* > *Hôtes* > <un hôte> > onglet *Chiffrement*, ainsi que le champ *Connexions vers le proxy* dans *Administration* > *Proxies* > <un proxy> > onglet *Chiffrement*. Si le type de chiffrement configuré pour la connexion échoue, aucun autre type de chiffrement ne sera essayé.

`TLSAccept` est utilisé dans les fichiers de configuration du proxy Zabbix (en mode passif, spécifie uniquement les connexions provenant du serveur) et de l'agent Zabbix (pour les vérifications passives). Dans l'interface Zabbix, l'équivalent de `TLSAccept` est le champ *Connexions depuis l'hôte* dans *Collecte de données* > *Hôtes* > <un hôte> > onglet *Chiffrement*, ainsi que le champ *Connexions depuis le proxy* dans *Administration* > *Proxies* > <un proxy> > onglet *Chiffrement*.

Normalement, vous ne configurez qu'un seul type de chiffrement pour les connexions entrantes. Cependant, vous pouvez vouloir changer le type de chiffrement, par exemple passer d'une connexion sans chiffrement à une connexion basée sur un certificat, avec un temps d'arrêt minimal et la possibilité d'effectuer une restauration. Pour ce faire :

- Définissez `TLSAccept=unencrypted,cert` dans le fichier de configuration de l'agent et redémarrez l'agent Zabbix.
- Testez la connexion avec `zabbix_get` vers l'agent à l'aide d'un certificat. Si cela fonctionne, vous pouvez reconfigurer le chiffrement de cet agent dans l'interface Zabbix, dans *Collecte de données* > *Hôtes* > <un hôte> > onglet *Chiffrement*, en définissant *Connexions vers l'hôte* sur *Certificat*.
- Lorsque le cache de configuration du serveur est mis à jour (et que la configuration du proxy est mise à jour si l'hôte est surveillé par un proxy), les connexions à cet agent seront chiffrées.
- Si tout fonctionne comme prévu, vous pouvez définir `TLSAccept=cert` dans le fichier de configuration de l'agent et redémarrer l'agent Zabbix. L'agent n'acceptera alors que les connexions chiffrées basées sur un certificat. Les connexions sans chiffrement et celles basées sur une PSK seront rejetées.

Le fonctionnement est similaire sur le serveur et le proxy. Si, dans l'interface Zabbix, le champ *Connexions depuis l'hôte* de la configuration de l'hôte est défini sur « Certificat », seules les connexions chiffrées basées sur un certificat seront acceptées depuis l'agent (vérifications actives) et `zabbix_sender` (éléments trapper).

Il est très probable que vous configuriez les connexions entrantes et sortantes pour utiliser le même type de chiffrement, ou aucun chiffrement. Cependant, il est techniquement possible de les configurer de manière asymétrique, par exemple avec un chiffrement basé sur un certificat pour les connexions entrantes et un chiffrement basé sur une PSK pour les connexions sortantes.

La configuration du chiffrement de chaque hôte est affichée dans l'interface Zabbix, dans *Collecte de données* > *Hôtes*, dans la colonne *Chiffrement de l'agent*. Par exemple :

Exemple	Connexions vers l'hôte	Connexions autorisées depuis l'hôte	Connexions rejetées depuis l'hôte
NONE	Sans chiffrement	Sans chiffrement	Chiffrées, basées sur un certificat ou une PSK
CERT NONE PSK CERT	Chiffrées, basées sur un certificat	Chiffrées, basées sur un certificat	Sans chiffrement et chiffrées, basées sur une PSK
PSK NONE PSK CERT	Chiffrées, basées sur une PSK	Chiffrées, basées sur une PSK	Sans chiffrement et chiffrées, basées sur un certificat

Exemple	Connexions vers l'hôte	Connexions autorisées depuis l'hôte	Connexions rejetées depuis l'hôte
PSK NONE PSK CERT	Chiffrées, basées sur une PSK	Sans chiffrement et chiffrées, basées sur une PSK	Chiffrées, basées sur un certificat
CERT NONE PSK CERT	Chiffrées, basées sur un certificat	Sans chiffrement, chiffrées avec une PSK ou basées sur un certificat	-

Attention:
Les connexions ne sont pas chiffrées par défaut. Le chiffrement doit être configuré individuellement pour chaque hôte et chaque proxy.

zabbix_get et zabbix_sender avec chiffrement Voir les pages de manuel zabbix_get et zabbix_sender pour leur utilisation avec le chiffrement.

Suites de chiffrement Les suites de chiffrement sont configurées par défaut en interne lors du démarrage de Zabbix.

Les suites de chiffrement configurées par l'utilisateur sont également prises en charge pour GnuTLS et OpenSSL. Les utilisateurs peuvent configurer les suites de chiffrement selon leurs politiques de sécurité. L'utilisation de cette fonctionnalité est facultative (les suites de chiffrement par défaut intégrées continuent de fonctionner).

Pour les bibliothèques cryptographiques compilées avec les paramètres par défaut, les règles intégrées de Zabbix produisent généralement les suites de chiffrement suivantes (dans l'ordre de priorité décroissante) :

Library	Certificate ciphersuites	PSK ciphersuites
GnuTLS 3.1.18	TLS_ECDHE_RSA_AES_128_GCM_SHA256 TLS_ECDHE_RSA_AES_128_CBC_SHA256 TLS_ECDHE_RSA_AES_128_CBC_SHA1 TLS_RSA_AES_128_GCM_SHA256 TLS_RSA_AES_128_CBC_SHA256 TLS_RSA_AES_128_CBC_SHA1	TLS_ECDHE_PSK_AES_128_CBC_SHA256 TLS_ECDHE_PSK_AES_128_CBC_SHA1 TLS_PSK_AES_128_GCM_SHA256 TLS_PSK_AES_128_CBC_SHA256 TLS_PSK_AES_128_CBC_SHA1
OpenSSL 1.0.2c	ECDHE-RSA-AES128-GCM-SHA256 ECDHE-RSA-AES128-SHA256 ECDHE-RSA-AES128-SHA AES128-GCM-SHA256 AES128-SHA256 AES128-SHA	PSK-AES128-CBC-SHA
OpenSSL 1.1.0	ECDHE-RSA-AES128-GCM-SHA256 ECDHE-RSA-AES128-SHA256 ECDHE-RSA-AES128-SHA AES128-GCM-SHA256 AES128-CCM8 AES128-CCM AES128-SHA256 AES128-SHA	ECDHE-PSK-AES128-CBC-SHA256 ECDHE-PSK-AES128-CBC-SHA PSK-AES128-GCM-SHA256 PSK-AES128-CCM8 PSK-AES128-CCM PSK-AES128-CBC-SHA256 PSK-AES128-CBC-SHA
OpenSSL 1.1.1d	TLS_AES_256_GCM_SHA384 TLS_CHACHA20_POLY1305_SHA256 TLS_AES_128_GCM_SHA256 ECDHE-RSA-AES128-GCM-SHA256 ECDHE-RSA-AES128-SHA256 ECDHE-RSA-AES128-SHA AES128-GCM-SHA256 AES128-CCM8 AES128-CCM AES128-SHA256 AES128-SHA	TLS_CHACHA20_POLY1305_SHA256 TLS_AES_128_GCM_SHA256 ECDHE-PSK-AES128-CBC-SHA256 ECDHE-PSK-AES128-CBC-SHA PSK-AES128-GCM-SHA256 PSK-AES128-CCM8 PSK-AES128-CCM PSK-AES128-CBC-SHA256 PSK-AES128-CBC-SHA

Suites de chiffrement configurées par l'utilisateur Les critères de sélection des suites de chiffrement intégrés peuvent être remplacés par des suites de chiffrement configurées par l'utilisateur.

Attention:
Les suites de chiffrement configurées par l'utilisateur sont une fonctionnalité destinée aux utilisateurs avancés qui comprennent les suites de chiffrement TLS, leur sécurité et les conséquences d'erreurs, et qui sont à l'aise avec le dépannage TLS.

Les critères de sélection des suites de chiffrement intégrés peuvent être remplacés à l'aide des paramètres suivants :

Portée du remplacement	Paramètre	Valeur	Description
Sélection des suites de chiffrement pour les certificats	TLSCipherCert13	Chaînes de chiffrement OpenSSL 1.1.1 valides cipher strings pour le protocole TLS 1.3 (leurs valeurs sont transmises à la fonction OpenSSL `SSL_CTX_set_ciphersuites()`).	Critères de sélection des suites de chiffrement basés sur les certificats pour TLS 1.3 OpenSSL 1.1.1 ou version plus récente uniquement.
	TLSCipherCert	Chaînes de chiffrement OpenSSL valides cipher strings pour TLS 1.2 ou chaînes de priorité GnuTLS valides priority strings. Leurs valeurs sont transmises respectivement aux fonctions `SSL_CTX_set_cipher_list()` ou `gnutls_priority_init()`.	Critères de sélection des suites de chiffrement basés sur les certificats pour TLS 1.2/1.3 (GnuTLS), TLS 1.2 (OpenSSL)
Sélection des suites de chiffrement pour PSK	TLSCipherPSK13	Chaînes de chiffrement OpenSSL 1.1.1 valides cipher strings pour le protocole TLS 1.3 (leurs valeurs sont transmises à la fonction OpenSSL `SSL_CTX_set_ciphersuites()`).	Critères de sélection des suites de chiffrement basés sur PSK pour TLS 1.3 OpenSSL 1.1.1 ou version plus récente uniquement.
	TLSCipherPSK	Chaînes de chiffrement OpenSSL valides cipher strings pour TLS 1.2 ou chaînes de priorité GnuTLS valides priority strings. Leurs valeurs sont transmises respectivement aux fonctions `SSL_CTX_set_cipher_list()` ou `gnutls_priority_init()`.	Critères de sélection des suites de chiffrement basés sur PSK pour TLS 1.2/1.3 (GnuTLS), TLS 1.2 (OpenSSL)
Liste combinée des suites de chiffrement pour certificat et PSK	TLSCipherAll13	Chaînes de chiffrement OpenSSL 1.1.1 valides cipher strings pour le protocole TLS 1.3 (leurs valeurs sont transmises à la fonction OpenSSL `SSL_CTX_set_ciphersuites()`).	Critères de sélection des suites de chiffrement pour TLS 1.3 OpenSSL 1.1.1 ou version plus récente uniquement.
	TLSCipherAll	Chaînes de chiffrement OpenSSL valides cipher strings pour TLS 1.2 ou chaînes de priorité GnuTLS valides priority strings. Leurs valeurs sont transmises respectivement aux fonctions `SSL_CTX_set_cipher_list()` ou `gnutls_priority_init()`.	Critères de sélection des suites de chiffrement pour TLS 1.2/1.3 (GnuTLS), TLS 1.2 (OpenSSL)

Pour remplacer la sélection des suites de chiffrement dans les utilitaires zabbix_get et zabbix_sender, utilisez les paramètres de ligne de commande :

- `--tls-cipher13`
- `--tls-cipher`

Les nouveaux paramètres sont facultatifs. Si un paramètre n'est pas spécifié, la valeur interne par défaut est utilisée. Si un paramètre est défini, il ne peut pas être vide.

Si la définition d'une valeur TLSCipher* dans la bibliothèque cryptographique échoue, le serveur, le proxy ou l'agent ne démarrera pas et une erreur sera consignée.

Il est important de comprendre quand chaque paramètre s'applique.

Connexions sortantes

Le cas le plus simple est celui des connexions sortantes :

- Pour les connexions sortantes avec certificat - utilisez TLSCipherCert13 ou TLSCipherCert
- Pour les connexions sortantes avec PSK - utilisez TLSCipherPSK13 ou TLSCipherPSK
- Dans le cas des utilitaires zabbix_get et zabbix_sender, les paramètres de ligne de commande `--tls-cipher13` ou `--tls-cipher` peuvent être utilisés (le chiffrement est spécifié sans ambiguïté avec un paramètre `--tls-connect`)

Connexions entrantes

La situation est un peu plus complexe pour les connexions entrantes, car les règles sont spécifiques aux composants et à la configuration.

Pour l'**agent** Zabbix :

Configuration de connexion de l'agent	Configuration des chiffrements
TLSConnect=cert	TLSCipherCert, TLSCipherCert13
TLSConnect=psk	TLSCipherPSK, TLSCipherPSK13
TLSAccept=cert	TLSCipherCert, TLSCipherCert13
TLSAccept=psk	TLSCipherPSK, TLSCipherPSK13
TLSAccept=cert,psk	TLSCipherAll, TLSCipherAll13

Pour le **serveur** et le **proxy** Zabbix :

Configuration de connexion	Configuration des chiffrements
Connexions sortantes utilisant PSK	TLSCipherPSK, TLSCipherPSK13
Connexions entrantes utilisant des certificats	TLSCipherAll, TLSCipherAll13
Connexions entrantes utilisant PSK si le serveur n'a pas de certificat	TLSCipherPSK, TLSCipherPSK13
Connexions entrantes utilisant PSK si le serveur a un certificat	TLSCipherAll, TLSCipherAll13

On peut observer un certain schéma dans les deux tableaux ci-dessus :

- TLSCipherAll et TLSCipherAll13 ne peuvent être spécifiés que si une liste combinée de suites de chiffrement basées sur les certificats **et** sur PSK est utilisée. Cela se produit dans deux cas : serveur (proxy) avec un certificat configuré (les suites de chiffrement PSK sont toujours configurées sur le serveur, et sur le proxy si la bibliothèque cryptographique prend en charge PSK), agent configuré pour accepter à la fois les connexions entrantes basées sur des certificats et celles basées sur PSK
- dans les autres cas, TLSCipherCert* et/ou TLSCipherPSK* suffisent

Les tableaux suivants présentent les valeurs par défaut intégrées de `TLSCipher*`. Elles peuvent constituer un bon point de départ pour vos propres valeurs personnalisées.

Paramètre	GnuTLS 3.6.12
TLSCipherCert	NONE:+VERS-TLS1.2:+ECDHE-RSA:+RSA:+AES-128-GCM:+AES-128-CBC:+AEAD:+SHA256:+SHA1:+CURVE-ALL:+COMP-NULL:+SIGN-ALL:+CTYPE-X.509
TLSCipherPSK	NONE:+VERS-TLS1.2:+ECDHE-PSK:+PSK:+AES-128-GCM:+AES-128-CBC:+AEAD:+SHA256:+SHA1:+CURVE-ALL:+COMP-NULL:+SIGN-ALL
TLSCipherAll	NONE:+VERS-TLS1.2:+ECDHE-RSA:+RSA:+ECDHE-PSK:+PSK:+AES-128-GCM:+AES-128-CBC:+AEAD:+SHA256:+SHA1:+CURVE-ALL:+COMP-NULL:+SIGN-ALL:+CTYPE-X.509

Paramètre	OpenSSL 1.1.1d [1]
TLSCipherCert13	
TLSCipherCert	EECDH+aRSA+AES128:RSA+aRSA+AES128
TLSCipherPSK13	TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256
TLSCipherPSK	kECDHEPSK+AES128:kPSK+AES128
TLSCipherAll13	
TLSCipherAll	EECDH+aRSA+AES128:RSA+aRSA+AES128:kECDHEPSK+AES128:kPSK+AES128

[1] Les valeurs par défaut sont différentes pour les versions plus anciennes d'OpenSSL (1.0.1, 1.0.2, 1.1.0), pour LibreSSL et si OpenSSL est compilé sans prise en charge de PSK.

Exemples de suites de chiffrement configurées par l'utilisateur

Voir ci-dessous les exemples suivants de suites de chiffrement configurées par l'utilisateur :

- Test des chaînes de chiffrement et autorisation des suites de chiffrement PFS uniquement
- Passage de AES128 à AES256

Tester les chaînes de chiffrement et autoriser uniquement les suites de chiffrement PFS

Pour voir quelles suites de chiffrement ont été sélectionnées, vous devez définir `DebugLevel=4` dans le fichier de configuration ou utiliser l'option `-vv` pour zabbix_sender.

Il peut être nécessaire de faire quelques essais avec les paramètres `TLSCipher*` avant d'obtenir les suites de chiffrement souhaitées. Il n'est pas pratique de redémarrer plusieurs fois le serveur Zabbix, le proxy ou l'agent simplement pour ajuster les paramètres `TLSCipher*`. Des options plus pratiques consistent à utiliser zabbix_sender ou la commande `openssl`. Examinons les deux.

1. Utilisation de zabbix_sender.

Créons un fichier de configuration de test, par exemple `/home/zabbix/test.conf`, avec la syntaxe d'un fichier `zabbix_agentd.conf` :

```
Hostname=nonexisting
ServerActive=nonexisting

TLSConnect=cert
TLSCAFile=/home/zabbix/ca.crt
TLSCertFile=/home/zabbix/agent.crt
TLSKeyFile=/home/zabbix/agent.key
TLSPSKIdentity=nonexisting
TLSPSKFile=/home/zabbix/agent.psk
```

Vous avez besoin de certificats CA et agent valides ainsi que d'une clé PSK pour cet exemple. Adaptez les chemins et les noms des fichiers de certificats et de clé PSK à votre environnement.

Si vous n'utilisez pas de certificats, mais uniquement une clé PSK, vous pouvez créer un fichier de test plus simple :

```
Hostname=nonexisting
ServerActive=nonexisting

TLSConnect=psk
TLSPSKIdentity=nonexisting
TLSPSKFile=/home/zabbix/agentd.psk
```

Les suites de chiffrement sélectionnées peuvent être affichées en exécutant zabbix_sender (exemple compilé avec OpenSSL 1.1.d) :

```
$ zabbix_sender -vv -c /home/zabbix/test.conf -k nonexisting_item -o 1 2>&1 | grep ciphersuites
zabbix_sender [41271]: DEBUG: zbx_tls_init_child() certificate ciphersuites: TLS_AES_256_GCM_SHA384 TLS_
zabbix_sender [41271]: DEBUG: zbx_tls_init_child() PSK ciphersuites: TLS_CHACHA20_POLY1305_SHA256 TLS_AE
zabbix_sender [41271]: DEBUG: zbx_tls_init_child() certificate and PSK ciphersuites: TLS_AES_256_GCM_SHA
```

Vous voyez ici les suites de chiffrement sélectionnées par défaut. Ces valeurs par défaut sont choisies afin d'assurer l'interopérabilité avec les agents Zabbix exécutés sur des systèmes utilisant des versions plus anciennes d'OpenSSL (à partir de la version 1.0.1).

Sur les systèmes plus récents, vous pouvez renforcer la sécurité en n'autorisant que quelques suites de chiffrement, par exemple uniquement celles prenant en charge la PFS (Perfect Forward Secrecy, confidentialité persistante parfaite). Configurons les paramètres `TLSCipher*` afin de n'autoriser que les suites de chiffrement prenant en charge la PFS.

> **Attention:**
> Le résultat ne sera pas interopérable avec les systèmes utilisant OpenSSL 1.0.1 et 1.0.2 si une clé PSK est utilisée. Le chiffrement basé sur des certificats devrait fonctionner.

Ajoutez deux lignes au fichier de configuration `test.conf` :

```
TLSCipherCert=EECDH+aRSA+AES128
TLSCipherPSK=kECDHEPSK+AES128
```

Puis effectuez à nouveau le test :

```
$ zabbix_sender -vv -c /home/zabbix/test.conf -k nonexisting_item -o 1 2>&1 | grep ciphersuites
zabbix_sender [42892]: DEBUG: zbx_tls_init_child() certificate ciphersuites: TLS_AES_256_GCM_SHA384 TLS_
zabbix_sender [42892]: DEBUG: zbx_tls_init_child() PSK ciphersuites: TLS_CHACHA20_POLY1305_SHA256 TLS_AE
zabbix_sender [42892]: DEBUG: zbx_tls_init_child() certificate and PSK ciphersuites: TLS_AES_256_GCM_SHA
```

Les listes « certificate ciphersuites » et « PSK ciphersuites » ont changé : elles sont plus courtes qu'auparavant et contiennent uniquement les suites de chiffrement TLS 1.3 et les suites de chiffrement TLS 1.2 ECDHE-*, comme prévu.

2. TLSCipherAll et TLSCipherAll13 ne peuvent pas être testés avec zabbix_sender ; ils n'ont aucun effet sur la valeur « certificate and PSK ciphersuites » affichée dans l'exemple ci-dessus. Pour ajuster TLSCipherAll et TLSCipherAll13, vous devez faire des essais avec l'agent, le proxy ou le serveur.

Ainsi, pour n'autoriser que les suites de chiffrement PFS, vous devrez peut-être ajouter jusqu'à trois paramètres

```
TLSCipherCert=EECDH+aRSA+AES128
TLSCipherPSK=kECDHEPSK+AES128
TLSCipherAll=EECDH+aRSA+AES128:kECDHEPSK+AES128
```

à `zabbix_agentd.conf`, `zabbix_proxy.conf` et `zabbix_server.conf` si chacun d'eux possède un certificat configuré et si l'agent dispose également d'une clé PSK.

Si votre environnement Zabbix utilise uniquement un chiffrement basé sur une clé PSK et aucun certificat, un seul paramètre suffit :

```
TLSCipherPSK=kECDHEPSK+AES128
```

Maintenant que vous comprenez le fonctionnement, vous pouvez tester la sélection des suites de chiffrement même en dehors de Zabbix, avec la commande `openssl`. Testons les trois valeurs des paramètres `TLSCipher*` :

```
$ openssl ciphers EECDH+aRSA+AES128 | sed 's/:/ /g'
TLS_AES_256_GCM_SHA384 TLS_CHACHA20_POLY1305_SHA256 TLS_AES_128_GCM_SHA256 ECDHE-RSA-AES128-GCM-SHA256 E
$ openssl ciphers kECDHEPSK+AES128 | sed 's/:/ /g'
TLS_AES_256_GCM_SHA384 TLS_CHACHA20_POLY1305_SHA256 TLS_AES_128_GCM_SHA256 ECDHE-PSK-AES128-CBC-SHA256 E
$ openssl ciphers EECDH+aRSA+AES128:kECDHEPSK+AES128 | sed 's/:/ /g'
TLS_AES_256_GCM_SHA384 TLS_CHACHA20_POLY1305_SHA256 TLS_AES_128_GCM_SHA256 ECDHE-RSA-AES128-GCM-SHA256 E
```

Vous pouvez préférer la commande `openssl ciphers` avec l'option `-V` pour obtenir une sortie plus détaillée :

```
$ openssl ciphers -V EECDH+aRSA+AES128:kECDHEPSK+AES128
          0x13,0x02 - TLS_AES_256_GCM_SHA384  TLSv1.3 Kx=any      Au=any  Enc=AESGCM(256) Mac=AEAD
          0x13,0x03 - TLS_CHACHA20_POLY1305_SHA256 TLSv1.3 Kx=any      Au=any  Enc=CHACHA20/POLY1305(256
          0x13,0x01 - TLS_AES_128_GCM_SHA256  TLSv1.3 Kx=any      Au=any  Enc=AESGCM(128) Mac=AEAD
          0xC0,0x2F - ECDHE-RSA-AES128-GCM-SHA256 TLSv1.2 Kx=ECDH     Au=RSA  Enc=AESGCM(128) Mac=AEAD
          0xC0,0x27 - ECDHE-RSA-AES128-SHA256 TLSv1.2 Kx=ECDH     Au=RSA  Enc=AES(128)  Mac=SHA256
          0xC0,0x13 - ECDHE-RSA-AES128-SHA    TLSv1 Kx=ECDH     Au=RSA  Enc=AES(128)  Mac=SHA1
          0xC0,0x37 - ECDHE-PSK-AES128-CBC-SHA256 TLSv1 Kx=ECDHEPSK Au=PSK  Enc=AES(128)  Mac=SHA256
          0xC0,0x35 - ECDHE-PSK-AES128-CBC-SHA TLSv1 Kx=ECDHEPSK Au=PSK  Enc=AES(128)  Mac=SHA1
```

De même, vous pouvez tester les chaînes de priorité pour GnuTLS :

```
$ gnutls-cli -l --priority=NONE:+VERS-TLS1.2:+ECDHE-RSA:+AES-128-GCM:+AES-128-CBC:+AEAD:+SHA256:+CURVE-A
Cipher suites for NONE:+VERS-TLS1.2:+ECDHE-RSA:+AES-128-GCM:+AES-128-CBC:+AEAD:+SHA256:+CURVE-ALL:+COMP-
TLS_ECDHE_RSA_AES_128_GCM_SHA256                        0xc0, 0x2f      TLS1.2
TLS_ECDHE_RSA_AES_128_CBC_SHA256                        0xc0, 0x27      TLS1.2

Protocols: VERS-TLS1.2
Ciphers: AES-128-GCM, AES-128-CBC
MACs: AEAD, SHA256
Key Exchange Algorithms: ECDHE-RSA
Groups: GROUP-SECP256R1, GROUP-SECP384R1, GROUP-SECP521R1, GROUP-X25519, GROUP-X448, GROUP-FFDHE2048, GR
PK-signatures: SIGN-RSA-SHA256, SIGN-RSA-PSS-SHA256, SIGN-RSA-PSS-RSAE-SHA256, SIGN-ECDSA-SHA256, SIGN-E
```

Passage de AES128 à AES256

Zabbix utilise AES128 comme valeur par défaut intégrée pour les données. Supposons que vous utilisiez des certificats et que vous souhaitiez passer à AES256, sur OpenSSL 1.1.1.

Cela peut être réalisé en ajoutant les paramètres correspondants dans `zabbix_server.conf` :

```
TLSCAFile=/home/zabbix/ca.crt
```

```
TLSCertFile=/home/zabbix/server.crt
TLSKeyFile=/home/zabbix/server.key
TLSCipherCert13=TLS_AES_256_GCM_SHA384
TLSCipherCert=EECDH+aRSA+AES256:-SHA1:-SHA384
TLSCipherPSK13=TLS_CHACHA20_POLY1305_SHA256
TLSCipherPSK=kECDHEPSK+AES256:-SHA1
TLSCipherAll13=TLS_AES_256_GCM_SHA384
TLSCipherAll=EECDH+aRSA+AES256:-SHA1:-SHA384
```

Attention:
Bien que seules les suites de chiffrement liées aux certificats seront utilisées, les paramètres `TLSCipherPSK*` sont également définis afin d'éviter leurs valeurs par défaut, qui incluent des chiffrements moins sécurisés pour une meilleure interopérabilité. Les suites de chiffrement PSK ne peuvent pas être complètement désactivées sur le serveur/proxy.

Et dans `zabbix_agentd.conf` :

```
TLSConnect=cert
TLSAccept=cert
TLSCAFile=/home/zabbix/ca.crt
TLSCertFile=/home/zabbix/agent.crt
TLSKeyFile=/home/zabbix/agent.key
TLSCipherCert13=TLS_AES_256_GCM_SHA384
TLSCipherCert=EECDH+aRSA+AES256:-SHA1:-SHA384
```

Dépannage des problèmes de chiffrement Recommandations générales :

- Commencez par comprendre quel composant agit comme client TLS et lequel agit comme serveur TLS dans le cas problématique.
 Le serveur Zabbix, les proxies et les agents, selon leur interaction, peuvent tous fonctionner comme serveurs TLS et clients TLS.
 Par exemple, le serveur Zabbix se connectant à un agent pour une vérification passive agit comme client TLS. L'agent joue le rôle de serveur TLS.
 L'agent Zabbix, demandant une liste de vérifications actives au proxy, agit comme client TLS. Le proxy joue le rôle de serveur TLS.
 Les utilitaires `zabbix_get` et `zabbix_sender` agissent toujours comme clients TLS.
- Zabbix utilise une authentification mutuelle.
 Chaque partie vérifie son pair et peut refuser la connexion.
 Par exemple, le serveur Zabbix se connectant à un agent peut fermer la connexion immédiatement si le certificat de l'agent n'est pas valide. Et inversement - l'agent Zabbix acceptant une connexion du serveur peut fermer la connexion si le serveur n'est pas approuvé par l'agent.
- Examinez les fichiers journaux des deux côtés - côté client TLS et côté serveur TLS.
 Le côté qui refuse la connexion peut consigner une raison précise du refus. L'autre côté signale souvent une erreur plus générale (par exemple, "Connection closed by peer", "connection was non-properly terminated").
- Parfois, une configuration incorrecte du chiffrement entraîne des messages d'erreur déroutants qui n'indiquent en rien la cause réelle.
 Dans les sous-sections ci-dessous, nous essayons de fournir une collection (loin d'être exhaustive) de messages et de causes possibles qui peuvent aider au dépannage.
 Veuillez noter que différents outils cryptographiques (OpenSSL, GnuTLS) produisent souvent des messages d'erreur différents dans les mêmes situations problématiques.
 Parfois, les messages d'erreur dépendent même de la combinaison particulière d'outils cryptographiques des deux côtés.

Dépannage des problèmes de type de connexion ou d'autorisation Le serveur est configuré pour se connecter avec PSK à l'agent, mais l'agent n'accepte que des connexions non chiffrées

Dans le journal du serveur ou du proxy (avec *GnuTLS* 3.3.16)

```
Échec de l'obtention de la valeur depuis l'agent : zbx_tls_connect(): gnutls_handshake() failed: \
    -110 La connexion TLS a été interrompue de manière incorrecte.
```

Dans le journal du serveur ou du proxy (avec *OpenSSL* 1.0.2c)

```
Échec de l'obtention de la valeur depuis l'agent : connexion TCP réussie, impossible d'établir TLS vers [[
    Connexion fermée par le pair. Vérifiez les types de connexion autorisés et les droits d'accès
```

Un côté se connecte avec certificat, mais l'autre côté n'accepte que PSK, ou inversement

Dans n'importe quel journal (avec *GnuTLS*) :

```
failed to accept an incoming connection: from 127.0.0.1: zbx_tls_accept(): gnutls_handshake() failed:\
    -21 Could not negotiate a supported cipher suite.
```

Dans n'importe quel journal (avec *OpenSSL* 1.0.2c) :

```
failed to accept an incoming connection: from 127.0.0.1: TLS handshake returned error code 1:\
    file .\ssl\s3_srvr.c line 1411: error:1408A0C1:SSL routines:ssl3_get_client_hello:no shared cipher:\
    TLS write fatal alert "handshake failure"
```

Tentative d'utilisation de Zabbix sender compilé avec la prise en charge TLS pour envoyer des données à un serveur/proxy Zabbix compilé sans TLS Dans le journal du côté de la connexion :

Linux :

```
...Dans zbx_tls_init_child()
...Bibliothèque OpenSSL (version OpenSSL 1.1.1  11 Sep 2018) initialisée
...
...Dans zbx_tls_connect(): psk_identity:"PSK test sender"
...Fin de zbx_tls_connect():ÉCHEC erreur:'connexion fermée par le pair'
...erreur d'envoi de valeur : TCP réussi, impossible d'établir TLS vers [[localhost]:10051] : connexion fe
```

Windows :

```
...Bibliothèque OpenSSL (version OpenSSL 1.1.1a  20 Nov 2018) initialisée
...
...Dans zbx_tls_connect(): psk_identity:"PSK test sender"
...zbx_psk_client_cb() a demandé l'identité PSK "PSK test sender"
...Fin de zbx_tls_connect():ÉCHEC erreur:'erreur d'E/S SSL_connect() : [0x00000000] L'opération s'est term
...erreur d'envoi de valeur : TCP réussi, impossible d'établir TLS vers [[192.168.1.2]:10051] : erreur d'E
```

Dans le journal du côté récepteur :

```
...échec de l'acceptation d'une connexion entrante : depuis 127.0.0.1 : la prise en charge de TLS n'a pas
```

Un côté se connecte avec PSK, mais l'autre côté utilise LibreSSL ou a été compilé sans prise en charge du chiffrement LibreSSL ne prend pas en charge PSK.

Dans le journal du côté connectant :

```
...TCP successful, cannot establish TLS to [[192.168.1.2]:10050]: SSL_connect() I/O error: [0] Success
```

Dans le journal du côté acceptant :

```
...failed to accept an incoming connection: from 192.168.1.2: support for PSK was not compiled in
```

Dans l'interface Zabbix :

```
Get value from agent failed: TCP successful, cannot establish TLS to [[192.168.1.2]:10050]: SSL_connect()
```

Un côté se connecte avec PSK, mais l'autre côté utilise OpenSSL avec la prise en charge de PSK désactivée Dans le journal du côté connectant :

```
...TCP réussi, impossible d'établir TLS vers [[192.168.1.2]:10050] : SSL_connect() a défini le code de rés
```

Dans le journal du côté acceptant :

```
...échec de l'acceptation d'une connexion entrante : depuis 192.168.1.2 : TLS handshake a défini le code d
```

Chiffrement à l'aide de certificats

Vue d'ensemble Zabbix peut utiliser des certificats RSA au format PEM, signés par une autorité de certification (CA) publique ou interne.

La vérification du certificat est effectuée à l'aide d'un certificat CA préconfiguré. En option, des listes de révocation de certificats (CRL) peuvent être utilisées.

Chaque composant Zabbix ne peut avoir qu'un seul certificat configuré.

Pour plus d'informations sur la mise en place et l'exploitation d'une CA interne, la génération et la signature de demandes de certificat, ainsi que la révocation de certificats, consultez des tutoriels tels que le OpenSSL PKI Tutorial v2.0.

Examinez et testez soigneusement les extensions de votre certificat. Pour plus de détails, voir Limitations on using X.509 v3 certificate extensions.

Paramètres de configuration des certificats Les paramètres de configuration suivants sont pris en charge pour la configuration des certificats sur les composants Zabbix.

Parameter	Mandatory	Description
TLSCAFile	yes	Chemin complet d'un fichier contenant les certificats des CA de niveau supérieur pour la vérification du certificat du pair. Si vous utilisez une chaîne de certificats avec plusieurs membres, ordonnez les certificats en plaçant d'abord les certificats des CA de niveau inférieur, puis ceux des CA de niveau supérieur. Des certificats provenant de plusieurs CA peuvent être inclus dans un seul fichier.
TLSCRLFile	no	Chemin complet d'un fichier contenant les listes de révocation de certificats (CRL).
TLSCertFile	yes	Chemin complet d'un fichier contenant le certificat. Si vous utilisez une chaîne de certificats avec plusieurs membres, ordonnez les certificats en plaçant d'abord le certificat du serveur, du proxy ou de l'agent, puis les certificats des CA de niveau inférieur, et terminez par les certificats des CA de niveau supérieur.
TLSKeyFile	yes	Chemin complet d'un fichier contenant la clé privée. Assurez-vous que ce fichier n'est lisible que par l'utilisateur Zabbix en définissant les droits d'accès appropriés.
TLSServerCertIssuer	no	Émetteur autorisé du certificat du serveur.
TLSServerCertSubject	no	Sujet autorisé du certificat du serveur.

Exemples de configuration Après avoir configuré les certificats nécessaires, configurez les composants Zabbix pour utiliser le chiffrement basé sur des certificats.

Vous trouverez ci-dessous les étapes détaillées pour configurer :

- serveur Zabbix
- proxy Zabbix
- agent Zabbix

Serveur Zabbix

1. Préparez le fichier de certificat CA.

Pour vérifier les certificats des pairs, le serveur Zabbix doit pouvoir accéder au fichier contenant les certificats CA racine autosignés de niveau supérieur. Par exemple, si des certificats provenant de deux CA racine indépendantes sont nécessaires, placez-les dans un fichier situé à `/home/zabbix/zabbix_ca_file.crt` :

```
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 1 (0x1)
    Signature Algorithm: sha1WithRSAEncryption
        Issuer: DC=com, DC=zabbix, O=Zabbix SIA, OU=Development group, CN=Root1 CA
            ...
        Subject: DC=com, DC=zabbix, O=Zabbix SIA, OU=Development group, CN=Root1 CA
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                Public-Key: (2048 bit)
            ...
        X509v3 extensions:
            X509v3 Key Usage: critical
                Certificate Sign, CRL Sign
            X509v3 Basic Constraints: critical
                CA:TRUE
            ...
-----BEGIN CERTIFICATE-----
MIID2jCCAsKgAwIBAgIBATANBgkqhkiG9w0BAQUFADB+MRMwEQYKCZImiZPyLGQB
....
9wEzdN8uTrqoyU78gi12npLj08LegRKjb5hFTVmO
-----END CERTIFICATE-----
Certificate:
    Data:
        Version: 3 (0x2)
```

```
        Serial Number: 1 (0x1)
    Signature Algorithm: sha1WithRSAEncryption
        Issuer: DC=com, DC=zabbix, O=Zabbix SIA, OU=Development group, CN=Root2 CA
            ...
        Subject: DC=com, DC=zabbix, O=Zabbix SIA, OU=Development group, CN=Root2 CA
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                Public-Key: (2048 bit)
            ....
        X509v3 extensions:
            X509v3 Key Usage: critical
                Certificate Sign, CRL Sign
            X509v3 Basic Constraints: critical
                CA:TRUE
            ....
-----BEGIN CERTIFICATE-----
MIID3DCCAsSgAwIBAgIBATANBgkqhkiG9w0BAQUFADB/MRMwEQYKCZImiZPyLGQB
...
vdGNYoSfvu41GQAR5Vj5FnRJRzv5XQOZ3B6894GY1zY=
-----END CERTIFICATE-----
```

2. Placez le certificat/la chaîne de certificats du serveur Zabbix dans un fichier, par exemple à l'emplacement `/home/zabbix/zabbix_serve`
Le premier certificat est le certificat du serveur Zabbix, suivi du certificat de l'autorité de certification intermédiaire :

```
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 1 (0x1)
    Signature Algorithm: sha1WithRSAEncryption
        Issuer: DC=com, DC=zabbix, O=Zabbix SIA, OU=Development group, CN=Signing CA
        ...
        Subject: DC=com, DC=zabbix, O=Zabbix SIA, OU=Development group, CN=Zabbix server
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                Public-Key: (2048 bit)
                ...
        X509v3 extensions:
            X509v3 Key Usage: critical
                Digital Signature, Key Encipherment
            X509v3 Basic Constraints:
                CA:FALSE
            ...
-----BEGIN CERTIFICATE-----
MIIECDCCAvCgAwIBAgIBATANBgkqhkiG9w0BAQUFADCBgTETMBEGCgmSJomT8ixk
...
h02u1GHiy46GI+xfR3LsPwFKlkTaaLaL/6aaoQ==
-----END CERTIFICATE-----
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 2 (0x2)
    Signature Algorithm: sha1WithRSAEncryption
        Issuer: DC=com, DC=zabbix, O=Zabbix SIA, OU=Development group, CN=Root1 CA
        ...
        Subject: DC=com, DC=zabbix, O=Zabbix SIA, OU=Development group, CN=Signing CA
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                Public-Key: (2048 bit)
            ...
        X509v3 extensions:
            X509v3 Key Usage: critical
                Certificate Sign, CRL Sign
            X509v3 Basic Constraints: critical
```

```
                CA:TRUE, pathlen:0
        ...
-----BEGIN CERTIFICATE-----
MIID4TCCAsmgAwIBAgIBAjANBgkqhkiG9w0BAQUFADB+MRMwEQYKCZImiZPyLGQB
...
dyCeWnvL7u5sd6ffo8iRny0QzbHKmQt/wUtcVIvWXdMIFJM0Hw==
-----END CERTIFICATE-----
```

Note:
Utilisez uniquement les attributs mentionnés ci-dessus pour les certificats client et serveur afin d'éviter d'affecter le processus de vérification des certificats. Par exemple, OpenSSL peut échouer à établir une connexion chiffrée si les extensions *X509v3 Subject Alternative Name* ou *Netscape Cert Type* sont utilisées. Pour plus d'informations, consultez Limitations liées à l'utilisation des extensions de certificat X.509 v3.

3. Placez la clé privée du serveur Zabbix dans un fichier, par exemple à l'emplacement `/home/zabbix/zabbix_server.key` :

```
-----BEGIN PRIVATE KEY-----
MIIEwAIBADANBgkqhkiG9w0BAQEFAASCBKowggSmAgEAAoIBAQC9tIXIJoVnNXDl
...
IJLkhbybBYEf47MLhffWa7XvZTY=
-----END PRIVATE KEY-----
```

4. Modifiez les paramètres de configuration TLS dans le fichier de configuration du serveur Zabbix :

```
TLSCAFile=/home/zabbix/zabbix_ca_file.crt
TLSCertFile=/home/zabbix/zabbix_server.crt
TLSKeyFile=/home/zabbix/zabbix_server.key
```

Proxy Zabbix

1. Préparez les fichiers contenant les certificats CA racine, le certificat/la chaîne de certificats du proxy Zabbix et la clé privée, comme décrit dans la section Serveur Zabbix. Modifiez ensuite les paramètres `TLSCAFile`, `TLSCertFile` et `TLSKeyFile` dans le fichier de configuration du proxy Zabbix en conséquence.

2. Modifiez les paramètres TLS supplémentaires dans le fichier de configuration du proxy Zabbix :

- Pour un proxy actif : `TLSConnect=cert`
- Pour un proxy passif : `TLSAccept=cert`

Note:
Pour améliorer la sécurité du proxy, vous pouvez également définir les paramètres `TLSServerCertIssuer` et `TLSServerCertSubject`. Pour plus d'informations, consultez Restreindre l'émetteur et le sujet des certificats autorisés.

Les paramètres TLS du fichier de configuration final du proxy peuvent se présenter comme suit :

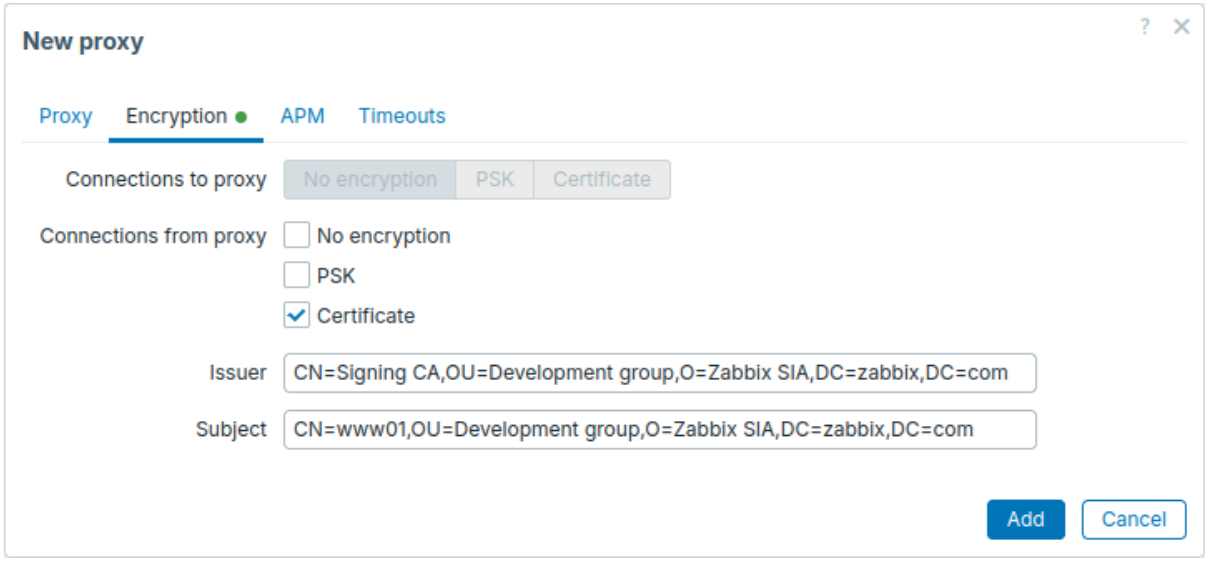
```
TLSConnect=cert
TLSAccept=cert
TLSCAFile=/home/zabbix/zabbix_ca_file.crt
TLSServerCertIssuer=CN=Signing CA,OU=Development group,O=Zabbix SIA,DC=zabbix,DC=com
TLSServerCertSubject=CN=Zabbix server,OU=Development group,O=Zabbix SIA,DC=zabbix,DC=com
TLSCertFile=/home/zabbix/zabbix_proxy.crt
TLSKeyFile=/home/zabbix/zabbix_proxy.key
```

3. Configurez le chiffrement de ce proxy dans l'interface Zabbix :

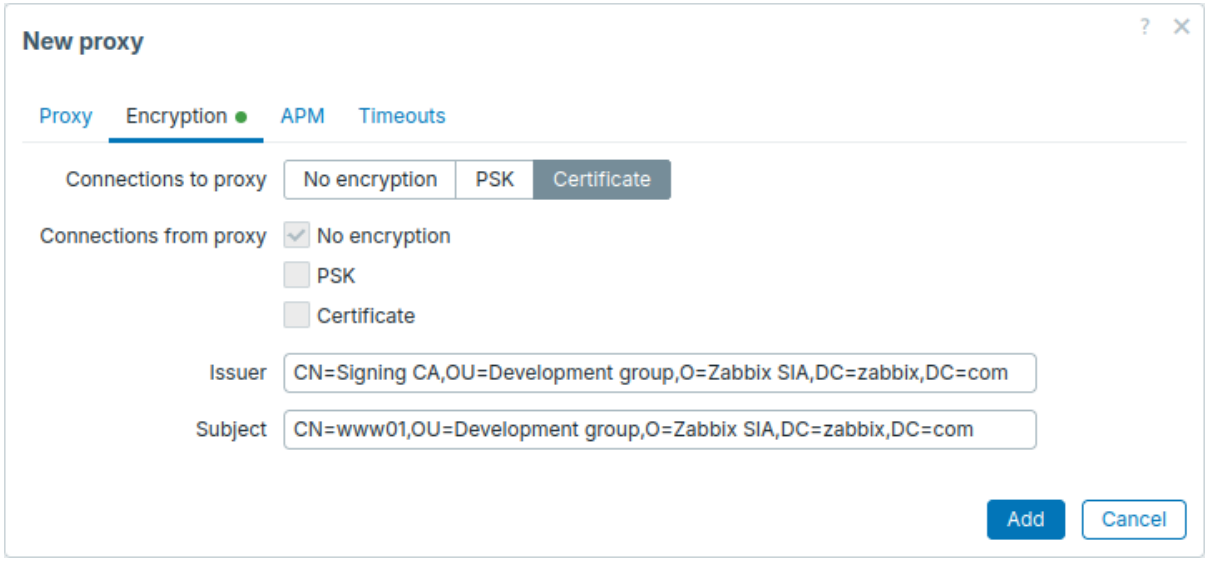
- Accédez à : *Administration* > *Proxies*.
- Sélectionnez le proxy et cliquez sur l'onglet *Chiffrement*.

Dans les exemples ci-dessous, les champs *Émetteur* et *Sujet* sont renseignés. Pour plus d'informations sur les raisons et la manière d'utiliser ces champs, consultez Restreindre l'émetteur et le sujet des certificats autorisés.

Pour un proxy actif :



Pour un proxy passif :



Agent Zabbix

1. Préparez les fichiers contenant les certificats CA racine, le certificat/la chaîne de certificats de l'agent Zabbix et la clé privée, comme décrit dans la section Serveur Zabbix. Modifiez ensuite les paramètres `TLSCAFile`, `TLSCertFile` et `TLSKeyFile` dans le fichier de configuration de l'agent Zabbix en conséquence.

2. Modifiez les paramètres TLS supplémentaires dans le fichier de configuration de l'agent Zabbix :

- Pour un agent actif : `TLSConnect=cert`
- Pour un agent passif : `TLSAccept=cert`

Note:
Pour améliorer la sécurité de l'agent, vous pouvez définir les paramètres `TLSServerCertIssuer` et `TLSServerCertSubject`. Pour plus d'informations, consultez Restriction de l'émetteur et du sujet des certificats autorisés.

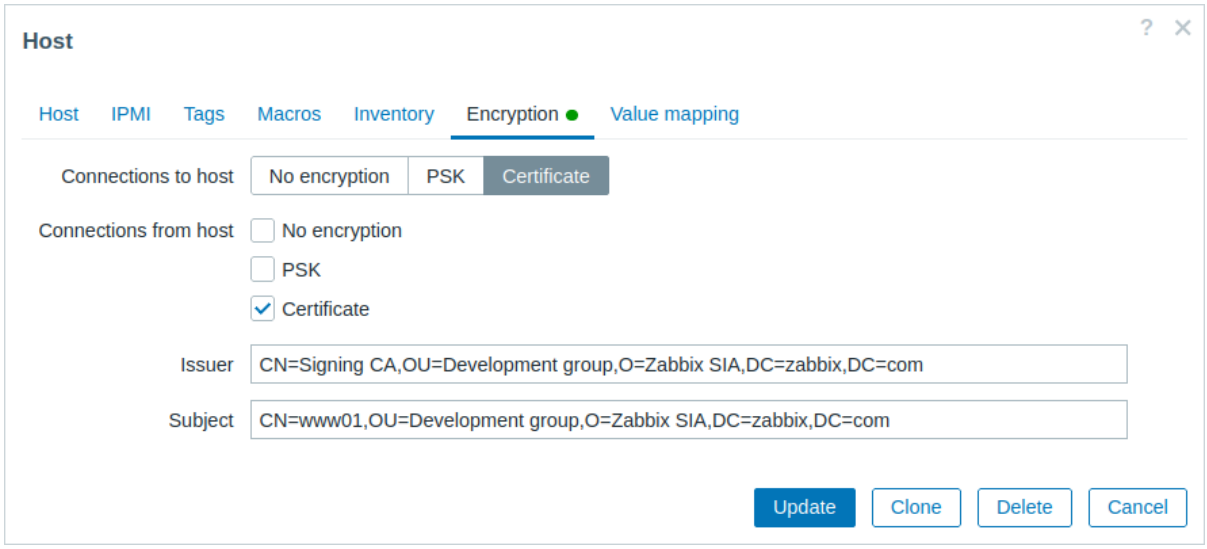
Les paramètres TLS du fichier de configuration final de l'agent peuvent se présenter comme suit. Notez que l'exemple suppose que le hôte est surveillé par un proxy ; celui-ci est donc indiqué comme sujet du certificat :

```
TLSConnect=cert
TLSAccept=cert
TLSCAFile=/home/zabbix/zabbix_ca_file.crt
TLSServerCertIssuer=CN=Signing CA,OU=Development group,O=Zabbix SIA,DC=zabbix,DC=com
TLSServerCertSubject=CN=Zabbix proxy,OU=Development group,O=Zabbix SIA,DC=zabbix,DC=com
TLSCertFile=/home/zabbix/zabbix_agentd.crt
TLSKeyFile=/home/zabbix/zabbix_agentd.key
```

3. Configurez le chiffrement dans l'interface Zabbix pour le hôte surveillé par cet agent.

- Accédez à : *Collecte de données* > *Hôtes*.
- Sélectionnez le hôte et cliquez sur l'onglet *Chiffrement*.

Dans l'exemple ci-dessous, les champs *Émetteur* et *Sujet* sont renseignés. Pour plus d'informations sur les raisons et la manière d'utiliser ces champs, consultez Restriction de l'émetteur et du sujet des certificats autorisés.



Service web Zabbix

1. Préparez les fichiers contenant les certificats CA de niveau supérieur, le certificat/la chaîne de certificats du service web Zabbix et la clé privée, comme décrit dans la section serveur Zabbix. Modifiez ensuite les paramètres `TLSCAFile`, `TLSCertFile` et `TLSKeyFile` dans le fichier de configuration du service web Zabbix en conséquence.

2. Modifiez un paramètre TLS supplémentaire dans le fichier de configuration du service web Zabbix : `TLSAccept=cert`

Les paramètres TLS du fichier de configuration final du service web peuvent se présenter comme suit :

```
TLSAccept=cert
TLSCAFile=/home/zabbix/zabbix_ca_file.crt
TLSCertFile=/home/zabbix/zabbix_web_service.crt
TLSKeyFile=/home/zabbix/zabbix_web_service.key
```

3. Configurez le serveur Zabbix pour qu'il se connecte au service web Zabbix configuré avec TLS en modifiant le paramètre `WebServiceURL` dans le fichier de configuration du serveur Zabbix :

```
WebServiceURL=https://example.com:443/report
```

Restriction de l'émetteur et du sujet des certificats autorisés Lorsque deux composants Zabbix (par exemple, un serveur et un agent) établissent une connexion TLS, ils valident leurs certificats respectifs. Si le certificat d'un pair est signé par une autorité de certification approuvée (avec un certificat de niveau supérieur préconfiguré dans `TLSCAFile`), qu'il est valide, qu'il n'a pas expiré et qu'il satisfait aux autres vérifications, la communication entre les composants peut avoir lieu. Dans ce cas le plus simple, l'émetteur et le sujet du certificat ne sont pas vérifiés.

Cependant, cela présente un risque : toute personne disposant d'un certificat valide peut usurper l'identité de n'importe qui d'autre (par exemple, un certificat d'hôte pourrait être utilisé pour usurper l'identité d'un serveur). Bien que cela puisse être acceptable dans les petits environnements où les certificats sont signés par une autorité de certification interne dédiée et où le risque d'usurpation est faible, cela peut ne pas être suffisant dans les environnements plus vastes ou plus sensibles en matière de sécurité.

Si votre autorité de certification de niveau supérieur émet des certificats qui ne devraient pas être acceptés par Zabbix ou si vous souhaitez réduire le risque d'usurpation, vous pouvez restreindre les certificats autorisés en spécifiant leur émetteur et leur sujet.

Par exemple, dans le fichier de configuration du proxy Zabbix, vous pouvez spécifier :

```
TLSServerCertIssuer=CN=Signing CA,OU=Development group,O=Zabbix SIA,DC=zabbix,DC=com
TLSServerCertSubject=CN=Zabbix server,OU=Development group,O=Zabbix SIA,DC=zabbix,DC=com
```

Avec ces paramètres, un proxy actif ne communiquera pas avec un serveur Zabbix dont le certificat possède un émetteur ou un sujet différent. De même, un proxy passif n'acceptera pas les requêtes provenant d'un tel serveur.

Règles de correspondance des chaînes `Issuer` et `Subject`

Les règles de correspondance des chaînes `Issuer` et `Subject` sont les suivantes :

- Les chaînes `Issuer` et `Subject` sont vérifiées indépendamment. Les deux sont facultatives.
- Une chaîne non spécifiée signifie que n'importe quelle chaîne est acceptée.
- Les chaînes sont comparées *telles quelles* et doivent correspondre exactement.
- Les caractères UTF-8 sont pris en charge. En revanche, les jokers (*) et les expressions régulières ne sont pas pris en charge.
- Les exigences suivantes de RFC 4514 sont implémentées - les caractères qui nécessitent un échappement (avec une barre oblique inverse '\', U+005C) :
  - partout dans la chaîne : '"' (U+0022), '+' (U+002B), ',' (U+002C), ';' (U+003B), '<' (U+003C), '>' (U+003E), '\\' (U+005C) ;
  - au début de la chaîne : espace (' ', U+0020) ou dièse ('#', U+0023) ;
  - à la fin de la chaîne : espace (' ', U+0020).
- Les caractères nuls (U+0000) ne sont pas pris en charge. Si un caractère nul est rencontré, la correspondance échouera.
- Les normes RFC 4517 et RFC 4518 ne sont pas prises en charge.

Par exemple, si les chaînes d'organisation (O) de `Issuer` et `Subject` contiennent des espaces de fin et si la chaîne d'unité organisationnelle (OU) de `Subject` contient des guillemets doubles, ces caractères doivent être échappés :

```
TLSServerCertIssuer=CN=Signing CA,OU=Development head,O=\ Example SIA\ ,DC=example,DC=com
TLSServerCertSubject=CN=Zabbix server,OU=Development group \"5\",O=\ Example SIA\ ,DC=example,DC=com
```

Ordre et format des champs

Zabbix suit les recommandations de RFC 4514, qui spécifie un ordre "inverse" pour ces champs, en commençant par les champs de niveau le plus bas (CN), en poursuivant avec les champs de niveau intermédiaire (OU, O), et en terminant par les champs de niveau le plus élevé (DC).

```
TLSServerCertIssuer=CN=Signing CA,OU=Development group,O=Zabbix SIA,DC=zabbix,DC=com
TLSServerCertSubject=CN=Zabbix proxy,OU=Development group,O=Zabbix SIA,DC=zabbix,DC=com
```

En revanche, OpenSSL affiche par défaut les chaînes `Issuer` et `Subject` dans l'ordre du niveau le plus élevé vers le niveau le plus bas. Dans l'exemple suivant, les champs `Issuer` et `Subject` commencent par le niveau le plus élevé (DC) et se terminent par le champ de niveau le plus bas (CN). Le formatage avec des espaces et des séparateurs de champs varie également selon les options utilisées et ne correspond donc pas au format requis par Zabbix.

```
$ openssl x509 -noout -in /home/zabbix/zabbix_proxy.crt -issuer -subject
issuer= /DC=com/DC=zabbix/O=Zabbix SIA/OU=Development group/CN=Signing CA
subject= /DC=com/DC=zabbix/O=Zabbix SIA/OU=Development group/CN=Zabbix proxy

$ openssl x509 -noout -text -in /home/zabbix/zabbix_proxy.crt
Certificate:
    ...
        Issuer: DC=com, DC=zabbix, O=Zabbix SIA, OU=Development group, CN=Signing CA
        ...
        Subject: DC=com, DC=zabbix, O=Zabbix SIA, OU=Development group, CN=Zabbix proxy
```

Pour formater correctement les chaînes *Issuer* et *Subject* pour Zabbix, invoquez OpenSSL avec les options suivantes :

```
$ openssl x509 -noout -issuer -subject \
    -nameopt esc_2253,esc_ctrl,utf8,dump_nostr,dump_unknown,dump_der,sep_comma_plus,dn_rev,sname\
    -in /home/zabbix/zabbix_proxy.crt
```

La sortie sera alors dans l'ordre inverse, séparée par des virgules, et utilisable dans les fichiers de configuration Zabbix et l'interface :

```
issuer=CN=Signing CA,OU=Development group,O=Zabbix SIA,DC=zabbix,DC=com
subject=CN=Zabbix proxy,OU=Development group,O=Zabbix SIA,DC=zabbix,DC=com
```

Limitations sur l'utilisation des extensions de certificat X.509 v3 Lors de l'implémentation de certificats X.509 v3 dans Zabbix, certaines extensions peuvent ne pas être entièrement prises en charge ou peuvent entraîner un comportement incohérent.

Extension Subject Alternative Name

Zabbix ne prend pas en charge l'extension *Subject Alternative Name*, qui est utilisée pour spécifier des noms DNS alternatifs tels que des adresses IP ou des adresses e-mail. Zabbix peut uniquement valider la valeur du champ *Subject* du certificat (voir Restriction des émetteurs et sujets de certificat autorisés). Si les certificats incluent le champ `subjectAltName`, le résultat de la validation du certificat peut varier selon les bibliothèques cryptographiques spécifiques utilisées pour compiler les composants Zabbix. Par conséquent, Zabbix peut accepter ou rejeter des certificats en fonction de ces combinaisons.

Extension Extended Key Usage

Zabbix prend en charge l'extension *Extended Key Usage*. Cependant, si elle est utilisée, il est généralement nécessaire que les attributs *clientAuth* (pour l'authentification du client TLS WWW) et *serverAuth* (pour l'authentification du serveur TLS WWW) soient tous deux spécifiés. Par exemple :

- Dans les vérifications passives, où l'agent Zabbix fonctionne comme un serveur TLS, l'attribut *serverAuth* doit être inclus dans le certificat de l'agent.
- Pour les vérifications actives, où l'agent fonctionne comme un client TLS, l'attribut *clientAuth* doit être inclus dans le certificat de l'agent.

Bien que GnuTLS puisse émettre un avertissement en cas de violation de l'utilisation des clés, il autorise généralement la communication à se poursuivre malgré ces avertissements.

Extension Name Constraints

La prise en charge de l'extension *Name Constraints* varie selon les bibliothèques cryptographiques. Assurez-vous que la bibliothèque choisie prend en charge cette extension. Cette extension peut empêcher Zabbix de charger des certificats CA si cette section est marquée comme critique, selon la bibliothèque utilisée.

Listes de révocation de certificats (CRL) Si un certificat est compromis, l'autorité de certification (CA) peut le révoquer en l'incluant dans une liste de révocation de certificats (CRL). Les CRL sont gérées via des fichiers de configuration et peuvent être spécifiées à l'aide du paramètre `TLSCRLFile` dans les fichiers de configuration du serveur, du proxy et de l'agent. Par exemple :

```
TLSCRLFile=/home/zabbix/zabbix_crl_file.crt
```

Dans ce cas, `zabbix_crl_file.crt` peut contenir des CRL provenant de plusieurs CA, et pourrait ressembler à ceci :

```
-----BEGIN X509 CRL-----
MIIB/DCB5QIBATANBgkqhkiG9w0BAQUFADCBgTETMBEGCgmSJomT8ixkARkWA2Nv
...
treZeUPjb7LSmZ3K2hpbZN7SoOZcAoHQ3GWd9npuctg=
-----END X509 CRL-----
-----BEGIN X509 CRL-----
MIIB+TCB4gIBATANBgkqhkiG9w0BAQUFADB/MRMwEQYKCZImiZPyLGQBGRYDY29t
...
CAEebS2CND3ShBedZ8YSil59O6JvaDP61lR5lNs=
-----END X509 CRL-----
```

Le fichier CRL n'est chargé qu'au démarrage de Zabbix. Pour mettre à jour la CRL, redémarrez Zabbix.

> **Attention:**
> Si les composants Zabbix sont compilés avec OpenSSL et que des CRL sont utilisées, assurez-vous que chaque CA de niveau supérieur et intermédiaire dans les chaînes de certificats dispose d'une CRL correspondante (même vide) incluse dans `TLSCRLFile`.

Résolution des problèmes de certificat OpenSSL utilisé avec des CRL et, pour certains CA dans la chaîne de certificats, leur CRL n'est pas incluse dans `TLSCRLFile`

Dans le journal du serveur TLS en cas de pair *OpenSSL* :

```
failed to accept an incoming connection: from 127.0.0.1: TLS handshake with 127.0.0.1 returned error code
    file s3_srvr.c line 3251: error:14089086: SSL routines:ssl3_get_client_certificate:certificate verify
    TLS write fatal alert "unknown CA"
```

Dans le journal du serveur TLS en cas de pair *GnuTLS* :

```
failed to accept an incoming connection: from 127.0.0.1: TLS handshake with 127.0.0.1 returned error code
    file rsa_pk1.c line 103: error:0407006A: rsa routines:RSA_padding_check_PKCS1_type_1:\
    block type is not 01 file rsa_eay.c line 705: error:04067072: rsa routines:RSA_EAY_PUBLIC_DECRYPT:padd
```

CRL expirée ou expirant pendant le fonctionnement du serveur

OpenSSL, dans le journal du serveur :

- avant l'expiration :

```
cannot connect to proxy "proxy-openssl-1.0.1e": TCP successful, cannot establish TLS to [[127.0.0.1]:20004
    SSL_connect() returned SSL_ERROR_SSL: file s3_clnt.c line 1253: error:14090086:\
    SSL routines:ssl3_get_server_certificate:certificate verify failed:\
```

```
    TLS write fatal alert "certificate revoked"
```

- après l'expiration :

```
cannot connect to proxy "proxy-openssl-1.0.1e": TCP successful, cannot establish TLS to [[127.0.0.1]:20004
    SSL_connect() returned SSL_ERROR_SSL: file s3_clnt.c line 1253: error:14090086:\
    SSL routines:ssl3_get_server_certificate:certificate verify failed:\
    TLS write fatal alert "certificate expired"
```

Le point ici est qu'avec une CRL valide, un certificat révoqué est signalé comme "certificate revoked". Lorsque la CRL expire, le message d'erreur change en "certificate expired", ce qui est assez trompeur.

GnuTLS, dans le journal du serveur :

- avant et après l'expiration, identique :

```
cannot connect to proxy "proxy-openssl-1.0.1e": TCP successful, cannot establish TLS to [[127.0.0.1]:20004
      invalid peer certificate: The certificate is NOT trusted. The certificate chain is revoked.
```

Certificat auto-signé, CA inconnue

OpenSSL, dans le journal :

```
error:'self signed certificate: SSL_connect() set result code to SSL_ERROR_SSL: file ../ssl/statem/statem_
      line 1924: error:1416F086:SSL routines:tls_process_server_certificate:certificate verify failed:\
      TLS write fatal alert "unknown CA"'
```

Ce problème a été observé lorsque le certificat du serveur avait par erreur la même chaîne Issuer et Subject, bien qu'il ait été signé par une CA. Issuer et Subject sont égaux dans le certificat CA racine, mais ils ne peuvent pas être égaux dans le certificat du serveur. (La même règle s'applique aux certificats du proxy et de l'agent.)

Pour vérifier si un certificat contient les mêmes entrées Issuer et Subject, exécutez :

```
openssl x509 -in <yourcertificate.crt> -noout -text
```

Il est acceptable que le certificat racine (de niveau supérieur) ait des valeurs identiques pour Issuer et Subject.

Chiffrement à l'aide de clés prépartagées

Vue d'ensemble Chaque clé prépartagée (PSK) dans Zabbix est en réalité composée de deux éléments :

- une chaîne d'identité PSK non secrète ;
- une valeur de chaîne PSK secrète.

La chaîne d'identité PSK est une chaîne UTF-8 non vide. Par exemple : « PSK ID 001 Zabbix agentd ». Il s'agit d'un nom unique utilisé par les composants Zabbix pour désigner cette PSK spécifique. Ne placez pas d'informations sensibles dans la chaîne d'identité PSK : elle est transmise sur le réseau sans chiffrement.

La valeur PSK est une chaîne de chiffres hexadécimaux difficile à deviner, par exemple : `e560cb0d918d26d31b4f642181f5f570ad89a3909`

> **Warning:**
> N'utilisez jamais en production les valeurs PSK fournies à titre d'exemple dans la documentation. Générez plutôt une PSK unique et sécurisée sur le plan cryptographique.

Limites de taille Il existe des limites de taille pour l'identité et la valeur PSK dans Zabbix. Dans certains cas, une bibliothèque cryptographique peut avoir une limite inférieure :

Composant	Taille maximale de l'identité PSK	Taille minimale de la valeur PSK	Taille maximale de la valeur PSK
Zabbix	128 caractères UTF-8	128 bits (PSK de 16 octets, saisie sous forme de 32 chiffres hexadécimaux)	2048 bits (PSK de 256 octets, saisie sous forme de 512 chiffres hexadécimaux)
GnuTLS	128 octets (peut inclure des caractères UTF-8)	-	2048 bits (PSK de 256 octets, saisie sous forme de 512 chiffres hexadécimaux)
OpenSSL 1.0.x, 1.1.0	127 octets (peut inclure des caractères UTF-8)	-	2048 bits (PSK de 256 octets, saisie sous forme de 512 chiffres hexadécimaux)

Composant	Taille maximale de l'identité PSK	Taille minimale de la valeur PSK	Taille maximale de la valeur PSK
OpenSSL 1.1.1	127 octets (peut inclure des caractères UTF-8)	-	512 bits (PSK de 64 octets, saisie sous forme de 128 chiffres hexadécimaux)
OpenSSL 1.1.1a et versions ultérieures	127 octets (peut inclure des caractères UTF-8)	-	2048 bits (PSK de 256 octets, saisie sous forme de 512 chiffres hexadécimaux)

Attention:
L'interface Zabbix permet de configurer une chaîne d'identité PSK d'une longueur maximale de 128 caractères et une PSK de 2048 bits, quelles que soient les bibliothèques cryptographiques utilisées.
 Si certains composants Zabbix prennent en charge des limites inférieures, il incombe à l'utilisateur de configurer l'identité et la valeur PSK avec une longueur autorisée pour ces composants.
 Le dépassement des limites de longueur entraîne des échecs de communication entre les composants Zabbix.

Avant que le serveur Zabbix ne se connecte à l'agent à l'aide d'une PSK, le serveur recherche dans la base de données l'identité et la valeur PSK configurées pour cet agent (en réalité, dans le cache de configuration). À la réception d'une connexion, l'agent utilise l'identité et la valeur PSK définies dans son fichier de configuration. Si les deux parties disposent de la même chaîne d'identité PSK et de la même valeur PSK, la connexion peut aboutir.

Attention:
Chaque identité PSK doit être associée à une seule valeur. Il incombe à l'utilisateur de s'assurer qu'il n'existe pas deux PSK ayant la même chaîne d'identité, mais des valeurs différentes. Le non-respect de cette règle peut entraîner des erreurs imprévisibles ou des interruptions de communication entre les composants Zabbix utilisant des PSK avec cette chaîne d'identité PSK.

Génération d'une PSK Par exemple, une PSK de 256 bits (32 octets) peut être générée à l'aide des commandes suivantes :

- avec *OpenSSL* :

```
$ openssl rand -hex 32
af8ced32dfe8714e548694e2d29e1a14ba6fa13f216cb35c19d0feb1084b0429
```

- avec *GnuTLS* :

```
$ psktool -u psk_identity -p database.psk -s 32
Generating a random key for user 'psk_identity'
Key stored to database.psk

$ cat database.psk
psk_identity:9b8eafedfaae00cece62e85d5f4792c7d9c9bcc851b23216a1d300311cc4f7cb
```

Notez que « psktool » ci-dessus génère un fichier de base de données contenant une identité PSK et la PSK associée. Zabbix attend uniquement une PSK dans le fichier PSK ; la chaîne d'identité et les deux-points (':') doivent donc être supprimés du fichier.

Configuration du PSK pour la communication serveur-agent (exemple) Sur l'hôte agent, écrivez la valeur PSK générée précédemment dans un fichier, par exemple `/home/zabbix/zabbix_agentd.psk`. Le fichier doit contenir la valeur PSK sur la première ligne.

Définissez les droits d'accès au fichier PSK : il doit être lisible uniquement par l'utilisateur Zabbix.

Modifiez les paramètres TLS dans le fichier de configuration de l'agent Zabbix (`zabbix_agentd.conf`). Par exemple, définissez :

```
TLSConnect=psk
TLSAccept=psk
TLSPSKFile=/home/zabbix/zabbix_agentd.psk
TLSPSKIdentity=PSK 001
```

L'agent se connectera au serveur (vérifications actives) et n'acceptera du serveur et de `zabbix_get` que les connexions utilisant le PSK. L'identité PSK sera « PSK 001 ».

Redémarrez l'agent. Vous pouvez maintenant tester la connexion à l'aide de `zabbix_get`, par exemple :

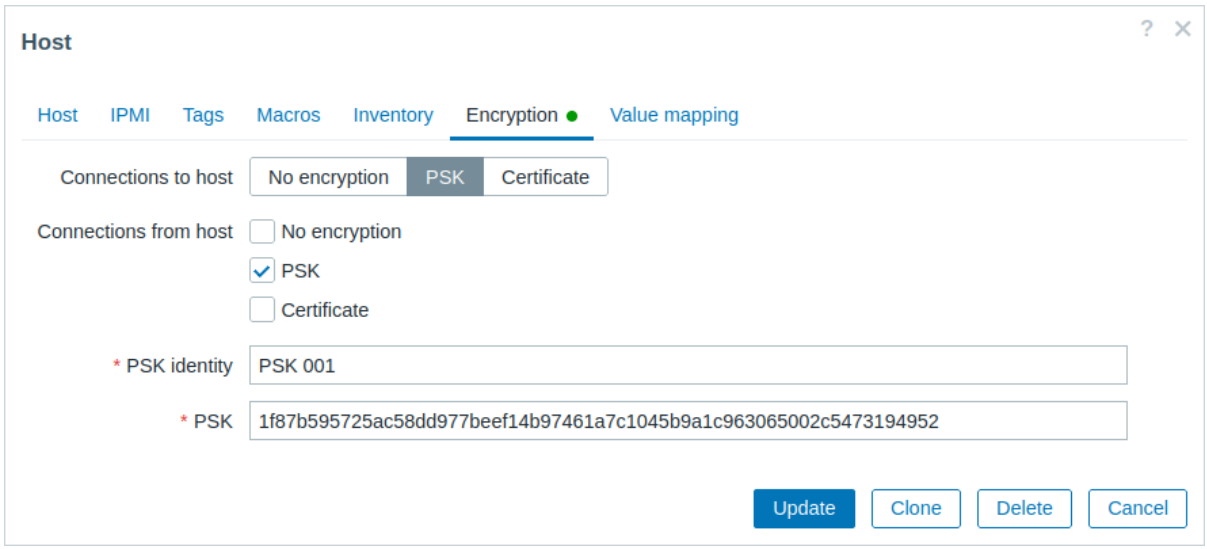
```
zabbix_get -s 127.0.0.1 -k "system.cpu.load[all,avg1]" --tls-connect=psk --tls-psk-identity="PSK 001" --tl
```

(Pour réduire les temps d'arrêt, consultez la procédure de modification du type de connexion dans la gestion du chiffrement des connexions.)

Configurez le chiffrement PSK pour cet agent dans l'interface Zabbix :

1. Accédez à : *Collecte de données* > *Hôtes*.
2. Sélectionnez l'hôte et cliquez sur l'onglet *Chiffrement*.
3. Configurez les détails du chiffrement PSK.

Exemple :



Tous les champs de saisie obligatoires sont marqués d'un astérisque rouge.

Lorsque le cache de configuration est synchronisé avec la base de données, les nouvelles connexions utiliseront le PSK. Vérifiez les fichiers journaux du serveur et de l'agent pour détecter d'éventuels messages d'erreur.

Configuration du PSK pour la communication entre le serveur et le proxy actif (exemple) Sur le proxy, écrivez la valeur du PSK dans un fichier, par exemple `/home/zabbix/zabbix_proxy.psk`. Le fichier doit contenir le PSK dans la première chaîne de texte, par exemple :

```
e560cb0d918d26d31b4f642181f5f570ad89a390931102e5391d08327ba434e9
```

Définissez les droits d'accès au fichier PSK : il doit être lisible uniquement par l'utilisateur Zabbix.

Modifiez les paramètres TLS dans le fichier de configuration du proxy Zabbix (`zabbix_proxy.conf`). Par exemple, définissez :

```
TLSConnect=psk
TLSPSKFile=/home/zabbix/zabbix_proxy.psk
TLSPSKIdentity=PSK 002
```

Le proxy se connectera au serveur à l'aide du PSK. L'identité PSK sera « PSK 002 ».

(Pour réduire les temps d'arrêt, consultez la procédure de modification du type de connexion dans Gestion du chiffrement des connexions.)

Configurez le PSK pour ce proxy dans l'interface Zabbix. Accédez à *Administration* > *Proxies*, sélectionnez le proxy, puis ouvrez l'onglet *Chiffrement*. Dans « Connexions depuis le proxy », cochez PSK. Saisissez « PSK 002 » dans le champ « Identité PSK » et « e560cb0d918d26d31b4f642181f5f570ad89a390931102e5391d08327ba434e9 » dans le champ « PSK ». Cliquez sur « Mettre à jour ».

Redémarrez le proxy. Il commencera à utiliser des connexions chiffrées basées sur le PSK vers le serveur. Vérifiez les fichiers journaux du serveur et du proxy pour détecter d'éventuels messages d'erreur.

Pour un proxy passif, la procédure est très similaire. La seule différence consiste à définir `TLSAccept=psk` dans le fichier de configuration du proxy Zabbix et à définir *Connexions vers le proxy* sur PSK dans l'interface Zabbix.

Résolution des problèmes PSK Le PSK contient un nombre impair de chiffres hexadécimaux

Le proxy ou l'agent ne démarre pas, message dans le journal du proxy ou de l'agent :

```
PSK invalide dans le fichier "/home/zabbix/zabbix_proxy.psk"
```

Chaîne d'identité PSK de plus de 128 octets transmise à GnuTLS

Dans le journal du côté client TLS :

```
gnutls_handshake() failed: -110 The TLS connection was non-properly terminated.
```

Dans le journal du côté serveur TLS.

```
gnutls_handshake() failed: -90 The SRP username supplied is illegal.
```

Valeur PSK trop longue utilisée avec OpenSSL 1.1.1

Dans le journal du côté de la connexion :

```
...OpenSSL library (version OpenSSL 1.1.1  11 Sep 2018) initialized
...
...In zbx_tls_connect(): psk_identity:"PSK 1"
...zbx_psk_client_cb() requested PSK identity "PSK 1"
...End of zbx_tls_connect():FAIL error:'SSL_connect() set result code to SSL_ERROR_SSL: file ssl\statem\ex
```

Dans le journal du côté de l'acceptation :

```
...Message from 123.123.123.123 is missing header. Message ignored.
```

Ce problème survient généralement lors de la mise à niveau d'OpenSSL de 1.0.x ou 1.1.0 vers 1.1.1, et si la valeur PSK est supérieure à 512 bits (PSK de 64 octets, saisi sous forme de 128 chiffres hexadécimaux).

Voir aussi : Limites de taille des valeurs

Connexion sécurisée à la base de données

Aperçu

Cette section fournit des étapes de configuration de Zabbix et des exemples de configuration pour des connexions TLS sécurisées entre :

Database	Zabbix components
MySQL	interface Zabbix, serveur Zabbix, proxy Zabbix
PostgreSQL	interface Zabbix, serveur Zabbix, proxy Zabbix

Pour configurer le chiffrement des connexions au sein du SGBD, consultez la documentation officielle du fournisseur pour plus de détails :

- MySQL : serveurs de base de données de réplication source et réplique.
- MySQL : réplication de groupe, etc. serveurs de base de données.
- PostgreSQL options de chiffrement.

Tous les exemples sont basés sur les versions GA de MySQL CE (8.0) et PgSQL (13) disponibles via les dépôts officiels sous CentOS 8.

Exigences

Les éléments suivants sont requis pour configurer le chiffrement :

- Système d'exploitation pris en charge par le développeur avec OpenSSL >=1.1.X ou une alternative.

> **Note:**
> Il est recommandé d'éviter les systèmes d'exploitation en fin de vie, en particulier dans le cas de nouvelles installations

- Moteur de base de données (RDBMS) installé et maintenu à partir du dépôt officiel fourni par le développeur. Les systèmes d'exploitation sont souvent livrés avec des versions obsolètes de logiciels de base de données pour lesquelles la prise en charge du chiffrement n'est pas implémentée, par exemple les systèmes basés sur RHEL 7 et PostgreSQL 9.2, MariaDB 5.5 sans prise en charge du chiffrement.

Terminologie

La définition de cette option impose l'utilisation d'une connexion TLS à la base de données depuis le serveur/proxy Zabbix et l'interface vers la base de données :

- `required` - se connecter en utilisant TLS comme mode de transport sans vérification d'identité
- `verify_ca` - se connecter en utilisant TLS et vérifier le certificat
- `verify_full` - se connecter en utilisant TLS, vérifier le certificat et vérifier que l'identité de la base de données (CN) spécifiée par DBHost correspond à son certificat

Configuration Zabbix

Interface vers la base de données

Une connexion sécurisée à la base de données peut être configurée lors de l'installation de l'interface :

- Cochez la case *Chiffrement TLS de la base de données* à l'étape Configurer la connexion à la base de données pour activer le chiffrement du transport.
- Cochez la case *Vérifier le certificat de la base de données*, qui apparaît lorsque le champ *Chiffrement TLS* est coché, pour activer le chiffrement avec des certificats.

Note:
Pour MySQL, la case *Chiffrement TLS de la base de données* est désactivée si *Hôte de la base de données* est défini sur localhost, car une connexion utilisant un fichier socket (sous Unix) ou la mémoire partagée (sous Windows) ne peut pas être chiffrée.
 Pour PostgreSQL, la case *Chiffrement TLS* est désactivée si la valeur du champ *Hôte de la base de données* commence par une barre oblique ou si le champ est vide.

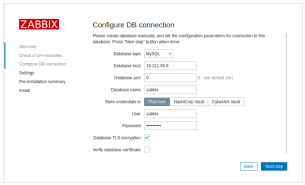
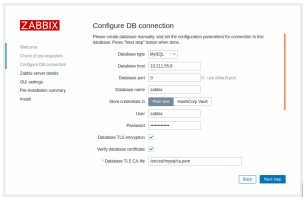
Les paramètres suivants deviennent disponibles en mode de chiffrement TLS avec certificats (si les deux cases sont cochées) :

Paramètre	Description
Fichier CA TLS de la base de données	Indiquez le chemin complet vers un fichier valide d'autorité de certification (CA) TLS.
Fichier de clé TLS de la base de données	Indiquez le chemin complet vers un fichier de clé TLS valide.
Fichier de certificat TLS de la base de données	Indiquez le chemin complet vers un fichier de certificat TLS valide.
Vérification de l'hôte de la base de données	Cochez cette case pour activer la vérification de l'hôte. Désactivé pour MySQL, car la bibliothèque PHP MySQL ne permet pas d'ignorer l'étape de validation du certificat homologue.
Liste de chiffrements TLS de la base de données	Indiquez une liste personnalisée de chiffrements valides. Le format de la liste de chiffrements doit être conforme à la norme OpenSSL. Disponible uniquement pour MySQL.

Attention:
Les paramètres TLS doivent pointer vers des fichiers valides. S'ils pointent vers des fichiers inexistants ou non valides, cela entraînera une erreur d'autorisation.
 Si les fichiers de certificats sont accessibles en écriture, l'interface génère un avertissement dans le rapport Informations système indiquant que « Les fichiers de certificats TLS doivent être en lecture seule. » (affiché uniquement si l'utilisateur PHP est propriétaire du certificat).
Les certificats protégés par des mots de passe ne sont pas pris en charge.

Cas d'utilisation

L'interface Zabbix utilise une interface graphique pour définir les options possibles : `required`, `verify_ca`, `verify_full`. Spécifiez les options requises à l'étape *Configurer les connexions à la base de données* de l'assistant d'installation. Ces options sont mappées dans le fichier de configuration de l'interface (`zabbix.conf.php`) de la manière suivante :

Paramètres de l'interface graphique	Fichier de configuration	Description	Résultat
	... // Used for TLS connection. $DB['ENCRYPTION'] = true; $DB['KEY_FILE'] = ''; $DB['CERT_FILE'] = ''; $DB['CA_FILE'] = ''; $DB['VERIFY_HOST'] = false; $DB['CIPHER_LIST'] = ''; ...	1. Cochez *Chiffrement TLS de la base de données* 2. Laissez *Vérifier le certificat de la base de données* décoché	Active le mode `required`.
	... $DB['ENCRYPTION'] = true; $DB['KEY_FILE'] = ''; $DB['CERT_FILE'] = ''; $DB['CA_FILE'] = '/etc/ssl/mysql/ca.pem'; $DB['VERIFY_HOST'] = false; $DB['CIPHER_LIST'] = ''; ...	1. Cochez *Chiffrement TLS de la base de données* et *Vérifier le certificat de la base de données* 2. Spécifiez le chemin d'accès au *fichier CA TLS de la base de données*	Active le mode `verify_ca`.

Paramètres de l'interface graphique	Fichier de configuration	Description	Résultat

	...
	// Used for TLS connection with strictly defined Cipher list.
	`$DB['ENCRYPTION'] = true;`
	`$DB['KEY_FILE'] = '<key_file_path>';`
	`$DB['CERT_FILE'] = '<key_file_path>';`
	`$DB['CA_FILE'] = '<key_file_path>';`
	`$DB['VERIFY_HOST'] = true;`
	`$DB['CIPHER_LIST'] = '<cipher_list>';`
	...
	
	Ou :
	
	...
	// Used for TLS connection without Cipher list defined - selected by MySQL server
	`$DB['ENCRYPTION'] = true;`
	`$DB['KEY_FILE'] = '<key_file_path>';`
	`$DB['CERT_FILE'] = '<key_file_path>';`
	`$DB['CA_FILE'] = '<key_file_path>';`
	`$DB['VERIFY_HOST'] = true;`
	`$DB['CIPHER_LIST'] = '';`
	...
		1. Cochez *Chiffrement TLS de la base de données* et *Vérifier le certificat de la base de données*
		2. Spécifiez le chemin d'accès au *fichier de clé TLS de la base de données*
		3. Spécifiez le chemin d'accès au *fichier CA TLS de la base de données*
		4. Spécifiez le chemin d'accès au *fichier de certificat TLS de la base de données*
		5. Spécifiez la *liste de chiffrements TLS de la base de données* (facultatif)
			Active le mode `verify_full` pour MySQL.

Paramètres de l'interface graphique	Fichier de configuration	Description	Résultat

	...
	`$DB['ENCRYPTION'] = true;`
	`$DB['KEY_FILE'] = '<key_file_path>';`
	`$DB['CERT_FILE'] = '<key_file_path>';`
	`$DB['CA_FILE'] = '<key_file_path>';`
	`$DB['VERIFY_HOST'] = true;`
	`$DB['CIPHER_LIST'] = ' ';`
	...
		1. Cochez *Chiffrement TLS de la base de données* et *Vérifier le certificat de la base de données*
		2. Spécifiez le chemin d'accès au *fichier de clé TLS de la base de données*
		3. Spécifiez le chemin d'accès au *fichier CA TLS de la base de données*
		4. Spécifiez le chemin d'accès au *fichier de certificat TLS de la base de données*
		5. Cochez *Vérification de l'hôte de la base de données*
			Active le mode `verify_full` pour PostgreSQL.

Voir aussi : Exemples de configuration du chiffrement pour MySQL, Exemples de configuration du chiffrement pour PostgreSQL.

Configuration du serveur/proxy Zabbix

Les connexions sécurisées à la base de données peuvent être configurées à l'aide des paramètres correspondants dans le fichier de configuration du serveur et/ou du proxy Zabbix.

Configuration	Résultat
Aucune	Connexion à la base de données sans chiffrement.
Définir `DBTLSConnect=required`	Le serveur/proxy établit une connexion TLS à la base de données. Une connexion non chiffrée n'est pas autorisée.
1. Définir `DBTLSConnect=verify_ca` 2. Définir `DBTLSCAFile` - spécifier le fichier de l'autorité de certification TLS	Le serveur/proxy établit une connexion TLS à la base de données après avoir vérifié le certificat de la base de données.
1. Définir `DBTLSConnect=verify_full` 2. Définir `DBTLSCAFile` - spécifier le fichier de l'autorité de certification TLS	Le serveur/proxy établit une connexion TLS à la base de données après avoir vérifié le certificat de la base de données et l'identité de l'hôte de la base de données.
1. Définir `DBTLSCAFile` - spécifier le fichier de l'autorité de certification TLS 2. Définir `DBTLSCertFile` - spécifier le fichier de certificat de clé publique du client 3. Définir `DBTLSKeyFile` - spécifier le fichier de clé privée du client	Le serveur/proxy fournit un certificat client lors de la connexion à la base de données.
Définir `DBTLSCipher` - la liste des chiffrements de chiffrement que le client autorise pour les connexions utilisant des protocoles TLS jusqu'à TLS 1.2 ou `DBTLSCipher13` - la liste des chiffrements de chiffrement que le client autorise pour les connexions utilisant le protocole TLS 1.3	(MySQL) La connexion TLS est établie à l'aide d'un chiffrement de la liste fournie. (PostgreSQL) La définition de cette option sera considérée comme une erreur.

Configuration du chiffrement MySQL

Vue d'ensemble

Cette section fournit plusieurs exemples de configuration du chiffrement pour CentOS 8.2 et MySQL 8.4.0 et peut être utilisée comme guide de démarrage rapide pour chiffrer la connexion à la base de données.

Attention:
Si l'hôte MySQL est défini sur localhost, les options de chiffrement ne seront pas disponibles. Dans ce cas, une connexion entre l'interface Zabbix et la base de données utilise un fichier socket (sur Unix) ou la mémoire partagée (sur Windows) et ne peut pas être chiffrée.

Note:
La liste des combinaisons de chiffrement ne se limite pas à celles indiquées sur cette page. Il existe de nombreuses autres combinaisons possibles.

Prérequis

Installez la base de données MySQL depuis le dépôt officiel.

Consultez la documentation MySQL pour plus de détails sur l'utilisation du dépôt MySQL.

Le serveur MySQL est prêt à accepter des connexions sécurisées à l'aide d'un certificat auto-signé.

Pour voir quels utilisateurs utilisent une connexion chiffrée, exécutez la requête suivante (Performance Schema doit être activé) :

```
SELECT sbt.variable_value AS tls_version, t2.variable_value AS cipher, processlist_user AS user, processli
FROM performance_schema.status_by_thread  AS sbt
JOIN performance_schema.threads AS t ON t.thread_id = sbt.thread_id
JOIN performance_schema.status_by_thread AS t2 ON t2.thread_id = t.thread_id
WHERE sbt.variable_name = 'Ssl_version' and t2.variable_name = 'Ssl_cipher'
ORDER BY tls_version;
```

Chiffrement du transport uniquement

Configuration de MySQL

Les versions modernes de la base de données sont prêtes à l'emploi pour le mode de chiffrement `required`. Un certificat côté serveur sera créé après la configuration initiale et le démarrage.

Créez des utilisateurs et des rôles pour les principaux composants (remplacez `<strong_password>` par un mot de passe fort et unique) :

```
mysql> CREATE USER
 'zbx_srv'@'%' IDENTIFIED WITH caching_sha2_password BY '<strong_password>',
 'zbx_web'@'%' IDENTIFIED WITH caching_sha2_password BY '<strong_password>'
 REQUIRE SSL
 PASSWORD HISTORY 5;

mysql> CREATE ROLE 'zbx_srv_role', 'zbx_web_role';

mysql> GRANT SELECT, UPDATE, DELETE, INSERT, CREATE, DROP, ALTER, INDEX, REFERENCES ON zabbix.* TO 'zbx_sr
mysql> GRANT SELECT, UPDATE, DELETE, INSERT ON zabbix.* TO 'zbx_web_role';

mysql> GRANT 'zbx_srv_role' TO 'zbx_srv'@'%';
mysql> GRANT 'zbx_web_role' TO 'zbx_web'@'%';

mysql> SET DEFAULT ROLE 'zbx_srv_role' TO 'zbx_srv'@'%';
mysql> SET DEFAULT ROLE 'zbx_web_role' TO 'zbx_web'@'%';
```

Notez que le protocole X.509 n'est pas utilisé pour vérifier l'identité, mais que l'utilisateur est configuré pour utiliser uniquement des connexions chiffrées. Consultez la documentation MySQL pour plus d'informations sur la configuration des utilisateurs.

Exécutez la commande suivante pour vérifier la connexion (une connexion par socket ne peut pas être utilisée pour tester les connexions sécurisées) :

```
mysql -u zbx_srv -p -h 10.211.55.9 --ssl-mode=REQUIRED
```

Vérifiez l'état actuel et les suites de chiffrement disponibles :

```
mysql> status
--------------
mysql Ver 8.4.0 for Linux on x86_64 (MySQL Community Server - GPL)

Connection id: 62
```

```
Current database:
Current user: zbx_srv@bfdb.local
SSL: Cipher in use is TLS_AES_256_GCM_SHA384

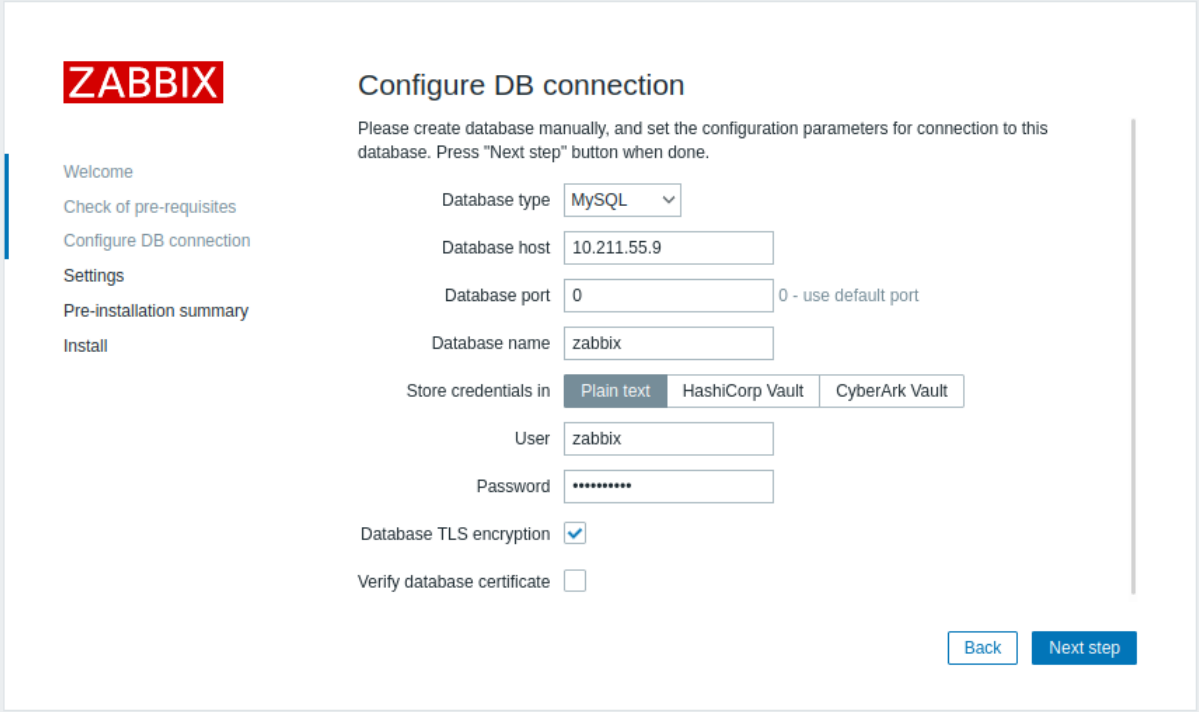

mysql> SHOW SESSION STATUS LIKE 'Ssl_cipher_list'\G;
*************************** 1. row ***************************
Variable_name: Ssl_cipher_list
Value: TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256:TLS_AES_128_CCM_SHA256:E
1 row in set (0.00 sec)

ERROR:
No query specified
```

Interface

Pour activer le chiffrement du transport uniquement pour les connexions entre l'interface Zabbix et la base de données :

- Cochez *Chiffrement TLS de la base de données*
- Laissez *Vérifier le certificat de la base de données* décoché



serveur

Pour activer le chiffrement du transport uniquement pour les connexions entre le serveur et la base de données, modifiez le fichier de configuration du serveur :

```
...
DBHost=10.211.55.9
DBName=zabbix
DBUser=zbx_srv
DBPassword=<strong_password>
DBTLSConnect=required
...
```

Chiffrement avec vérification de l'autorité de certification

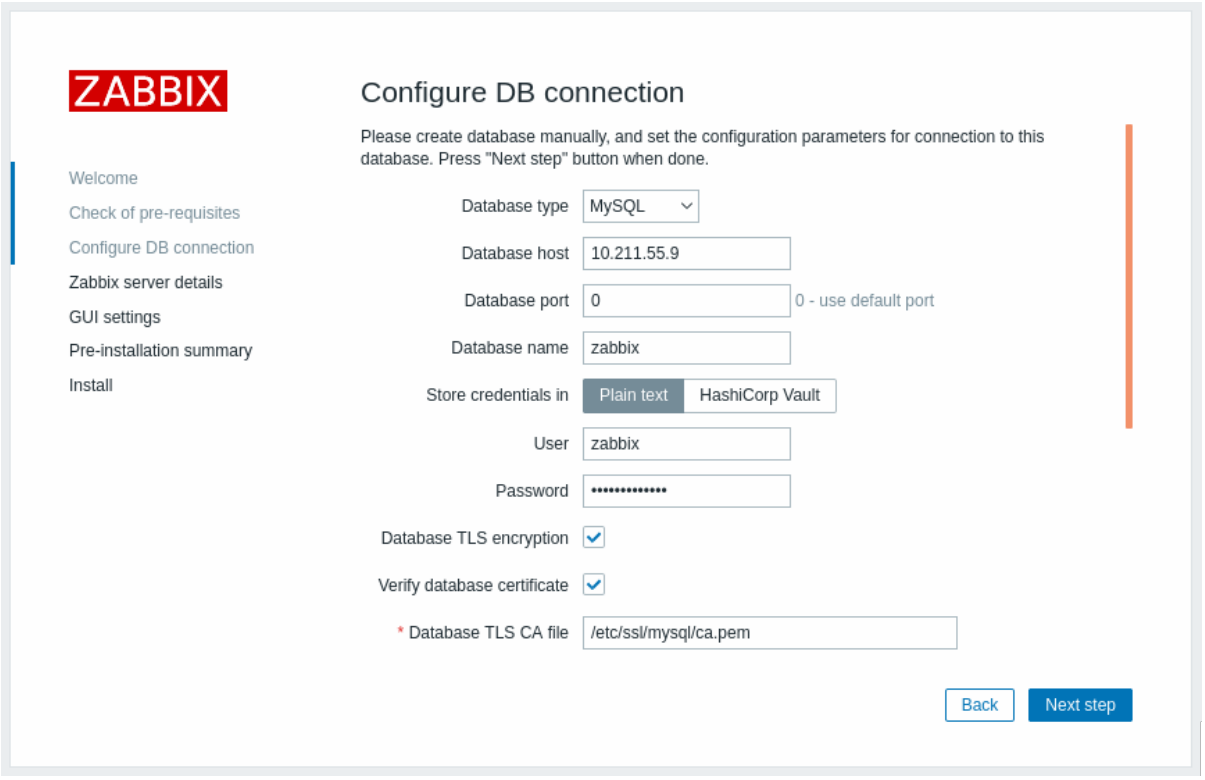
Copiez le CA MySQL requis sur le serveur de l'interface Zabbix, puis attribuez les autorisations appropriées pour permettre au serveur web de lire ce fichier.

> **Note:**
> Ce mode ne fonctionne pas sur RHEL 7 en raison de bibliothèques MySQL plus anciennes.

Interface

Pour activer le chiffrement avec vérification du certificat pour les connexions entre l'interface Zabbix et la base de données :

- Cochez *Chiffrement TLS de la base de données* et *Vérifier le certificat de la base de données*
- Indiquez le chemin vers le fichier CA TLS de la base de données



Vous pouvez également définir cette configuration dans le fichier de configuration de l'interface (`zabbix.conf.php`) :

```
...
$DB['ENCRYPTION'] = true;
$DB['KEY_FILE'] = '';
$DB['CERT_FILE'] = '';
$DB['CA_FILE'] = '/etc/ssl/mysql/ca.pem';
$DB['VERIFY_HOST'] = false;
$DB['CIPHER_LIST'] = '';
...
```

Pour résoudre les problèmes, utilisez l'outil de ligne de commande afin de vérifier si la connexion est possible pour l'utilisateur requis :

```
    mysql -u zbx_web -p -h 10.211.55.9 --ssl-mode=REQUIRED --ssl-ca=/var/lib/mysql/ca.pem
```

Serveur

Pour activer le chiffrement avec vérification du certificat pour les connexions entre le serveur Zabbix et la base de données, modifiez le fichier de configuration du serveur :

```
...
DBHost=10.211.55.9
DBName=zabbix
DBUser=zbx_srv
DBPassword=<strong_password>
DBTLSConnect=verify_ca
DBTLSCAFile=/etc/ssl/mysql/ca.pem
...
```

Chiffrement avec vérification complète

Configuration MySQL

Définissez l'option de configuration du serveur MySQL CE (*/etc/my.cnf.d/server-tls.cnf*) sur :

```
[mysqld]
...
#### dans cet exemple, les clés se trouvent dans le répertoire datadir de MySQL CE
```

```
ssl_ca=ca.pem
ssl_cert=server-cert.pem
ssl_key=server-key.pem

require_secure_transport=ON
tls_version=TLSv1.3
...
```

Les clés pour le serveur MySQL CE et le client (interface Zabbix) doivent être créées manuellement conformément à la documentation MySQL CE : Creating SSL and RSA certificates and keys using MySQL ou Creating SSL certificates and keys using openssl

Attention:
Le certificat du serveur MySQL doit contenir le champ Common Name défini sur le nom FQDN, car l'interface Zabbix utilisera le nom DNS pour communiquer avec la base de données ou l'adresse IP de l'hôte de la base de données.

Créez l'utilisateur MySQL :

```
mysql> CREATE USER
  'zbx_srv'@'%' IDENTIFIED WITH caching_sha2_password BY '<strong_password>',
  'zbx_web'@'%' IDENTIFIED WITH caching_sha2_password BY '<strong_password>'
  REQUIRE X509
  PASSWORD HISTORY 5;
```

Vérifiez qu'il est possible de se connecter avec cet utilisateur :

```
mysql -u zbx_web -p -h 10.211.55.9 --ssl-mode=VERIFY_IDENTITY --ssl-ca=/var/lib/mysql/ca.pem --ssl-cert=/v
```

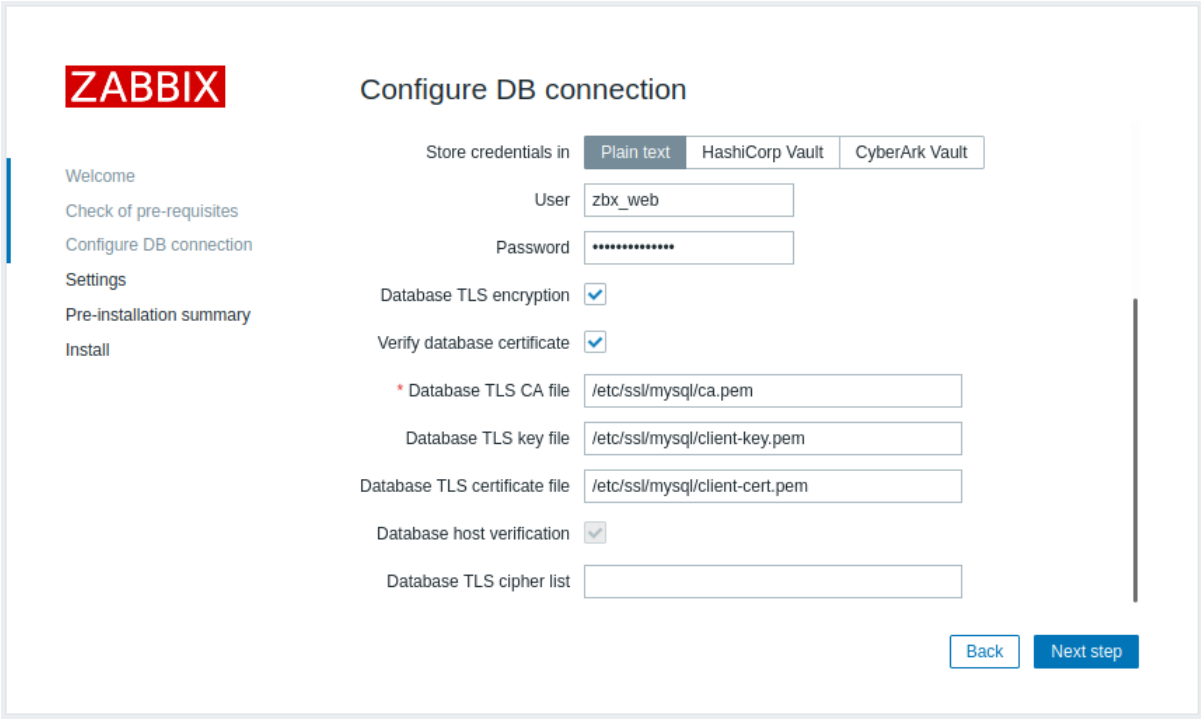
Interface

Pour activer le chiffrement avec vérification complète des connexions entre l'interface Zabbix et la base de données :

- Cochez *Chiffrement TLS de la base de données* et *Vérifier le certificat de la base de données*
- Indiquez le chemin vers le *fichier de clé TLS de la base de données*
- Indiquez le chemin vers le *fichier CA TLS de la base de données*
- Indiquez le chemin vers le *fichier de certificat TLS de la base de données*

Notez que *Vérification de l'hôte de la base de données* est cochée et grisée : cette étape ne peut pas être ignorée pour MySQL.

Warning:
Si le champ *Liste des chiffrements TLS de la base de données* est laissé vide, les chiffrements communs autorisés à la fois par l'interface (client) et le serveur seront activés. Vous pouvez également définir explicitement les chiffrements, conformément aux exigences de configuration des chiffrements.



Vous pouvez également configurer cela dans le fichier de configuration de l'interface (`zabbix.conf.php`) :

```
...
// Utilisé pour une connexion TLS avec une liste de chiffrements strictement définie.
$DB['ENCRYPTION'] = true;
$DB['KEY_FILE'] = '/etc/ssl/mysql/client-key.pem';
$DB['CERT_FILE'] = '/etc/ssl/mysql/client-cert.pem';
$DB['CA_FILE'] = '/etc/ssl/mysql/ca.pem';
$DB['VERIFY_HOST'] = true;
$DB['CIPHER_LIST'] = 'TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256:TLS_AES_1
...
// ou
...
// Utilisé pour une connexion TLS sans liste de chiffrements définie : sélectionnée par le serveur MySQL
$DB['ENCRYPTION'] = true;
$DB['KEY_FILE'] = '/etc/ssl/mysql/client-key.pem';
$DB['CERT_FILE'] = '/etc/ssl/mysql/client-cert.pem';
$DB['CA_FILE'] = '/etc/ssl/mysql/ca.pem';
$DB['VERIFY_HOST'] = true;
$DB['CIPHER_LIST'] = '';
...
```

Serveur

Pour activer le chiffrement avec vérification complète des connexions entre le serveur Zabbix et la base de données, modifiez le fichier de configuration du serveur :

```
...
DBHost=10.211.55.9
DBName=zabbix
DBUser=zbx_srv
DBPassword=<strong_password>
DBTLSConnect=verify_full
DBTLSCAFile=/etc/ssl/mysql/ca.pem
DBTLSCertFile=/etc/ssl/mysql/client-cert.pem
DBTLSKeyFile=/etc/ssl/mysql/client-key.pem
...
```

Configuration du chiffrement PostgreSQL

Vue d'ensemble

Cette section fournit plusieurs exemples de configuration du chiffrement pour CentOS 8.2 et PostgreSQL 13.

> **Note:**
> La connexion entre l'interface Zabbix et PostgreSQL ne peut pas être chiffrée (les paramètres dans l'interface graphique sont désactivés) si la valeur du champ *Database host* commence par une barre oblique ou si le champ est vide.

Pré-requis

Installez la base de données PostgreSQL à l'aide du dépôt officiel.

PostgreSQL n'est pas configuré par défaut pour accepter les connexions TLS. Veuillez suivre les instructions de la documentation PostgreSQL pour la préparation des certificats avec postgresql.conf ainsi que pour le contrôle d'accès des utilisateurs via `ph_hba.conf`.

Par défaut, le socket PostgreSQL est lié à localhost, afin que les connexions réseau distantes puissent écouter sur la véritable interface réseau.

Les paramètres PostgreSQL pour tous les modes peuvent ressembler à ceci :

`/var/lib/pgsql/13/data/postgresql.conf` :

```
...
ssl = on
ssl_ca_file = 'root.crt'
ssl_cert_file = 'server.crt'
ssl_key_file = 'server.key'
```

```
ssl_ciphers = 'HIGH:MEDIUM:+3DES:!aNULL'
ssl_prefer_server_ciphers = on
ssl_min_protocol_version = 'TLSv1.3'
...
```

Pour le contrôle d'accès, ajustez `/var/lib/pgsql/13/data/pg_hba.conf` :

```
...
### require
hostssl all all 0.0.0.0/0 md5

### verify CA
hostssl all all 0.0.0.0/0 md5 clientcert=verify-ca

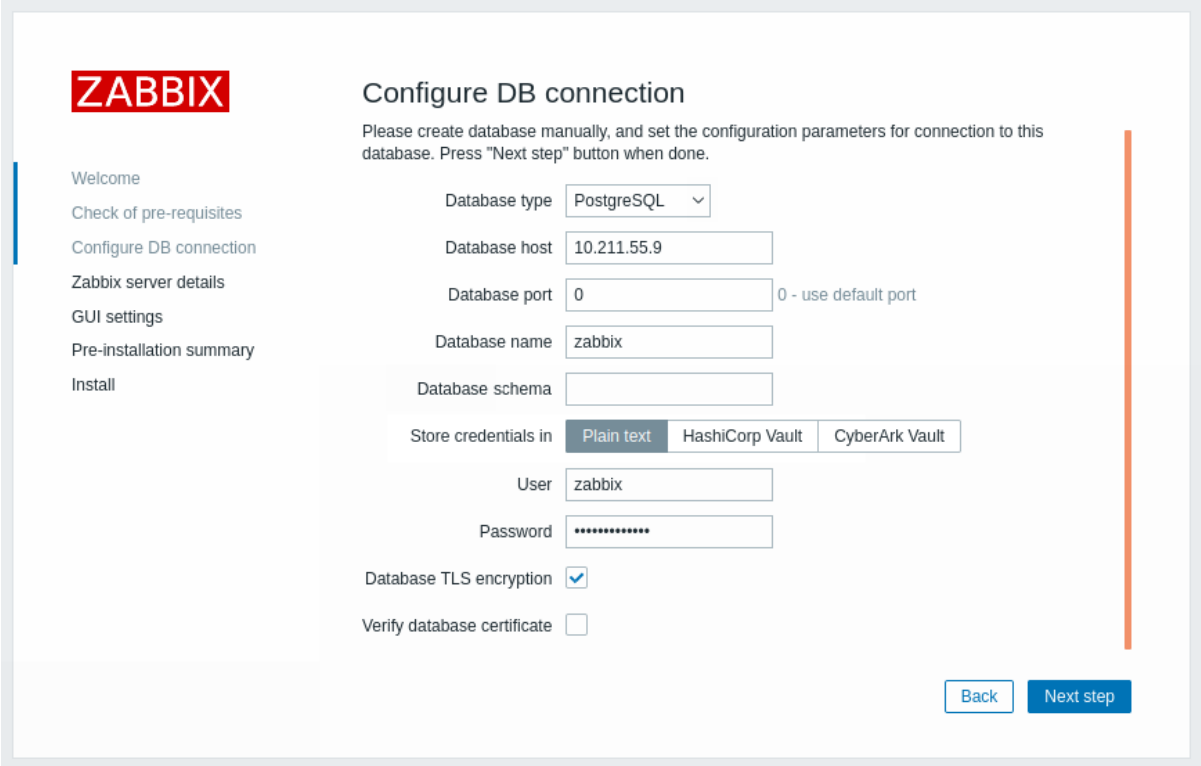
### verify full
hostssl all all 0.0.0.0/0 md5 clientcert=verify-full
...
```

Chiffrement du transport uniquement

Interface

Pour activer le chiffrement du transport uniquement pour les connexions entre l'interface Zabbix et la base de données :

- Cochez *Chiffrement TLS de la base de données*
- Laissez *Vérifier le certificat de la base de données* décoché



Serveur

Pour activer le chiffrement du transport uniquement pour les connexions entre le serveur et la base de données, modifiez le fichier de configuration du serveur :

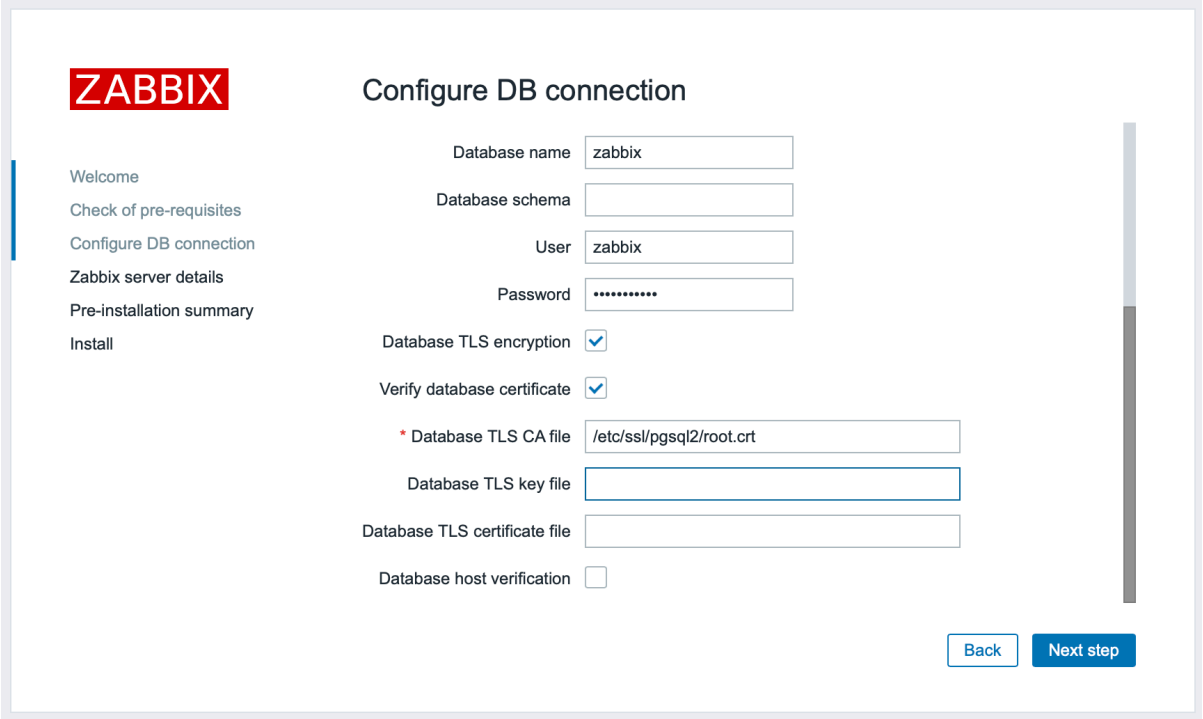
```
...
DBHost=10.211.55.9
DBName=zabbix
DBUser=zbx_srv
DBPassword=<strong_password>
DBTLSConnect=required
...
```

Chiffrement avec vérification de l'autorité de certification

Interface

Pour activer le chiffrement avec vérification de l'autorité de certification pour les connexions entre l'interface Zabbix et la base de données :

- Cochez *Chiffrement TLS de la base de données* et *Vérifier le certificat de la base de données*
- Indiquez le chemin vers le *fichier CA TLS de la base de données*



Vous pouvez également définir ce paramètre dans le fichier de configuration de l'interface (`zabbix.conf.php`) :

```
...
$DB['ENCRYPTION'] = true;
$DB['KEY_FILE'] = '';
$DB['CERT_FILE'] = '';
$DB['CA_FILE'] = '/etc/ssl/pgsql/root.crt';
$DB['VERIFY_HOST'] = false;
$DB['CIPHER_LIST'] = '';
...
```

Serveur

Pour activer le chiffrement avec vérification du certificat pour les connexions entre le serveur Zabbix et la base de données, modifiez le fichier de configuration du serveur (remplacez `<strong_password>` par un mot de passe fort et unique) :

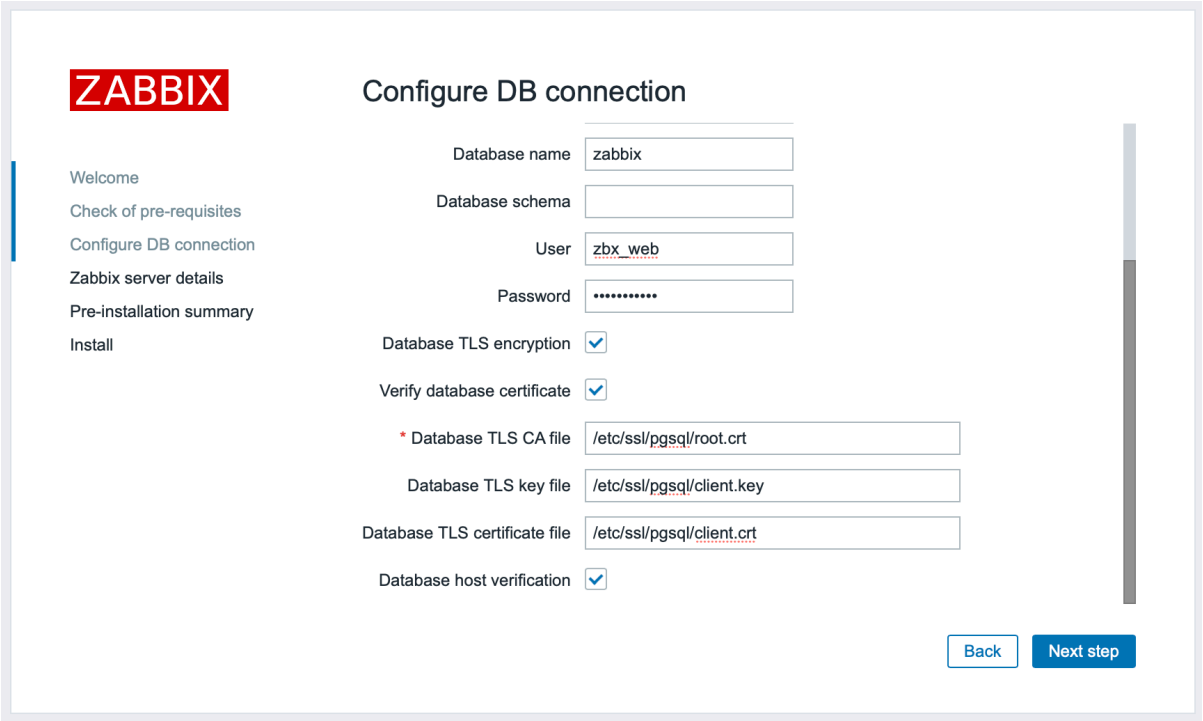
```
...
DBHost=10.211.55.9
DBName=zabbix
DBUser=zbx_srv
DBPassword=<strong_password>
DBTLSConnect=verify_ca
DBTLSCAFile=/etc/ssl/pgsql/root.crt
...
```

Chiffrement avec vérification complète

Interface

Pour activer le chiffrement avec un certificat et la vérification de l'identité de l'hôte de la base de données pour les connexions entre l'interface Zabbix et la base de données :

- Cochez *Chiffrement TLS de la base de données* et *Vérifier le certificat de la base de données*
- Indiquez le chemin vers le *fichier de clé TLS de la base de données*
- Indiquez le chemin vers le *fichier CA TLS de la base de données*
- Indiquez le chemin vers le *fichier de certificat TLS de la base de données*
- Cochez *Vérification de l'hôte de la base de données*



Vous pouvez également définir ces paramètres dans le fichier de configuration de l'interface (`zabbix.conf.php`) :

```
$DB['ENCRYPTION'] = true;
$DB['KEY_FILE'] = '';
$DB['CERT_FILE'] = '';
$DB['CA_FILE'] = '/etc/ssl/pgsql/root.crt';
$DB['VERIFY_HOST'] = true;
$DB['CIPHER_LIST'] = '';
...
```

Serveur

Pour activer le chiffrement avec vérification du certificat et de l'identité de l'hôte de la base de données pour les connexions entre le serveur Zabbix et la base de données, modifiez le fichier de configuration du serveur :

```
...
DBHost=10.211.55.9
DBName=zabbix
DBUser=zbx_srv
DBPassword=<strong_password>
DBTLSConnect=verify_full
DBTLSCAFile=/etc/ssl/pgsql/root.crt
DBTLSCertFile=/etc/ssl/pgsql/client.crt
DBTLSKeyFile=/etc/ssl/pgsql/client.key
...
```

Connexion sécurisée entre l'interface et le serveur

Vue d'ensemble

Cette section présente les étapes de configuration de Zabbix et des exemples de configuration pour établir des connexions TLS sécurisées entre l'interface Zabbix et le serveur Zabbix.

Configuration

Par défaut, la communication entre l'interface Zabbix et le serveur Zabbix n'est pas chiffrée.
Pour renforcer la sécurité, activez TLS des deux côtés.
Voici un exemple de la manière la plus simple de procéder.

1. Générez les certificats et les clés.

Créez un répertoire de travail :

```
sudo mkdir -p /etc/zabbix/ssl && cd /etc/zabbix/ssl
```

Créez un certificat d'autorité de certification (CA) (adaptez la valeur `MyZabbixCA` pour qu'elle corresponde au nom commun réel) :

```
sudo openssl genrsa -out ca.key 4096
sudo openssl req -new -x509 -days 3650 -key ca.key -out ca.crt -subj "/CN=MyZabbixCA/"
```

Générez une clé privée et un certificat pour le serveur Zabbix (adaptez la valeur `zabbix-server.example.com` pour qu'elle corresponde au nom commun réel) :

```
sudo openssl genrsa -out server.key 2048
sudo openssl req -new -key server.key -out server.csr -subj "/CN=zabbix-server.example.com/"
sudo openssl x509 -req -days 365 -in server.csr -CA ca.crt -CAkey ca.key -CAcreateserial -sha256 -out serv
```

Générez une clé privée et un certificat pour l'interface Zabbix (adaptez la valeur `zabbix-frontend-node` pour qu'elle corresponde au nom commun réel) :

```
sudo openssl genrsa -out frontend.key 2048
sudo openssl req -new -key frontend.key -out frontend.csr -subj "/CN=zabbix-frontend-node/"
sudo openssl x509 -req -days 365 -in frontend.csr -CA ca.crt -CAkey ca.key -CAcreateserial -sha256 -out fr
```

2. Définissez les autorisations appropriées.

Pour le serveur Zabbix (adaptez le propriétaire et le groupe en fonction de l'utilisateur du démon du serveur Zabbix de votre distribution) :

```
sudo chown root:zabbix /etc/zabbix/ssl/server.{crt,key} /etc/zabbix/ssl/ca.crt
sudo chmod 640 /etc/zabbix/ssl/server.key
sudo chmod 644 /etc/zabbix/ssl/server.crt /etc/zabbix/ssl/ca.crt
```

Pour l'interface (adaptez le propriétaire et le groupe en fonction de l'utilisateur du serveur web de votre distribution) :

```
sudo chown root:www-data /etc/zabbix/ssl/frontend.{crt,key}
sudo chmod 640 /etc/zabbix/ssl/frontend.key
sudo chmod 644 /etc/zabbix/ssl/frontend.crt
```

3. Configurez le serveur Zabbix.

Dans le fichier de configuration du serveur (`zabbix_server.conf`), ajoutez :

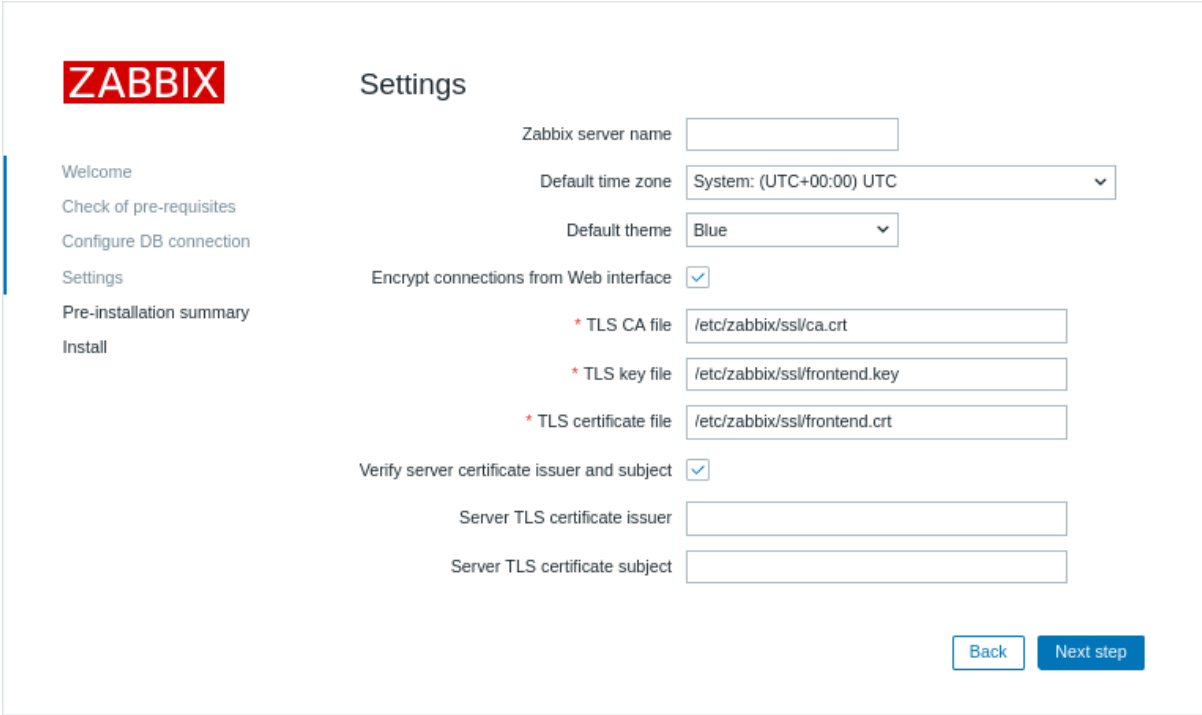
```
TLSFrontendAccept=cert
TLSCertFile=/etc/zabbix/ssl/server.crt
TLSKeyFile=/etc/zabbix/ssl/server.key
TLSCAFile=/etc/zabbix/ssl/ca.crt
### Facultatif :
### TLSFrontendCertIssuer=CN=MyZabbixCA
### TLSFrontendCertSubject=CN=zabbix-frontend-node
```

Redémarrez ensuite le serveur :

```
sudo systemctl restart zabbix-server
```

4. Configurez l'interface Zabbix.

Lors de l'installation de l'interface web, activez l'option *Chiffrer les connexions depuis l'interface web* (ainsi que l'option *Vérifier l'émetteur et le sujet du certificat du serveur*, si nécessaire) et renseignez les champs *Fichier CA TLS*, *Fichier de clé TLS*, *Fichier de certificat TLS* (ainsi que les champs *Émetteur du certificat TLS du serveur* et *Sujet du certificat TLS du serveur*, si nécessaire) :



Paramètre	Description
Fichier CA TLS	Indiquez le chemin complet vers le fichier de certificat de l'autorité de certification (CA) utilisé pour vérifier le certificat du serveur.
Fichier de clé TLS	Indiquez le chemin complet vers le fichier de clé privée du client correspondant au certificat du client.
Fichier de certificat TLS	Indiquez le chemin complet vers le fichier de certificat du client si une authentification TLS mutuelle est requise.
Émetteur du certificat TLS du serveur	Indiquez un nom distinctif (DN) de l'émetteur à comparer avec le certificat du serveur.
Sujet du certificat TLS du serveur	Indiquez un nom distinctif (DN) du sujet à comparer avec le certificat du serveur.

Dans les installations existantes, modifiez les paramètres suivants dans le fichier de configuration de l'interface :

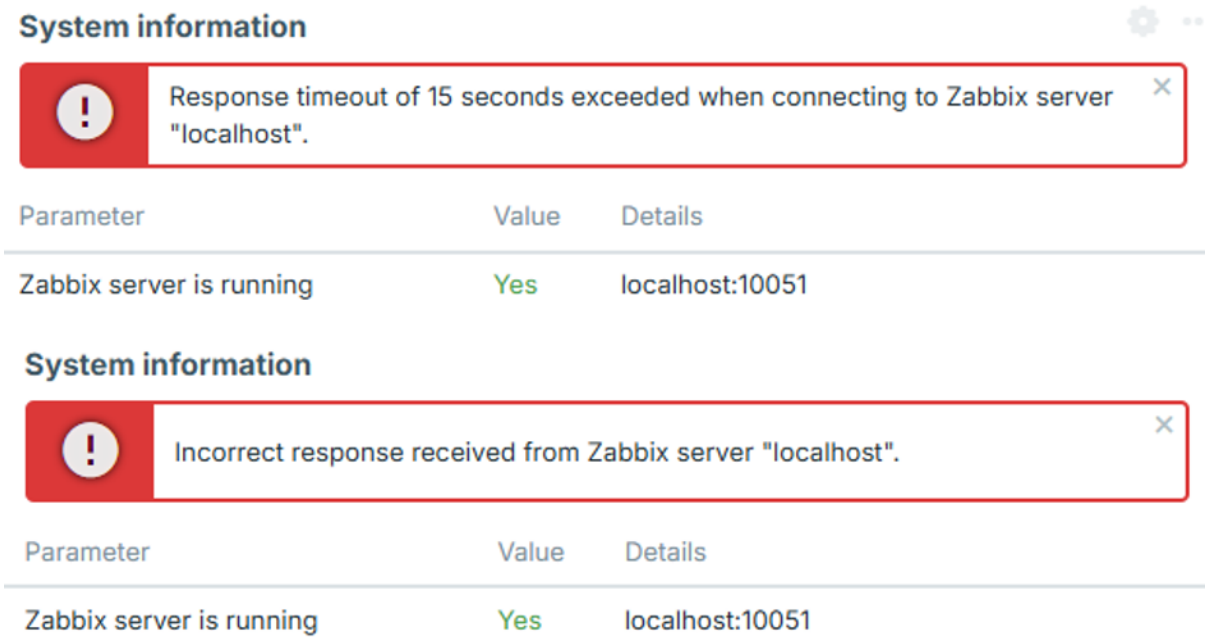
```
$ZBX_SERVER_TLS['ACTIVE'] = 'true';
$ZBX_SERVER_TLS['CA_FILE'] = '/etc/zabbix/ssl/ca.crt';
$ZBX_SERVER_TLS['KEY_FILE'] = '/etc/zabbix/ssl/frontend.key';
$ZBX_SERVER_TLS['CERT_FILE'] = '/etc/zabbix/ssl/frontend.crt';
// Facultatif :
// $ZBX_SERVER_TLS['CERTIFICATE_ISSUER']  = 'CN=MyZabbixCA';
// $ZBX_SERVER_TLS['CERTIFICATE_SUBJECT'] = 'CN=zabbix-server.example.com';
```

5. Vérifiez le chiffrement en vous assurant qu'aucun message d'erreur n'apparaît dans l'interface Zabbix ou dans le fichier journal du serveur Zabbix :

```
tail -f /var/log/zabbix/zabbix_server.log
```

Dépannage

Une configuration TLS incorrecte ou incomplète peut provoquer des erreurs dans l'interface Zabbix ou indiquer que le serveur Zabbix est en cours d'exécution, mais que la connexion ne s'établit pas correctement :



Pour identifier le problème de configuration exact, définissez le niveau de journalisation du serveur Zabbix sur 4, reproduisez l'erreur dans l'interface, puis consultez le journal du serveur Zabbix pour trouver le message d'erreur correspondant.

Les cas suivants décrivent les erreurs courantes et leurs causes possibles.

L'adresse IP de l'interface n'est pas autorisée

Le journal du serveur Zabbix contient une erreur similaire à celle-ci :

```
54283:20260819:105926.400 frontend connection from "127.0.0.1" is not allowed by FrontendAllowedIP
```

La connexion est interrompue après l'établissement de la liaison TCP, car l'adresse IP de l'interface n'est pas autorisée.

Ajoutez l'adresse IP de l'interface au paramètre `FrontendAllowedIP` dans `zabbix_server.conf`, puis redémarrez le serveur Zabbix.

La connexion non chiffrée n'est pas autorisée

Le journal du serveur Zabbix contient une erreur similaire à celle-ci :

```
56884:20260819:111154.285 frontend connection of type "unencrypted" is not allowed by TLSFrontendAccept
```

Le serveur Zabbix est configuré pour accepter uniquement les connexions TLS, par exemple avec `TLSFrontendAccept=cert`, mais l'interface tente de se connecter sans chiffrement.

Activez le chiffrement de l'interface et configurez les fichiers TLS requis dans `zabbix.conf.php`, comme décrit à l'étape 4 de la section Configuration.

L'interface tente d'utiliser TLS alors que le serveur attend une connexion non chiffrée

Le journal du serveur Zabbix contient une erreur similaire à celle-ci :

```
61091:20260819:115916.019 End of zbx_tls_accept():FAIL error:'TLS handshake set result code to 1: file ../
```

Le message d'erreur exact dépend de la version d'OpenSSL.

Le serveur Zabbix est configuré pour accepter uniquement les connexions non chiffrées, par exemple avec `TLSFrontendAccept=unencrypt` tandis que l'interface tente d'établir une connexion TLS.

Configurez le serveur Zabbix pour accepter les connexions TLS en définissant `TLSFrontendAccept=cert` et en spécifiant `TLSCertFile` et `TLSKeyFile` dans `zabbix_server.conf`. Vous pouvez également désactiver le chiffrement de l'interface dans `zabbix.conf.php`.

Le certificat de l'interface n'est pas approuvé

Le journal du serveur Zabbix contient une erreur similaire à celle-ci :

```
62248:20260819:124203.812 End of zbx_tls_accept():FAIL error:'self-signed certificate: TLS handshake set r
```

L'interface a présenté un certificat, mais le serveur Zabbix n'approuve pas l'autorité de certification (CA) qui l'a signé.

Le même problème peut également apparaître dans les informations système de l'interface sous la forme suivante :

```
fread(): SSL operation failed with code 1. OpenSSL Error messages:

error:0A000418:SSL routines::tlsv1 alert unknown ca [zabbix.php:17 → require_once() → ZBase->run() → ZBase
```

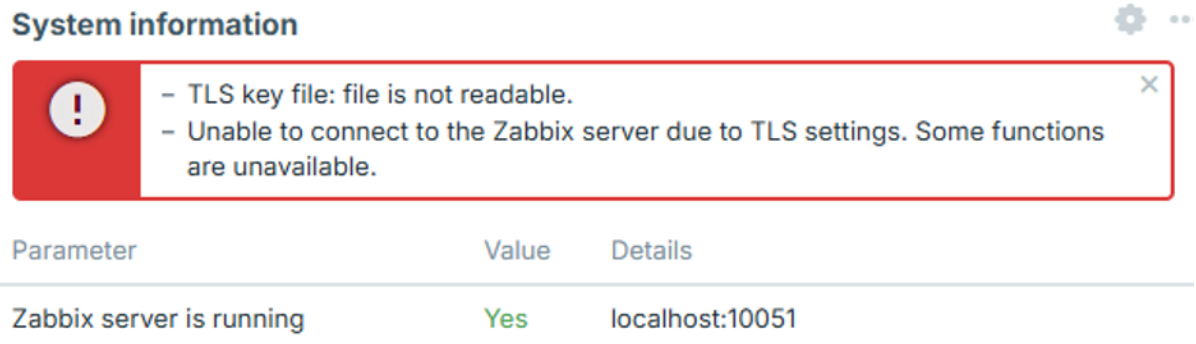
L'interface peut également afficher « Incorrect response received from Zabbix server "localhost" ».

Vérifiez que le certificat de l'interface est signé par l'autorité de certification spécifiée par `TLSCAFile` dans `zabbix_server.conf`.

Le fichier de clé TLS n'est pas lisible

L'interface peut afficher « Fichier de clé TLS : le fichier n'est pas lisible. »

Elle peut également afficher « Impossible de se connecter au serveur Zabbix en raison des paramètres TLS. Certaines fonctions ne sont pas disponibles. ».

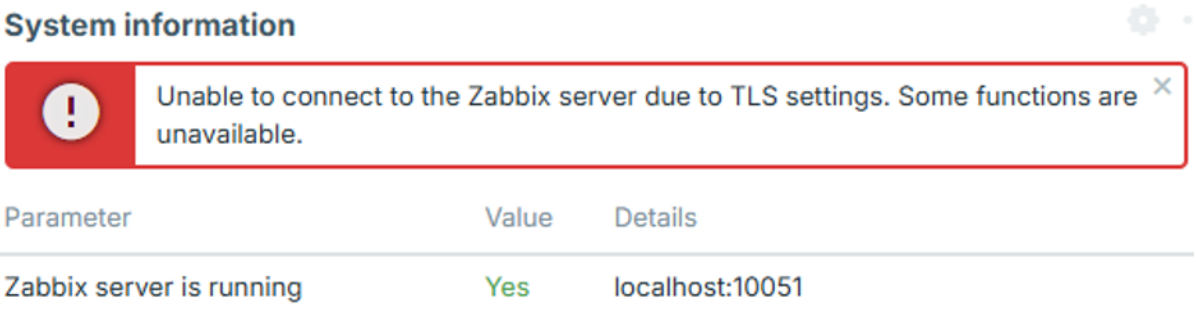


L'utilisateur du serveur web ne dispose pas des autorisations suffisantes pour lire un ou plusieurs fichiers de certificat, de clé ou d'autorité de certification configurés dans `zabbix.conf.php`. Cela peut se produire lorsque les fichiers de certificat sont stockés dans un répertoire restreint, tel que /root/ ou le répertoire personnel d'un utilisateur.

Déplacez les fichiers de certificat vers un répertoire accessible au serveur web et définissez les autorisations appropriées, comme décrit à l'étape 2 de la Configuration.

Les fichiers TLS ne sont pas valides ou sont corrompus

L'interface peut afficher « Impossible de se connecter au serveur Zabbix en raison des paramètres TLS. Certaines fonctions sont indisponibles. ».



Cela peut se produire lorsque les fichiers de certificat ou de clé configurés existent et sont lisibles, mais que leur contenu est vide, corrompu ou autrement non valide. Dans ce cas, l'interface n'établit pas de connexion au serveur Zabbix, et aucune tentative de connexion correspondante n'apparaît dans le journal du serveur Zabbix.

Vérifiez que les fichiers de certificat et de clé contiennent des données valides et correctement formatées, et que la clé privée correspond au certificat.

Contrôle d'accès

Vue d'ensemble

Cette section contient les bonnes pratiques pour configurer le contrôle d'accès de manière sécurisée.

Principe du moindre privilège

Les comptes utilisateur doivent, en permanence, fonctionner avec le moins de privilèges possible. Cela signifie que les comptes utilisateur dans l'interface Zabbix, les utilisateurs de la base de données ou l'utilisateur pour les processus du serveur/proxy/agent Zabbix ne doivent disposer que des privilèges indispensables à l'exécution des fonctions prévues.

Attention:
Accorder des privilèges supplémentaires à l'utilisateur 'zabbix' lui permettra d'accéder aux fichiers de configuration et d'exécuter des opérations susceptibles de compromettre la sécurité de l'infrastructure.

Lors de la configuration des privilèges des comptes utilisateur, les types d'utilisateurs de l'interface Zabbix doivent être pris en compte. Notez que, bien que le type d'utilisateur *Admin* dispose de moins de privilèges que le type d'utilisateur *Super Admin*, il peut tout de même gérer la configuration et exécuter des scripts personnalisés.

Note:
Certaines informations sont disponibles même pour les utilisateurs non privilégiés. Par exemple, bien que *Alerts → Scripts* ne soit disponible que pour les utilisateurs *Super Admin*, les scripts peuvent également être récupérés via l'API Zabbix. Dans ce cas, limiter les autorisations des scripts et exclure les informations sensibles des scripts (par exemple, les identifiants d'accès) peut aider à éviter l'exposition d'informations sensibles disponibles dans les scripts globaux.

Sécuriser l'utilisateur pour Zabbix agent

Par défaut, les processus du serveur Zabbix, du proxy et de l'agent (ou agent 2) partagent un seul utilisateur `zabbix`. Pour empêcher Zabbix agent/agent 2 (exécuté sur la même machine que le serveur/proxy) d'accéder à des informations sensibles dans la configuration du serveur/proxy (par exemple, les identifiants de base de données), l'agent doit être exécuté sous un utilisateur différent :

Pour Zabbix agent :

1. Créez un groupe et un utilisateur sécurisés (par exemple, `zabbix-agent`).
2. Définissez cet utilisateur dans le paramètre User du fichier de configuration de l'agent.
3. Redémarrez l'agent pour abandonner les privilèges et passer au nouvel utilisateur.

Pour Zabbix agent 2, la configuration doit être appliquée au niveau du service, car le fichier de configuration de agent 2 ne prend pas en charge le paramètre `User`. Pour un exemple, voir ZBX-26442.

Révoquer l'accès en écriture à la configuration SSL (Windows)

Si vous avez compilé l'agent Zabbix sous Windows, avec OpenSSL situé dans un répertoire non protégé (par exemple, `C:\zabbix`, `c:\openssl-64bit`, `C:\OpenSSL-Win64-111-static`, ou `C:\dev\openssl`), assurez-vous de révoquer l'accès en écriture à ce répertoire pour les utilisateurs non administrateurs. Sinon, l'agent charge les paramètres SSL depuis un chemin qui peut être modifié par des utilisateurs non privilégiés, ce qui entraîne une vulnérabilité de sécurité potentielle.

Renforcement de la sécurité des composants Zabbix

Certaines fonctionnalités peuvent être désactivées afin de renforcer la sécurité des composants Zabbix :

- L'exécution globale de scripts sur le serveur Zabbix peut être désactivée en définissant `EnableGlobalScripts=0` dans la configuration du serveur.
- L'exécution globale de scripts sur le proxy Zabbix est désactivée par défaut (elle peut être activée en définissant `EnableRemoteCommands=1` dans la configuration du proxy).
- L'exécution globale de scripts sur les agents Zabbix est désactivée par défaut (elle peut être activée en ajoutant un paramètre `AllowKey=system.run[<command>,*]` ou `AllowKeyRegexp=^system\.run\[.*\]$` pour chaque commande autorisée dans la configuration de l'agent).
- L'accès à la configuration de l'authentification HTTP des utilisateurs est activé par défaut ; il peut être désactivé en définissant `$ZBX_FEATURE_FLAGS['http_auth_enabled'] = false` dans la configuration de l'interface.
- L'accès à la configuration des modules est activé par défaut ; il peut être désactivé en définissant `$ZBX_FEATURE_FLAGS['modules_c = false` dans la configuration de l'interface.
- La modification de types de média spécifiques peut être restreinte en définissant `$ZBX_FEATURE_FLAGS['media_type_denylist'] = [<value1>, <value2>, ...]` dans la configuration de l'interface. Valeurs possibles : `'email'`, `'script'`, `'sms'`, `'webhook'` et `'push'`. Une ou plusieurs valeurs peuvent être spécifiées.

La réinstallation de l'interface en exécutant `setup.php` est restreinte si le fichier de configuration `zabbix.conf.php` existe.

Accès aux chemins UNC sous Windows par l'agent Zabbix

Les agents Zabbix sous Windows suivent les chemins UNC (partages SMB tels que `\\server\share\file.txt`) dans des éléments comme `vfs.file.*`, `vfs.dir.*`, `modbus.get` et `perf_counter*`. Cela peut présenter un risque de sécurité dans certains contextes.

Lorsque Windows est invité à accéder à un chemin UNC, il tente de s'authentifier auprès de ce serveur. Cela signifie qu'une requête malveillante adressée à l'agent Zabbix peut exposer le hachage NTLM au serveur à l'origine de la requête. Les utilisateurs peuvent atténuer ce risque en configurant, selon leurs besoins, les paramètres de configuration AllowKey, AllowKeyRegexp, DenyKey et DenyKeyRegexp.

Sécurisation de MySQL/MariaDB

Aperçu

Cette section contient les bonnes pratiques pour sécuriser une base de données MySQL/MariaDB.

> **Note:**
> Pour une configuration de base, consultez les instructions par défaut de création de base de données MySQL/MariaDB, qui incluent la création de l'utilisateur 'zabbix' avec tous les privilèges sur la base de données Zabbix. Cet utilisateur est le propriétaire de la base de données et dispose également des privilèges nécessaires pour modifier la structure de la base de données lors de la mise à niveau de Zabbix.

Pour améliorer la sécurité, il est recommandé de créer des rôles et des utilisateurs de base de données supplémentaires avec des privilèges minimaux. Ces rôles et utilisateurs doivent être configurés selon le principe du moindre privilège, c'est-à-dire qu'ils ne doivent disposer que des privilèges essentiels à l'exécution des fonctions prévues.

Création des rôles utilisateur

Créez les rôles suivants avec les privilèges correspondants :

- **zbx_srv** - rôle pour l'exécution du serveur Zabbix et du proxy :

```
CREATE ROLE 'zbx_srv';
GRANT DELETE, INSERT, SELECT, UPDATE ON zabbix.* TO 'zbx_srv';
FLUSH PRIVILEGES;
```

- **zbx_web** - rôle pour l'exécution de l'interface Zabbix et de l'API :

```
CREATE ROLE 'zbx_web';
GRANT DELETE, INSERT, SELECT, UPDATE ON zabbix.* TO 'zbx_web';
FLUSH PRIVILEGES;
```

- **zbx_bckp** - rôle pour la sauvegarde des tables :

```
CREATE ROLE 'zbx_bckp';
GRANT LOCK TABLES, TRIGGER, SELECT ON zabbix.* TO 'zbx_bckp';
GRANT process ON *.* TO 'zbx_bckp';
FLUSH PRIVILEGES;
```

> **Note:**
> La restauration et la mise à niveau des tables doivent être effectuées par le propriétaire de la base de données.

- **zbx_part** - rôle avec un ensemble réduit de privilèges pour le partitionnement de la base de données ; notez que ce rôle ne peut être créé qu'après la création de la base de données, car il accorde des privilèges sur des tables spécifiques de la base de données :

```
CREATE ROLE 'zbx_part';
GRANT SELECT, ALTER, DROP ON zabbix.history TO 'zbx_part';
GRANT SELECT, ALTER, DROP ON zabbix.history_uint TO 'zbx_part';
GRANT SELECT, ALTER, DROP ON zabbix.history_str TO 'zbx_part';
GRANT SELECT, ALTER, DROP ON zabbix.history_text TO 'zbx_part';
GRANT SELECT, ALTER, DROP ON zabbix.history_log TO 'zbx_part';
GRANT SELECT, ALTER, DROP ON zabbix.trends TO 'zbx_part';
GRANT SELECT, ALTER, DROP ON zabbix.trends_uint TO 'zbx_part';
-- For MariaDB: skip the next line (GRANT session_variables_admin ON *.* TO 'zbx_part';)
GRANT session_variables_admin ON *.* TO 'zbx_part';
GRANT SELECT ON zabbix.dbversion TO 'zbx_part';
GRANT SELECT, DELETE ON zabbix.housekeeper TO 'zbx_part';
FLUSH PRIVILEGES;
```

Après la création des rôles, ils peuvent être attribués aux utilisateurs.

Attribution des rôles utilisateur

Pour attribuer les rôles utilisateur créés, créez des utilisateurs et attribuez-leur les rôles appropriés. Remplacez <user>, <host>, <role> et <password> par les valeurs appropriées. Assurez-vous que <password> est robuste et unique.

```
CREATE USER '<user>'@'<host>' IDENTIFIED BY '<password>';
GRANT '<role>' TO '<user>'@'<host>';
SET DEFAULT ROLE '<role>' TO '<user>'@'<host>';
-- For MariaDB: SET DEFAULT ROLE '<role>' FOR '<user>'@'<host>'
FLUSH PRIVILEGES;
```

Par exemple, pour créer et attribuer le rôle permettant d'exécuter le serveur et le proxy Zabbix :

```
CREATE USER 'usr_srv'@'localhost' IDENTIFIED BY 'strong-password';
GRANT 'zbx_srv' TO 'usr_srv'@'localhost';
SET DEFAULT ROLE ALL TO 'usr_srv'@'localhost';
FLUSH PRIVILEGES;
```

Sécurisation de PostgreSQL/TimescaleDB

Aperçu

Cette section contient les bonnes pratiques pour sécuriser une base de données PostgreSQL.

> **Note:**
> Pour une configuration de base, consultez les instructions par défaut de création de la base de données PostgreSQL, qui incluent la création de l'utilisateur 'zabbix' avec tous les privilèges sur la base de données Zabbix. Cet utilisateur est le propriétaire de la base de données et dispose également des privilèges nécessaires pour modifier la structure de la base de données lors de la mise à niveau de Zabbix.

Pour améliorer la sécurité, il est recommandé de configurer un schéma d'utilisation sécurisé, ainsi que de créer des rôles et des utilisateurs de base de données supplémentaires avec des privilèges minimaux. Ces rôles et utilisateurs doivent être configurés selon le principe du moindre privilège, c'est-à-dire qu'ils ne doivent disposer que des privilèges essentiels à l'exécution des fonctions prévues.

Configuration de la base de données

Créez l'utilisateur qui sera le propriétaire de la base de données, puis créez la base de données Zabbix ; le propriétaire de la base de données est l'utilisateur spécifié lors de la création de la base de données :

```
createuser -U postgres -h localhost --pwprompt usr_owner
createdb -U postgres -h localhost -O usr_owner -E Unicode -T template0 zabbix
```

> **Note:**
> Une installation propre ou une mise à niveau de la base de données doit être effectuée par le propriétaire de la base de données. En effet, le droit de supprimer un objet de base de données ou de modifier sa définition est un privilège inhérent au propriétaire de la base de données et ne peut pas être accordé ni révoqué.

> **Attention:**
> Les commandes suivantes de cette page doivent être exécutées lorsque la connexion à PostgreSQL est établie spécifiquement sur la base de données `zabbix`.

Créez le schéma `zabbix` et définissez le propriétaire de la base de données (`usr_owner`) comme propriétaire de ce schéma :

```
CREATE SCHEMA zabbix AUTHORIZATION usr_owner;
```

Configurez un mode d'utilisation sécurisé du schéma :

```
REVOKE CREATE ON SCHEMA public FROM PUBLIC;
REVOKE ALL ON DATABASE zabbix FROM PUBLIC;
-- Note: search_path should point to the "zabbix" schema:
ALTER ROLE ALL IN DATABASE zabbix SET search_path = "zabbix";
```

Après avoir configuré la base de données, passez à la création des rôles utilisateur.

Création des rôles utilisateur

Créez les rôles suivants avec les privilèges correspondants :

- **zbx_srv** - rôle pour l'exécution du serveur Zabbix et du proxy :

```
CREATE ROLE zbx_srv;
GRANT CONNECT ON DATABASE zabbix TO zbx_srv;
GRANT USAGE ON SCHEMA zabbix TO zbx_srv;
ALTER DEFAULT PRIVILEGES FOR ROLE usr_owner IN SCHEMA zabbix GRANT DELETE, INSERT, SELECT, UPDATE ON TABLE
ALTER DEFAULT PRIVILEGES FOR ROLE usr_owner IN SCHEMA zabbix GRANT SELECT, UPDATE, USAGE ON sequences TO z
```

- **zbx_web** - rôle pour l'exécution de l'interface Zabbix et de l'API :

```
CREATE ROLE zbx_web;
GRANT CONNECT ON DATABASE zabbix TO zbx_web;
GRANT USAGE ON SCHEMA zabbix TO zbx_web;
ALTER DEFAULT PRIVILEGES FOR ROLE usr_owner IN SCHEMA zabbix GRANT DELETE, INSERT, SELECT, UPDATE ON TABLE
ALTER DEFAULT PRIVILEGES FOR ROLE usr_owner IN SCHEMA zabbix GRANT SELECT, UPDATE, USAGE ON sequences TO z
```

- **zbx_bckp** - rôle pour la sauvegarde des tables :

```
CREATE ROLE zbx_bckp;
GRANT CONNECT ON DATABASE zabbix TO zbx_bckp;
GRANT USAGE ON SCHEMA zabbix TO zbx_bckp;
ALTER DEFAULT PRIVILEGES FOR ROLE usr_owner IN SCHEMA zabbix GRANT SELECT ON TABLES TO zbx_bckp;
ALTER DEFAULT PRIVILEGES FOR ROLE usr_owner IN SCHEMA zabbix GRANT SELECT, UPDATE, USAGE ON sequences TO z
```

> **Note:**
> La restauration des tables n'est possible que par le propriétaire de la base de données.

Après la création des rôles, ils peuvent être attribués aux utilisateurs.

Attribution des rôles utilisateur

Pour attribuer les rôles utilisateur créés, créez des utilisateurs et attribuez-leur les rôles appropriés. Remplacez <user>, <role> et <password> par les valeurs appropriées. Assurez-vous que <password> est robuste et unique.

```
CREATE USER <user> WITH ENCRYPTED password '<password>';
GRANT <role> TO <user>;
```

Par exemple, pour créer et attribuer le rôle permettant d'exécuter le serveur Zabbix et le proxy :

```
CREATE USER usr_srv WITH ENCRYPTED password 'strong-password';
GRANT zbx_srv TO usr_srv;
```

Partitionnement de la base de données avec TimescaleDB

Le partitionnement de la base de données est facilité par TimescaleDB. Pour utiliser TimescaleDB, le serveur Zabbix nécessite des privilèges de propriétaire de la base de données.

Si le schéma PostgreSQL zabbix a déjà été créé dans la base de données zabbix, vous pouvez activer TimescaleDB avec la commande suivante :

```
echo "CREATE EXTENSION IF NOT EXISTS timescaledb WITH SCHEMA zabbix CASCADE;" | sudo -u postgres psql zabb
```

Cryptographie

Vue d'ensemble

Cette section contient les bonnes pratiques pour configurer la cryptographie de manière sécurisée.

Configuration de SSL pour l'interface Zabbix

Sur les systèmes basés sur RHEL, installez le paquet mod_ssl :

```
dnf install mod_ssl
```

Créez un répertoire pour les clés SSL :

```
mkdir -p /etc/httpd/ssl/private
chmod 700 /etc/httpd/ssl/private
```

Créez le certificat SSL :

```
openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/httpd/ssl/private/apache-selfsigned.key -
```

Renseignez les invites de manière appropriée. La ligne la plus importante est celle qui demande le `Common Name`. Vous devez saisir le nom de domaine que vous souhaitez associer à votre serveur. Vous pouvez saisir l'adresse IP publique à la place si vous n'avez pas de nom de domaine.

```
Country Name (2 letter code) [XX]:
State or Province Name (full name) []:
Locality Name (eg, city) [Default City]:
Organization Name (eg, company) [Default Company Ltd]:
Organizational Unit Name (eg, section) []:
Common Name (eg, your name or your server's hostname) []:example.com
Email Address []:
```

Modifiez le fichier de configuration SSL d'Apache (`/etc/httpd/conf.d/ssl.conf`) :

```
DocumentRoot "/usr/share/zabbix"
ServerName example.com:443
SSLCertificateFile /etc/httpd/ssl/apache-selfsigned.crt
SSLCertificateKeyFile /etc/httpd/ssl/private/apache-selfsigned.key
```

Redémarrez le service Apache pour appliquer les modifications :

```
systemctl restart httpd.service
```

Sécurisation du serveur web

Vue d'ensemble

Cette section contient les bonnes pratiques pour configurer le serveur web de manière sécurisée.

Forcer la redirection de l'URL racine vers Zabbix SSL

Sur les systèmes basés sur RHEL, ajoutez un hôte virtuel à la configuration d'Apache (`/etc/httpd/conf/httpd.conf`) et définissez une redirection permanente de la racine du document vers l'URL SSL de Zabbix. Notez que `example.com` doit être remplacé par le nom réel du serveur.

```
### Ajouter les lignes :

<VirtualHost *:*>
    ServerName example.com
    Redirect permanent / https://example.com
</VirtualHost>
```

Redémarrez le service Apache pour appliquer les modifications :

```
systemctl restart httpd.service
```

Activation de HTTP Strict Transport Security (HSTS) sur le serveur web

Pour protéger l'interface Zabbix contre les attaques de rétrogradation de protocole, nous recommandons d'activer la politique HSTS sur le serveur web.

Pour activer la politique HSTS pour votre interface Zabbix dans la configuration Apache, procédez comme suit :

1. Localisez le fichier de configuration de votre hôte virtuel :
   - `/etc/httpd/conf/httpd.conf` sur les systèmes basés sur RHEL
   - `/etc/apache2/sites-available/000-default.conf` sur Debian/Ubuntu
2. Ajoutez l'instruction suivante au fichier de configuration de votre hôte virtuel :

```
<VirtualHost *:*>
    Header set Strict-Transport-Security "max-age=31536000"
</VirtualHost>
```

3. Redémarrez le service Apache pour appliquer les modifications :

```
### Sur les systèmes basés sur RHEL :
systemctl restart httpd.service

### Sur Debian/Ubuntu
systemctl restart apache2.service
```

Application des cookies de session Secure et SameSite dans Zabbix

Lors de la configuration de Zabbix, il est essentiel d'imposer les attributs secure et SameSite pour les cookies de session afin d'améliorer la sécurité et d'empêcher les attaques de type cross-site request forgery (CSRF). Cependant, l'application de `SameSite=Strict` peut poser des problèmes dans certains cas, par exemple :

- Les widgets d'URL du tableau de bord affichent « user not logged in » lors de l'intégration d'iframes du même domaine.
- Les utilisateurs accédant au tableau de bord via HTTP au lieu de HTTPS peuvent rencontrer des problèmes de connexion.
- Impossible de partager des URL vers des sections spécifiques du menu Zabbix ou vers des hôtes.

Pour atténuer ces problèmes, les utilisateurs doivent pouvoir ajuster la politique SameSite.

1. Cookies sécurisés

La définition de l'indicateur `secure` garantit que les cookies ne sont transmis que via HTTPS, ce qui évite leur exposition sur des connexions non chiffrées.

Pour activer les cookies sécurisés dans Zabbix, ajoutez ou modifiez le paramètre suivant dans la configuration du serveur web :

Pour Apache :

```
Header always edit Set-Cookie ^(.*)$ $1;Secure
```

Pour Nginx :

```
proxy_cookie_path / "/; Secure";
```

Assurez-vous que votre interface Zabbix est accessible via HTTPS ; sinon, les cookies avec l'indicateur `Secure` ne seront pas envoyés.

2. Configuration de l'attribut SameSite

Les paramètres du serveur web peuvent également imposer l'attribut SameSite :

Pour Apache :

```
<IfModule mod_headers.c>
    Header onsuccess edit Set-Cookie (.*) "$1; SameSite=Strict"
</IfModule>
```

Pour Nginx (version 1.19.3+) :

```
proxy_cookie_flags ~ samesite=Strict; # Replace ~ with 'zbx_session' for specificity
```

Activation de Content Security Policy (CSP) sur le serveur web

Pour protéger l'interface Zabbix contre les attaques de type Cross Site Scripting (XSS), l'injection de données et les attaques similaires, nous recommandons d'activer Content Security Policy sur le serveur web. Pour ce faire, configurez le serveur web afin qu'il renvoie l'en-tête HTTP.

> **Attention:**
> La configuration d'en-tête CSP suivante s'applique uniquement à l'installation par défaut de l'interface Zabbix et aux cas où tout le contenu provient du domaine du site (à l'exclusion des sous-domaines). Une configuration d'en-tête CSP différente peut être nécessaire si, par exemple, vous configurez le widget *URL* pour afficher du contenu provenant des sous-domaines du site ou de domaines externes, si vous passez d'*OpenStreetMap* à un autre moteur de cartographie, ou si vous ajoutez du CSS externe ou des widgets. Si vous utilisez la méthode d'authentification multifacteur Duo Universal Prompt multi-factor authentication, veillez à ajouter "duo.com" à la directive CSP dans le fichier de configuration de votre hôte virtuel.

Pour activer CSP pour votre interface Zabbix dans la configuration Apache, procédez comme suit :

1. Localisez le fichier de configuration de votre hôte virtuel :

- `/etc/httpd/conf/httpd.conf` sur les systèmes basés sur RHEL
- `/etc/apache2/sites-available/000-default.conf` sur Debian/Ubuntu

2. Ajoutez la directive suivante au fichier de configuration de votre hôte virtuel :

```
<VirtualHost *:*>
    Header set Content-Security-Policy: "default-src 'self' *.openstreetmap.org; script-src 'self' 'unsafe
</VirtualHost>
```

3. Redémarrez le service Apache pour appliquer les modifications :

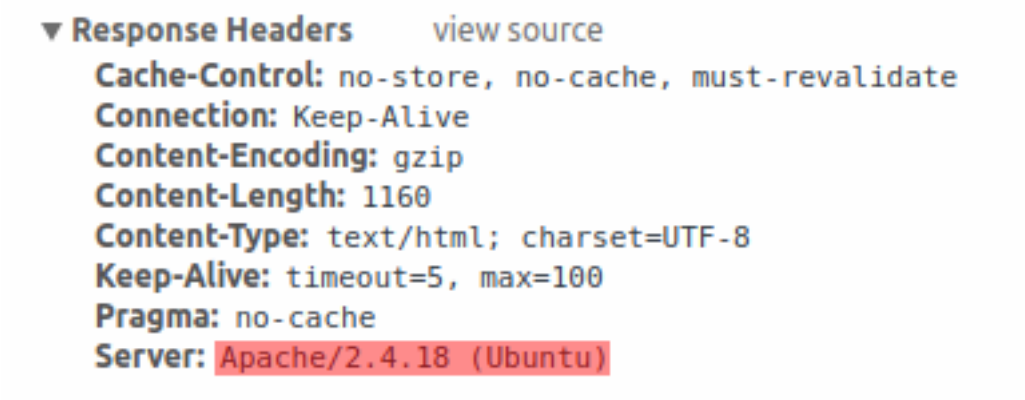
```
### Sur les systèmes basés sur RHEL :
systemctl restart httpd.service

### Sur Debian/Ubuntu
systemctl restart apache2.service
```

Désactivation de l'exposition des informations du serveur web

Pour améliorer la sécurité, il est recommandé de désactiver toutes les signatures du serveur web.

Par défaut, le serveur web expose la signature du logiciel :



La signature peut être désactivée en ajoutant les paramètres suivants au fichier de configuration Apache :

```
ServerSignature Off
ServerTokens Prod
```

La signature PHP (en-tête HTTP X-Powered-By) peut être désactivée en modifiant le fichier de configuration `php.ini` (par défaut, la signature est désactivée) :

```
expose_php = Off
```

Le redémarrage du serveur web est nécessaire pour que les modifications du fichier de configuration soient appliquées.

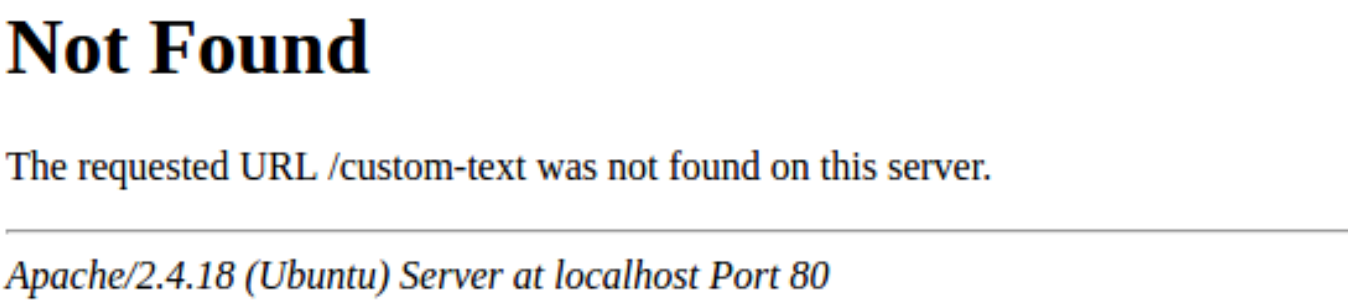
Pour renforcer la sécurité, vous pouvez utiliser l'outil *mod_security* avec Apache (paquet *libapache2-mod-security2*). Cet outil permet de supprimer la signature du serveur au lieu de supprimer uniquement la version de la signature du serveur. La signature du serveur peut être remplacée par n'importe quelle valeur en définissant « SecServerSignature » sur la valeur souhaitée après l'installation de *mod_security*.

Veuillez consulter la documentation de votre serveur web pour savoir comment supprimer ou modifier les signatures des logiciels.

Désactivation des pages d'erreur par défaut du serveur web

Pour éviter toute divulgation d'informations, il est recommandé de désactiver les pages d'erreur par défaut.

Par défaut, un serveur web utilise des pages d'erreur intégrées :



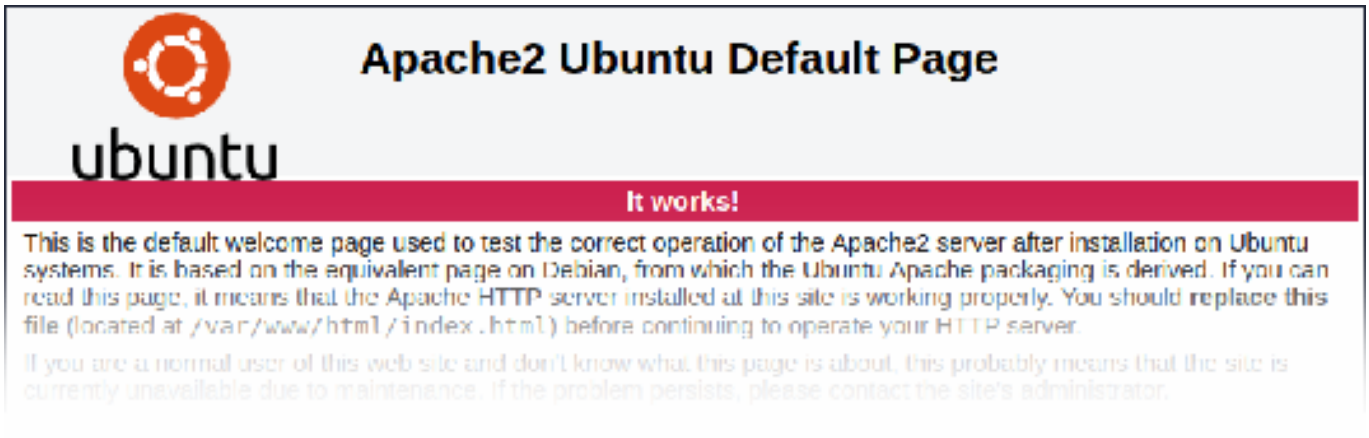
Ces pages d'erreur par défaut doivent être remplacées ou supprimées. Par exemple, la directive "ErrorDocument" peut être utilisée pour définir une page ou un texte d'erreur personnalisé pour le serveur web Apache.

Veuillez consulter la documentation de votre serveur web pour savoir comment remplacer ou supprimer les pages d'erreur par défaut.

Suppression de la page de test du serveur web

Pour éviter toute divulgation d'informations, il est recommandé de supprimer la page de test du serveur web.

Par défaut, la racine web du serveur web Apache contient la page de test `index.html` :



Veuillez consulter la documentation de votre serveur web pour savoir comment supprimer les pages de test par défaut.

Définir l'en-tête de réponse HTTP X-Frame-Options

Par défaut, Zabbix est configuré avec le paramètre *Use X-Frame-Options HTTP header* défini sur `SAMEORIGIN`. Cela signifie que le contenu ne peut être chargé que dans un cadre ayant la même origine que la page elle-même.

Les éléments de l'interface Zabbix qui récupèrent du contenu à partir d'URL externes (à savoir le widget de tableau de bord URL) affichent le contenu récupéré dans un sandbox avec toutes les restrictions de sandboxing activées.

Ces paramètres renforcent la sécurité de l'interface Zabbix et offrent une protection contre les attaques XSS et de clickjacking. Les utilisateurs *Super admin* peuvent modifier les paramètres *Use iframe sandboxing* et *Use X-Frame-Options HTTP header* selon les besoins. Veuillez évaluer soigneusement les risques et les avantages avant de modifier les paramètres par défaut. Il n'est pas recommandé de désactiver complètement le sandboxing des iframes ou l'en-tête HTTP X-Frame-Options.

Masquer le fichier contenant la liste des mots de passe courants

Pour augmenter la complexité des attaques par force brute sur les mots de passe, il est recommandé de limiter l'accès au fichier `ui/data/top_passwords.txt`. Ce fichier contient une liste des mots de passe les plus courants et spécifiques au contexte, et empêche les utilisateurs de définir de tels mots de passe (si le paramètre *Éviter les mots de passe faciles à deviner* est activé dans la politique de mot de passe).

Pour limiter l'accès au fichier `top_passwords.txt`, modifiez la configuration de votre serveur web.

Sur Apache, l'accès au fichier peut être limité à l'aide du fichier `.htaccess` :

```
<Files "top_passwords.txt">
    Order Allow,Deny
    Deny from all
</Files>
```

Sur NGINX, l'accès au fichier peut être limité à l'aide de la directive `location` :

```
location = /data/top_passwords.txt {
    deny all;
    return 404;
}
```

Stockage des secrets

Vue d'ensemble Zabbix peut être configuré pour récupérer des informations sensibles depuis un coffre-fort sécurisé. Les services de gestion des secrets suivants sont pris en charge: HashiCorp Vault KV Secrets Engine - Version 2, CyberArk Vault CV12.

Les secrets peuvent être utilisés pour récupérer:

- valeurs de macro utilisateur
- des identifiants d'accès à la base de données

Zabbix fournit un accès en lecture seule aux secrets dans un coffre-fort, en supposant que les secrets sont gérés par quelqu'un d'autre.

Pour plus d'informations sur la configuration d'un fournisseur de coffre-fort spécifique, consultez :

- Configuration HashiCorp
- Configuration CyberArk

Mise en cache des valeurs secrètes Par défaut, les valeurs des macros secrètes du vault sont récupérées par le serveur Zabbix à chaque actualisation des données de configuration, puis stockées dans le cache de configuration. Le proxy Zabbix reçoit les valeurs des macros secrètes du vault du serveur Zabbix lors de chaque synchronisation de la configuration et les stocke dans son propre cache de configuration.

> **Attention:**
> Le chiffrement doit être activé entre le serveur Zabbix et le proxy ; sinon, un message d'avertissement du serveur est enregistré.

Il est également possible de configurer le serveur Zabbix et le proxy Zabbix afin qu'ils récupèrent indépendamment les valeurs des macros.

Pour déclencher manuellement l'actualisation des valeurs secrètes mises en cache depuis un vault, utilisez l'option de ligne de commande `secrets_reload`.

La mise en cache des identifiants de la base de données de l'interface Zabbix est désactivée par défaut, mais peut être activée en définissant l'option `$DB['VAULT_CACHE'] = true` dans le fichier de configuration de l'interface. Les identifiants seront stockés dans un cache local utilisant le répertoire de fichiers temporaires du système de fichiers. Le serveur web doit autoriser l'écriture dans un dossier temporaire privé (par exemple, pour Apache, l'option de configuration `PrivateTmp=True` doit être définie). Pour contrôler la fréquence d'actualisation/invalidation du cache de données, utilisez la constante `ZBX_DATA_CACHE_TTL`.

Configuration TLS Pour configurer TLS pour la communication entre les composants Zabbix et le coffre-fort, ajoutez un certificat signé par une autorité de certification (CA) au magasin CA par défaut du système. Pour utiliser un autre emplacement, spécifiez le répertoire dans le paramètre de configuration SSLCALocation de Zabbix serveur/proxy, placez le fichier de certificat dans ce répertoire, puis exécutez la commande CLI :

```
c_rehash .
```

Configuration CyberArk

Cette section explique comment configurer Zabbix pour récupérer des secrets depuis CyberArk Vault CV12.

Le coffre-fort doit être installé et configuré comme décrit dans la documentation officielle CyberArk.

Pour en savoir plus sur la configuration de TLS dans Zabbix, consultez *Stockage des secrets*.

Identifiants de la base de données

L'accès à un secret contenant les identifiants de la base de données est configuré séparément pour chaque composant Zabbix.

Serveur et proxies

Pour obtenir les identifiants de la base de données depuis le coffre-fort pour le serveur ou le proxy Zabbix, spécifiez les paramètres de configuration suivants dans le fichier de configuration :

- `Vault` - fournisseur de coffre-fort à utiliser ;
- `VaultURL` - URL HTTP(S) du serveur de coffre-fort ;
- `VaultDBPath` - requête vers le secret du coffre-fort contenant les identifiants de la base de données, qui seront récupérés à l'aide des clés "Content" et "UserName" (cette option ne peut être utilisée que si DBUser et DBPassword ne sont pas spécifiés) ;
- `VaultTLSCertFile`, `VaultTLSKeyFile` - noms des fichiers de certificat et de clé SSL ; la configuration de ces options n'est pas obligatoire, mais elle est fortement recommandée ;

- `VaultPrefix` - préfixe personnalisé pour le chemin ou la requête du coffre-fort, selon le coffre-fort ; s'il n'est pas spécifié, la valeur par défaut la plus appropriée sera utilisée.

Attention:
Les paramètres de configuration `Vault`, `VaultURL`, `VaultTLSCertFile`, `VaultTLSKeyFile` et `VaultPrefix` sont également utilisés pour l'authentification auprès du coffre-fort lors du traitement des macros de coffre-fort secrètes par le serveur Zabbix (et le proxy Zabbix, s'il est configuré). Le serveur Zabbix et les proxies n'ouvriront pas les macros de coffre-fort secrètes contenant les identifiants de la base de données provenant de VaultDBPath.

Le serveur Zabbix et le proxy Zabbix lisent les paramètres de configuration liés au coffre-fort dans les fichiers `zabbix_server.conf` et `zabbix_proxy.conf` au démarrage.

Exemple

1. Dans `zabbix_server.conf`, spécifiez les paramètres suivants :

```
Vault=CyberArk
VaultURL=https://127.0.0.1:1858
VaultDBPath=AppID=zabbix_server&Query=Safe=passwordSafe;Object=zabbix_server_database
VaultTLSCertFile=cert.pem
VaultTLSKeyFile=key.pem
VaultPrefix=/AIMWebService/api/Accounts?
```

2. Zabbix enverra la requête API suivante au coffre-fort :

```
curl \
--header "Content-Type: application/json" \
--cert cert.pem \
--key key.pem \
https://127.0.0.1:1858/AIMWebService/api/Accounts?AppID=zabbix_server&Query=Safe=passwordSafe;Object=zabbi
```

3. La réponse du coffre-fort contiendra les clés "Content" et "UserName" :

```
{
    "Content": <password>,
    "UserName": <username>,
    "Address": <address>,
    "Database": <Database>,
    "PasswordChangeInProcess":<PasswordChangeInProcess>
}
```

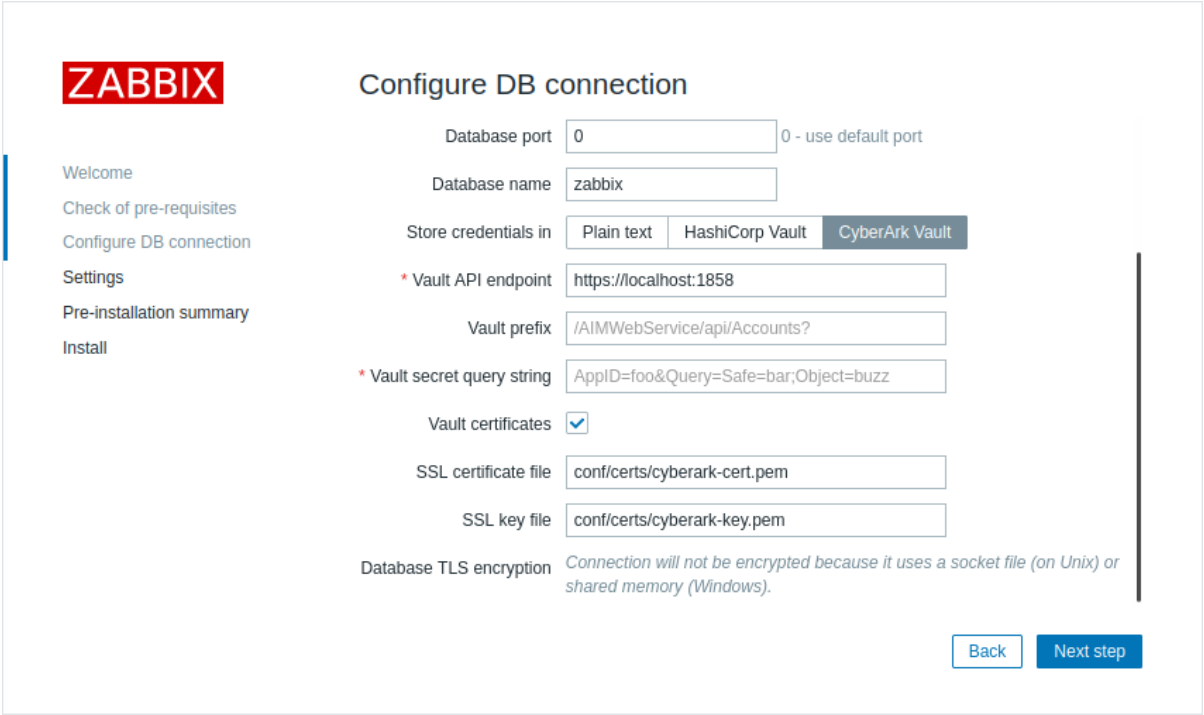
4. En conséquence, Zabbix utilisera les identifiants suivants pour l'authentification à la base de données :

- Nom d'utilisateur: <username>
- Mot de passe: <password>

Interface

Pour obtenir les identifiants de la base de données depuis le coffre-fort pour l'interface Zabbix, indiquez les paramètres suivants lors de l'installation de l'interface.

1. À l'étape *Configure DB Connection*, définissez le paramètre *Store credentials in* sur *CyberArk Vault*.



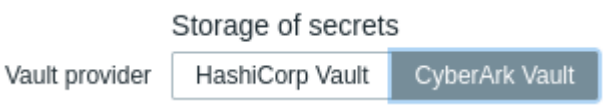
2. Ensuite, renseignez les paramètres supplémentaires :

Parameter	Mandatory	Default value	Description
Vault API endpoint	yes	https://localhost:1858	Indiquez l'URL de connexion au coffre-fort au format `scheme://host:port`
Vault prefix	no	/AIMWebService/api/Accounts?	Indiquez un préfixe personnalisé pour le chemin ou la requête du coffre-fort. S'il n'est pas indiqué, la valeur par défaut est utilisée.
Vault secret query string	yes		Requête indiquant où les identifiants de la base de données doivent être récupérés. Exemple : `AppID=foo&Query=Safe=bar;Object=buzz`
Vault certificates	no		Après avoir coché cette case, des paramètres supplémentaires apparaîtront pour vous permettre de configurer l'authentification du client. Bien que ce paramètre soit facultatif, il est fortement recommandé de l'activer pour les communications avec le CyberArk Vault.
SSL certificate file	no	conf/certs/cyberark-cert.pem	Chemin vers le fichier de certificat SSL. Le fichier doit être au format PEM. Si le fichier de certificat contient également la clé privée, laissez le paramètre SSL key file vide.
SSL key file	no	conf/certs/cyberark-key.pem	Nom du fichier de clé privée SSL utilisé pour l'authentification du client. Le fichier doit être au format PEM.

Valeurs des macros utilisateur

Pour utiliser CyberArk Vault afin de stocker les valeurs des macros utilisateur *Vault secret*, assurez-vous que :

- le serveur Zabbix est configuré pour fonctionner avec CyberArk Vault ;
- le paramètre *Vault provider* dans *Administration > General > Other* est défini sur "CyberArk Vault".



Note:
Le serveur Zabbix (et le proxy Zabbix, s'il est configuré) nécessite un accès aux valeurs des macros *Vault secret* depuis le coffre-fort. L'interface Zabbix n'a pas besoin d'un tel accès.

La valeur de la macro doit contenir une requête (au format `query:key`).

Consultez *Vault secret macros* pour obtenir des informations détaillées sur le traitement des valeurs de macro par Zabbix.

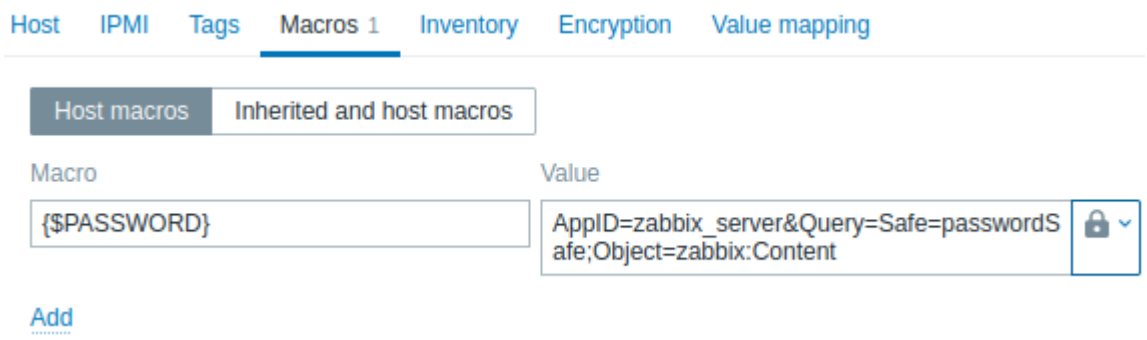
Syntaxe des requêtes

Le symbole deux-points (`:`) est réservé pour séparer la requête de la clé.

Si une requête contient elle-même une barre oblique ou un deux-points, ces symboles doivent être encodés dans l'URL (`/` est encodé en `%2F`, `:` est encodé en `%3A`).

Exemple

1. Dans Zabbix, ajoutez une macro utilisateur `{$PASSWORD}` de type *Vault secret* et avec la valeur `AppID=zabbix_server&Query=Safe`



2. Zabbix enverra la requête API suivante au coffre-fort :

```
curl \
--header "Content-Type: application/json" \
--cert cert.pem \
--key key.pem \
https://127.0.0.1:1858/AIMWebService/api/Accounts?AppID=zabbix_server&Query=Safe=passwordSafe;Object=zabbi
```

3. La réponse du coffre-fort contiendra la clé `Content` :

```
{
    "Content": <password>,
    "UserName": <username>,
    "Address": <address>,
    "Database" :<Database>,
    "PasswordChangeInProcess":<PasswordChangeInProcess>
}
```

4. Par conséquent, Zabbix résoudra la macro `{$PASSWORD}` en la valeur - `<password>`

Mettre à jour une configuration existante

Pour mettre à jour une configuration existante afin de récupérer des secrets depuis un CyberArk Vault :

1. Mettez à jour les paramètres du fichier de configuration du serveur Zabbix ou du proxy, comme décrit dans la section *Identifiants de base de données*.

2. Mettez à jour les paramètres du fichier de configuration de l'interface Zabbix :

```
$DB['VAULT']                    = 'CyberArk';
$DB['VAULT_URL']                = 'https://127.0.0.1:1858';
$DB['VAULT_DB_PATH']            = 'AppID=foo&Query=Safe=bar;Object=buzz';
$DB['VAULT_TOKEN']              = '';
$DB['VAULT_CERT_FILE']           = 'conf/certs/cyberark-cert.pem';
$DB['VAULT_KEY_FILE']            = 'conf/certs/cyberark-key.pem';
$DB['VAULT_PREFIX']              = '';
```

3. Configurez les macros utilisateur comme décrit dans la section *Valeurs des macros utilisateur*, si nécessaire.

Pour mettre à jour une configuration existante afin de récupérer des secrets depuis un HashiCorp Vault, consultez la section *Configuration HashiCorp*.

Configuration HashiCorp

Vue d'ensemble

Cette section explique comment configurer Zabbix pour récupérer des secrets depuis HashiCorp Vault KV Secrets Engine - Version 2.

Le vault doit être déployé et configuré comme décrit dans la documentation officielle de HashiCorp.

Pour en savoir plus sur la configuration de TLS dans Zabbix, consultez *Stockage des secrets*.

Récupération des identifiants de la base de données

Pour récupérer correctement un secret contenant les identifiants de la base de données, il est nécessaire de configurer les deux éléments suivants :

- serveur/proxy Zabbix
- interface Zabbix

Serveur/proxy

Pour configurer le serveur ou le proxy Zabbix, spécifiez les paramètres de configuration suivants dans le fichier de configuration :

- `Vault` - fournisseur de coffre-fort à utiliser.
- `VaultToken` - jeton d'authentification du coffre-fort (voir le fichier de configuration du serveur/proxy Zabbix pour plus de détails) ; ne peut pas être utilisé avec `VaultAppRoleID`.
- `VaultAppRoleID` et `VaultAppSecretID` - ID de rôle et ID secret d'authentification du coffre-fort (voir le fichier de configuration du serveur/proxy Zabbix pour plus de détails) ; ne peuvent pas être utilisés avec `VaultToken`.
- `VaultURL` - URL HTTP(S) du serveur de coffre-fort.
- `VaultDBPath` - chemin vers le secret du coffre-fort contenant les identifiants de la base de données (cette option ne peut être utilisée que si DBUser et DBPassword ne sont pas spécifiés). Le serveur ou le proxy Zabbix récupérera les identifiants à l'aide des clés « password » et « username ».
- `VaultPrefix` - préfixe personnalisé pour le chemin ou la requête du coffre-fort, selon le coffre-fort ; s'il n'est pas spécifié, la valeur par défaut la plus appropriée sera utilisée. Notez que le préfixe doit commencer et se terminer par `/`.

Attention:
Les paramètres de configuration `Vault`, `VaultToken` (ou `VaultAppRoleID`/`VaultAppSecretID`), `VaultURL` et `VaultPrefix` sont également utilisés pour l'authentification auprès du coffre-fort lors du traitement des macros secrètes du coffre-fort par le serveur Zabbix (et le proxy Zabbix, s'il est configuré). Le serveur Zabbix et les proxies n'ouvriront pas les macros secrètes du coffre-fort contenant les identifiants de la base de données provenant de VaultDBPath.

L'utilisation de jetons différents pour différents proxies est fortement recommandée.

Le serveur Zabbix et le proxy Zabbix lisent les paramètres de configuration liés au coffre-fort depuis `zabbix_server.conf` et `zabbix_proxy.conf` au démarrage.

De plus, le serveur Zabbix et le proxy Zabbix lisent la variable d'environnement `VAULT_TOKEN` une seule fois au démarrage, puis la désactivent afin qu'elle ne soit pas accessible via les scripts forkés ; une erreur se produit si les paramètres `VaultToken` et `VAULT_TOKEN` contiennent tous deux une valeur.

Exemple

1. Dans `zabbix_server.conf`, spécifiez les paramètres suivants.

En utilisant VaultToken :

```
Vault=HashiCorp
VaultToken=hvs.CAESIIG_PILmULFYOsEyWHxkZ2mF2a8VPKNLE8eHqd4autYGGh4KHGh2cy5aeTYONFNSaUp3ZnpWbDF1RUNjUkNTZEg
VaultURL=https://127.0.0.1:8200
VaultDBPath=database
VaultPrefix=/v1/secret/data/zabbix/
```

Ou, en utilisant VaultAppRoleID :

```
Vault=HashiCorp
VaultAppRoleID=6f1567f1-538d-7d70-e8e4-5d3e59471b2e
VaultAppSecretID=89dac8b1-f7d6-96a7-2ecf-f94d2de3d6f1
VaultURL=https://127.0.0.1:8200
```

```
VaultDBPath=database
VaultPrefix=/v1/secret/data/zabbix/
```

2. Exécutez les commandes CLI suivantes pour créer le secret requis dans le vault :

```
#### Activer le point de montage "secret/" s'il n'est pas déjà activé ; notez que "kv-v2" doit être utilis
vault secrets enable -path=secret/ kv-v2

#### Ajouter de nouveaux secrets avec les clés username et password sous le point de montage "secret/" et
vault kv put -mount=secret zabbix/database username=zabbix password=<password>

#### Vérifier que le secret a bien été ajouté.
vault kv get secret/zabbix/database

#### Enfin, tester avec Curl ; notez que "data" doit être ajouté manuellement après le point de montage et
curl --header "X-Vault-Token: <VaultToken>" https://127.0.0.1:8200/v1/secret/data/zabbix/database
```

3. En conséquence, le serveur Zabbix récupérera les identifiants suivants pour l'authentification à la base de données :

- Nom d'utilisateur : zabbix
- Mot de passe : <password>

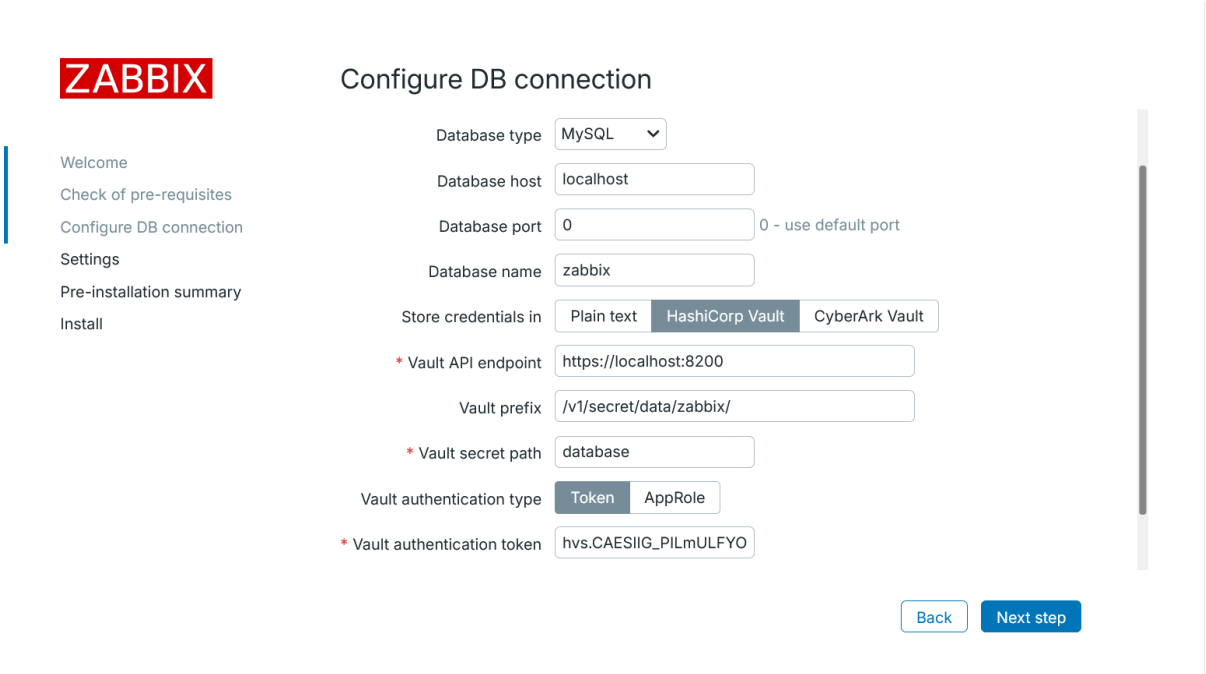
Interface

L'interface Zabbix peut être configurée pour récupérer les identifiants de la base de données depuis le coffre-fort, soit lors de l'installation de l'interface, soit en mettant à jour le fichier de configuration de l'interface (`zabbix.conf.php`).

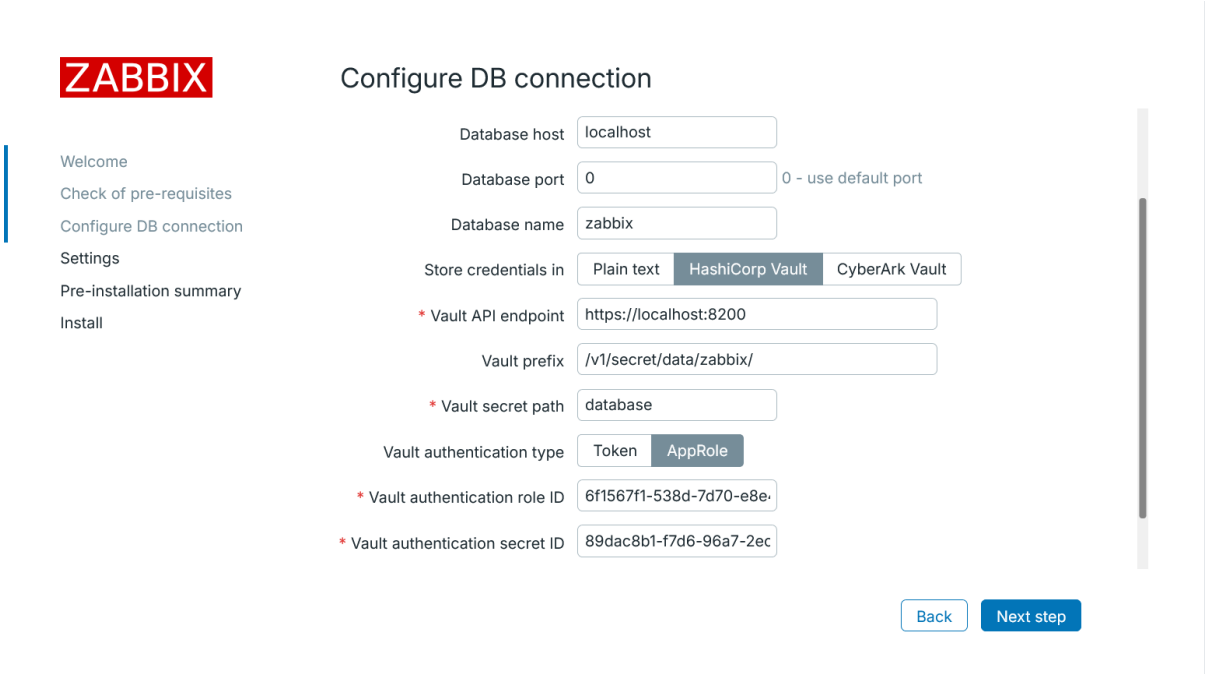
Attention:
Si les identifiants du coffre-fort ont changé depuis l'installation précédente de l'interface, mettez à jour la configuration existante.

Lors de l'**installation de l'interface**, les paramètres de connexion doivent être spécifiés à l'étape *Configure DB Connection*.

Exemple de configuration utilisant l'authentification par jeton :



Exemple de configuration utilisant l'authentification AppRole :



Paramètre	Obligatoire	Valeur par défaut	Description
Store credentials in	non		Sélectionnez **HashiCorp Vault**.
Vault API endpoint	oui	https://localhost:8200	Spécifiez l'URL de connexion au coffre-fort au format `scheme://host:port`
Vault prefix	non	/v1/secret/data/	Indiquez un préfixe personnalisé pour le chemin ou la requête du coffre-fort. S'il n'est pas spécifié, la valeur par défaut est utilisée. Exemple : `/v1/secret/data/zabbix/`
Vault secret path	oui		Chemin vers le secret à partir duquel les identifiants de la base de données doivent être récupérés à l'aide des clés « password » et « username ». Exemple : `database`
Vault authentication type	non		Sélectionnez le type d'authentification du coffre-fort : **Token** - utiliser l'authentification par jeton ; **AppRole** - utiliser la méthode d'authentification AppRole.
Vault authentication token	oui		Indiquez un jeton d'authentification permettant un accès en lecture seule au chemin du secret. Consultez la documentation HashiCorp pour obtenir des informations sur la création de jetons et les politiques du coffre-fort. Ce champ est disponible uniquement si *Token* est sélectionné comme type d'authentification du coffre-fort.
Vault authentication role ID	oui		Indiquez l'ID de rôle de l'AppRole. Consultez la documentation HashiCorp pour obtenir des informations sur les AppRoles. Ce champ est disponible uniquement si *AppRole* est sélectionné comme type d'authentification du coffre-fort.
Vault authentication secret ID	oui		Indiquez l'ID secret attribué à l'AppRole. Consultez la documentation HashiCorp pour obtenir des informations sur les AppRoles. Ce champ est disponible uniquement si *AppRole* est sélectionné comme type d'authentification du coffre-fort.

Récupération des valeurs des macros utilisateur

Pour utiliser HashiCorp Vault pour stocker les valeurs des macros utilisateur *Vault secret*, assurez-vous que :

- Zabbix serveur/proxy est configuré pour fonctionner avec HashiCorp Vault.
- Le paramètre *Vault provider* dans *Administration > General > Other* est défini sur "HashiCorp Vault" (par défaut).



Note:
Zabbix serveur (et Zabbix proxy, si configuré) nécessite un accès aux valeurs des macros *Vault secret* depuis le coffre-fort. L'interface Zabbix n'a pas besoin d'un tel accès.

La valeur de la macro doit contenir un chemin de référence (au format `path:key`, par exemple `macros:password`). Le jeton d'authentification spécifié lors de la configuration de Zabbix serveur/proxy (via le paramètre `VaultToken`) doit fournir un accès en lecture seule à ce chemin.

Consultez *Vault secret macros* pour des informations détaillées sur le traitement des valeurs de macro par Zabbix.

Syntaxe du chemin

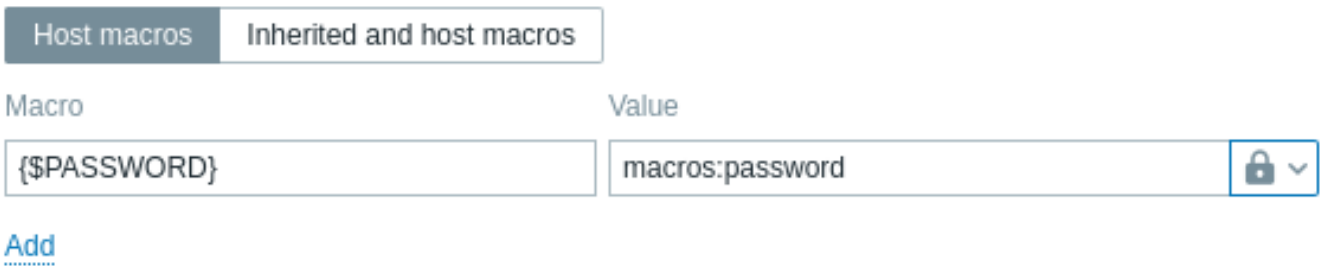
Les symboles barre oblique ("/") et deux-points (":") sont réservés.

Une barre oblique ne peut être utilisée que pour séparer un point de montage d'un chemin (par exemple, *secret/zabbix*, où le point de montage est "secret" et le chemin est "zabbix"). Dans le cas des macros Vault, un deux-points ne peut être utilisé que pour séparer un chemin/requête d'une clé.

Il est possible d'encoder en URL les symboles barre oblique et deux-points s'il est nécessaire de créer un point de montage dont le nom contient une barre oblique (par exemple, *foo/bar/zabbix*, où le point de montage est "foo/bar" et le chemin est "zabbix", peut être encodé en "foo%2Fbar/zabbix") et si le nom d'un point de montage ou un chemin doit contenir un deux-points.

Exemple

1. Dans Zabbix, ajoutez une macro utilisateur {$PASSWORD} de type "Vault secret" et avec la valeur `macros:password`



2. Exécutez les commandes CLI suivantes pour créer le secret requis dans le vault :

```
#### Activer le point de montage "secret/" s'il n'est pas déjà activé ; notez que "kv-v2" doit être utilis
vault secrets enable -path=secret/ kv-v2

#### Ajouter un nouveau secret avec la clé "password" sous le point de montage "secret/" et le chemin "zab
vault kv put -mount=secret zabbix/macros password=<password>

#### Vérifier que le secret a bien été ajouté.
vault kv get secret/zabbix/macros

#### Enfin, tester avec Curl ; notez que "data" doit être ajouté manuellement après le point de montage et
curl --header "X-Vault-Token: <VaultToken>" https://127.0.0.1:8200/v1/secret/data/zabbix/macros
```

3. En conséquence, Zabbix résoudra la macro {$PASSWORD} en la valeur : <password>

Mise à jour d'une configuration existante

Pour mettre à jour une configuration existante afin de récupérer des secrets depuis un HashiCorp Vault :

1. Mettez à jour les paramètres du fichier de configuration du serveur Zabbix ou du proxy, comme décrit dans la section *Identifiants de la base de données*.
2. Mettez à jour les paramètres du fichier de configuration de l'interface Zabbix :

Avec un jeton Vault :

```
$DB['VAULT']                    = 'HashiCorp';
$DB['VAULT_URL']                = 'https://localhost:8200';
$DB['VAULT_DB_PATH']            = 'database';
$DB['VAULT_TOKEN']              = 'hvs.CAESIIG_PILmULFYOsEyWHxkZ2mF2a8VPKNLE8eHqd4autYGGh4KHGh2cy5aeTY0NFN
$DB['VAULT_PREFIX']             = '/v1/secret/data/zabbix/';
```

Ou avec la méthode Vault AppRole :

```
$DB['VAULT']                    = 'HashiCorp';
$DB['VAULT_URL']                = 'https://localhost:8200';
$DB['VAULT_DB_PATH']            = 'database';
$DB['VAULT_APP_ROLE_ID']        = '6f1567f1-538d-7d70-e8e4-5d3e59471b2e';
$DB['VAULT_APP_SECRET_ID']      = '89dac8b1-f7d6-96a7-2ecf-f94d2de3d6f1';
$DB['VAULT_PREFIX']             = '/v1/secret/data/zabbix/';
```

3. Configurez les macros utilisateur comme décrit dans la section *Valeurs des macros utilisateur*, si nécessaire.

Pour mettre à jour une configuration existante afin de récupérer des secrets depuis un CyberArk Vault, consultez la section *Configuration CyberArk*.

Dépannage

1. Dans le fichier journal du serveur/proxy Zabbix :

```
237839:20260729:130916.425 Vault token renew started to fail: cannot login into HashiCorp vault with AppRo
```

Cause probable : HashiCorp Vault a été (re)démarré, mais n'a pas été déverrouillé (obligatoire dans la configuration par défaut, non requis si Auto Unseal est configuré).

2. Dans le fichier journal du serveur/proxy Zabbix :

```
235545:20260729:125617.944 Vault token renew started to fail: cannot login into HashiCorp vault with AppRo
```

ou

```
235545:20260729:125718.537 Vault token renew failed: cannot login into HashiCorp vault with AppRole method
```

Cause probable : VaultAppSecretID a expiré ou a atteint sa limite d'utilisation dans HashiCorp vault.

Confiance du certificat TLS de Google Chrome

Vue d'ensemble

Cette page fournit des étapes de configuration de Zabbix et des exemples de configuration pour utiliser Google Chrome avec l'interface Zabbix ou des points de terminaison zabbix-web-service sécurisés par un certificat auto-signé ou une autorité de certification privée.

Ces instructions supposent que le serveur web cible est déjà configuré pour HTTPS. Pour configurer TLS sur l'interface Zabbix, voir Connexion sécurisée à l'interface.

Configuration

Google Chrome sur Linux utilise une base de données de certificats NSS par utilisateur pour les certificats de confiance. Pour que Chrome fasse confiance à un certificat auto-signé ou à une autorité de certification privée, ajoutez les certificats requis à la base de données NSS du compte utilisateur qui exécute Chrome.

1. Installez les paquets requis.

Pour Debian/Ubuntu :

```
sudo apt install ca-certificates libnss3-tools
```

Pour les systèmes basés sur RHEL :

```
sudo dnf install ca-certificates nss-tools
```

Ou :

```
sudo yum install ca-certificates nss-tools
```

2. Préparez les fichiers de certificat.

Pour chaque certificat nécessaire : créez un fichier `.crt` au format PEM pour le certificat auto-signé, ou créez des fichiers `.crt` distincts au format PEM pour le certificat de l'autorité de certification racine et, si nécessaire, pour chaque certificat intermédiaire.

3. Créez le répertoire de la base de données NSS.

Utilisez le répertoire personnel de l'utilisateur qui exécute Chrome.

Par exemple :

```
sudo mkdir -p /var/lib/zabbix/.pki/nssdb
sudo -u zabbix certutil -N -d sql:/var/lib/zabbix/.pki/nssdb
sudo chown -R zabbix:zabbix /var/lib/zabbix/.pki/nssdb
```

Si Chrome sur le système cible utilise un autre emplacement pour la base de données NSS, utilisez ce répertoire à la place.

Assurez-vous que les fichiers de certificat sont lisibles par l'utilisateur qui exécute Google Chrome. Si nécessaire, copiez-les vers un emplacement accessible à cet utilisateur.

4. Importez les certificats.

Si vous utilisez un certificat auto-signé :

```
sudo -u zabbix certutil -d sql:/var/lib/zabbix/.pki/nssdb \
  -A -t "P,," \
  -n "Zabbix self-signed certificate" \
  -i /path/to/self-signed.crt
```

Si vous utilisez une autorité de certification privée, importez le certificat de l'autorité de certification racine dans la base de données :

```
sudo -u zabbix certutil -d sql:/var/lib/zabbix/.pki/nssdb \
  -A -t "C,," \
  -n "Zabbix root certificate authority" \
  -i /path/to/root-ca.crt
```

Si des certificats intermédiaires sont utilisés, importez chacun d'eux séparément :

```
sudo -u zabbix certutil -d sql:/var/lib/zabbix/.pki/nssdb \
  -A -t ",," \
  -n "Zabbix intermediate certificate" \
  -i /path/to/intermediate-ca.crt
```

5. Vérifiez la base de données de certificats.

```
sudo -u zabbix certutil -d sql:/var/lib/zabbix/.pki/nssdb -L
```

6. Testez Chrome.

Google Chrome utilise la base de données de certificats du compte utilisateur qui le lance. Si le compte utilisateur ne dispose pas d'un répertoire personnel inscriptible, configurez un répertoire personnel et des répertoires XDG inscriptibles avant d'exécuter le test.

Exemple (remplacez `google-chrome` par `chromium` si Chromium est utilisé.) :

```
sudo -u zabbix env \
  HOME=/var/lib/zabbix-home \
  XDG_CONFIG_HOME=/var/lib/zabbix-home/.config \
  XDG_DATA_HOME=/var/lib/zabbix-home/.local/share \
  XDG_CACHE_HOME=/var/lib/zabbix-home/.cache \
  google-chrome --headless --disable-gpu \
  --user-data-dir=/var/lib/zabbix-home/chrome-profile \
  --dump-dom https://FQDN_OF_ZABBIX_SERVER
```

Si la configuration est correcte, la commande renvoie le code HTML de la page demandée.

Si Chrome ne fait pas confiance au certificat, la commande renvoie une page d'erreur de certificat contenant des messages tels que `NET::ERR_CERT_AUTHORITY_INVALID`.

Dépannage

Error	Possible cause
NET::ERR_CERT_AUTHORITY_INVALID	Le certificat a été importé dans le mauvais profil utilisateur, le certificat de l'autorité de certification racine est manquant, ou un certificat intermédiaire est manquant.
NET::ERR_CERT_COMMON_NAME_INVALID	Le certificat ne correspond pas au nom d'hôte utilisé dans l'URL du navigateur.
NET::ERR_CERT_DATE_INVALID	Le certificat a expiré, n'est pas encore valide, ou l'horloge système est incorrecte.
Chrome affiche toujours un avertissement de certificat après l'importation	Chrome utilise une base de données NSS différente pour l'utilisateur actuel, ou les indicateurs de confiance du certificat sont incorrects.

Bonnes pratiques de sécurité

Vue d'ensemble Cette section présente les bonnes pratiques pour configurer Zabbix de manière sécurisée.

Les pratiques décrites dans cette section ne sont pas nécessaires au fonctionnement de Zabbix, mais elles sont recommandées pour renforcer la sécurité du système.

Pour connaître les bonnes pratiques visant à sécuriser l'application mobile Zabbix et l'adaptateur Zabbix bridge, consultez Accès mobile sécurisé.

Encodage UTF-8 UTF-8 est le seul encodage pris en charge par Zabbix. Il est connu pour fonctionner sans aucune faille de sécurité. Les utilisateurs doivent savoir qu'il existe des problèmes de sécurité connus lors de l'utilisation de certains autres encodages.

Chemins de l'installateur Windows Lors de l'utilisation des installateurs Windows, il est recommandé d'utiliser les chemins par défaut fournis par l'installateur. L'utilisation de chemins personnalisés sans autorisations appropriées pourrait compromettre la sécurité de l'installation.

Macros dans les scripts globaux définis par l'utilisateur Pour renforcer la sécurité, il est recommandé d'utiliser des fonctions de macro plutôt que des macros simples dans les scripts globaux définis par l'utilisateur, car les macros ne sont pas automatiquement échappées.

Avis de sécurité Zabbix et base de données CVE Voir *Avis de sécurité Zabbix et base de données CVE*.

Modèles d'e-mails HTML Lors de la création ou de la modification de modèles de messages utilisés pour les e-mails HTML, entourez toujours chaque macro de la fonction de macro `htmlencode()`. Par exemple :

```
<b>Problem started</b> at {{EVENT.TIME}.htmlencode()} on {{EVENT.DATE}.htmlencode()}<br><b>Problem name:</
{{EVENT.NAME}.htmlencode()}<br><b>Host:</b> {{HOST.NAME}.htmlencode()}
<br><b>Severity:</b>
{{EVENT.SEVERITY}.htmlencode()}<br><b>Operational
data:</b> {{EVENT.OPDATA}.htmlencode()}<br><b>Original problem ID:</b> {{EVENT.ID}.htmlencode()}<br>{{TRIG
```

L'utilisation de `htmlencode()` garantit que tous les caractères HTML présents dans les valeurs des macros sont échappés et empêche l'injection de code HTML dans les notifications (par exemple, l'insertion par un attaquant d'un lien malveillant ou d'hameçonnage dans une notification).

Remarque : les messages électroniques HTML par défaut fournis par Zabbix appliquent déjà `htmlencode()` aux macros. Ces recommandations s'appliquent lors de la modification de modèles existants ou de la création de nouveaux modèles : vérifiez que les macros sont encodées avant d'utiliser un modèle pour envoyer un e-mail HTML.

Interface

Présentation générale Pour accéder facilement à Zabbix depuis n'importe où et depuis n'importe quelle plateforme, une interface web est fournie.

Note:
Si vous utilisez plusieurs instances de l'interface, assurez-vous que les paramètres régionaux et les bibliothèques (LDAP, SAML, etc.) sont installés et configurés de manière identique sur toutes les interfaces.

Aide de l'interface Un lien d'aide est disponible dans les sections et les formulaires de l'interface Zabbix, dans le coin supérieur droit. Il renvoie directement aux sections correspondantes de la documentation.

Pour les champs de formulaire accompagnés d'une icône d'aide, le survol de l'icône affiche l'indication correspondante.

Mode kiosque Pour les sections de l'interface disposant du bouton *Mode kiosque* (situé dans le coin supérieur droit), cliquez sur ce bouton pour passer la section en mode kiosque. En mode kiosque, seul le contenu de la section est affiché.

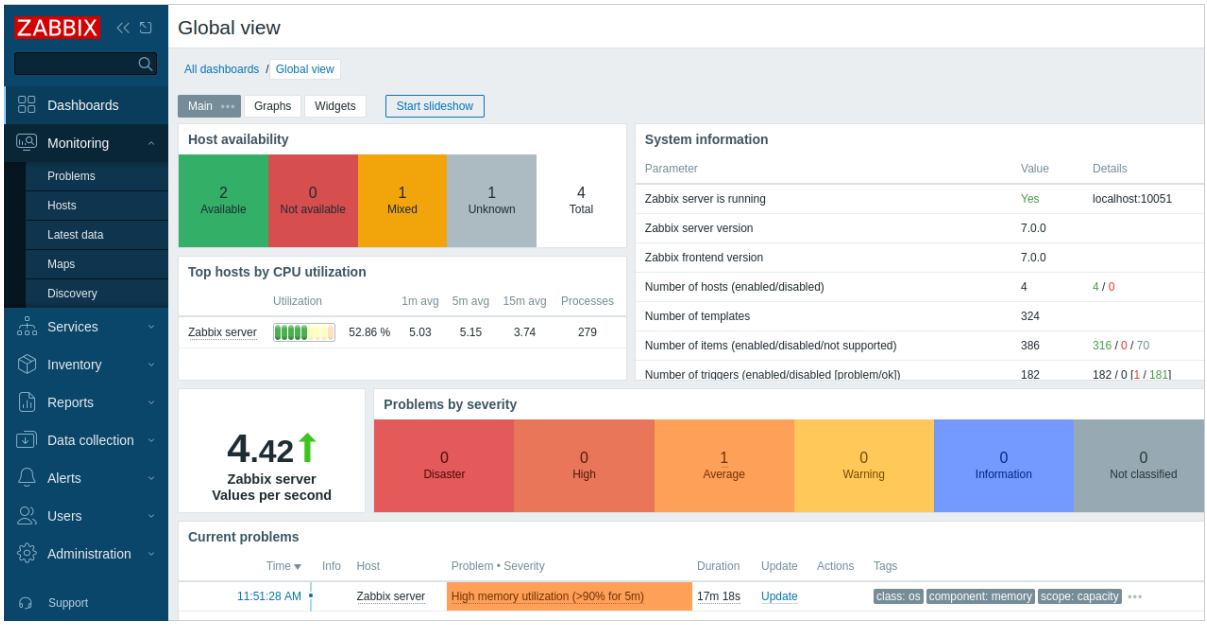
Pour quitter le mode kiosque, placez le curseur sur le contenu et cliquez sur le bouton *Vue normale* qui apparaît dans le coin supérieur droit.

Menu

Vue d'ensemble

Un menu vertical dans une barre latérale permet d'accéder à différentes sections de l'interface web de Zabbix.

Le menu est bleu foncé dans le thème par défaut.

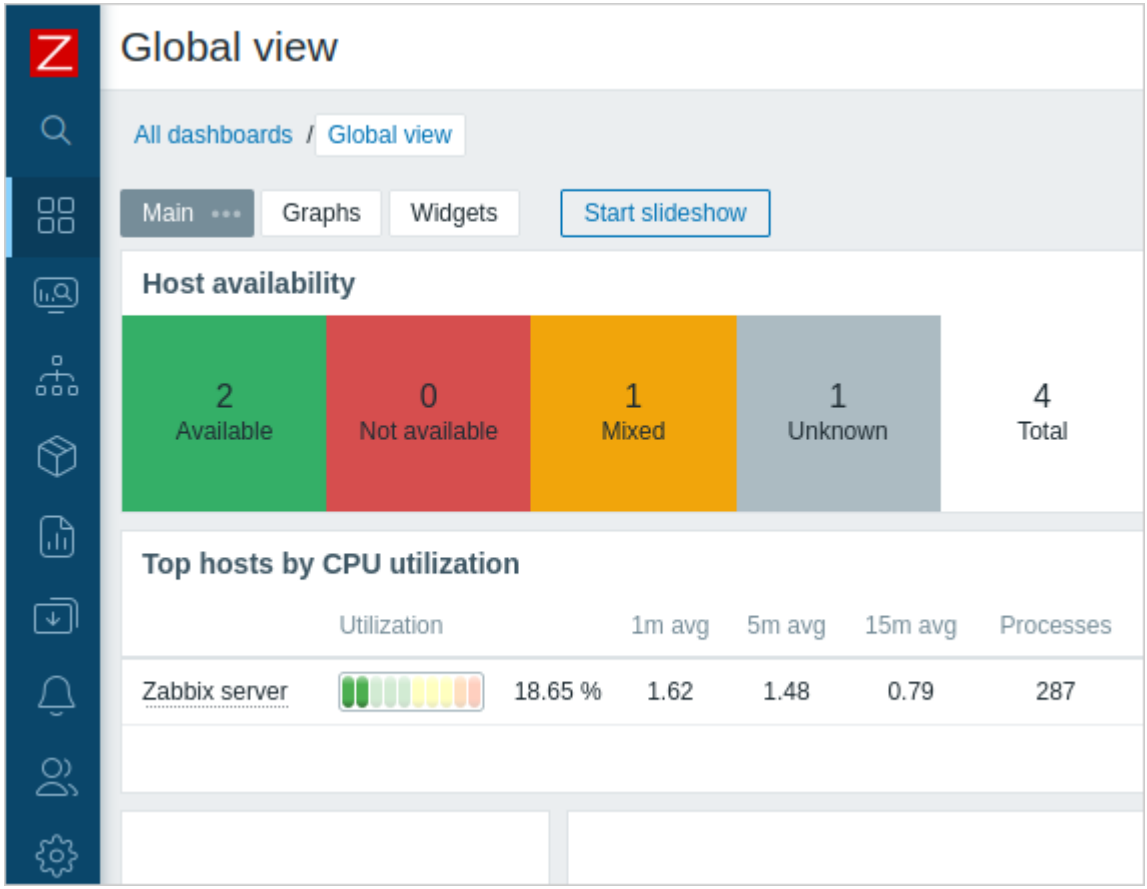


Travailler avec le menu

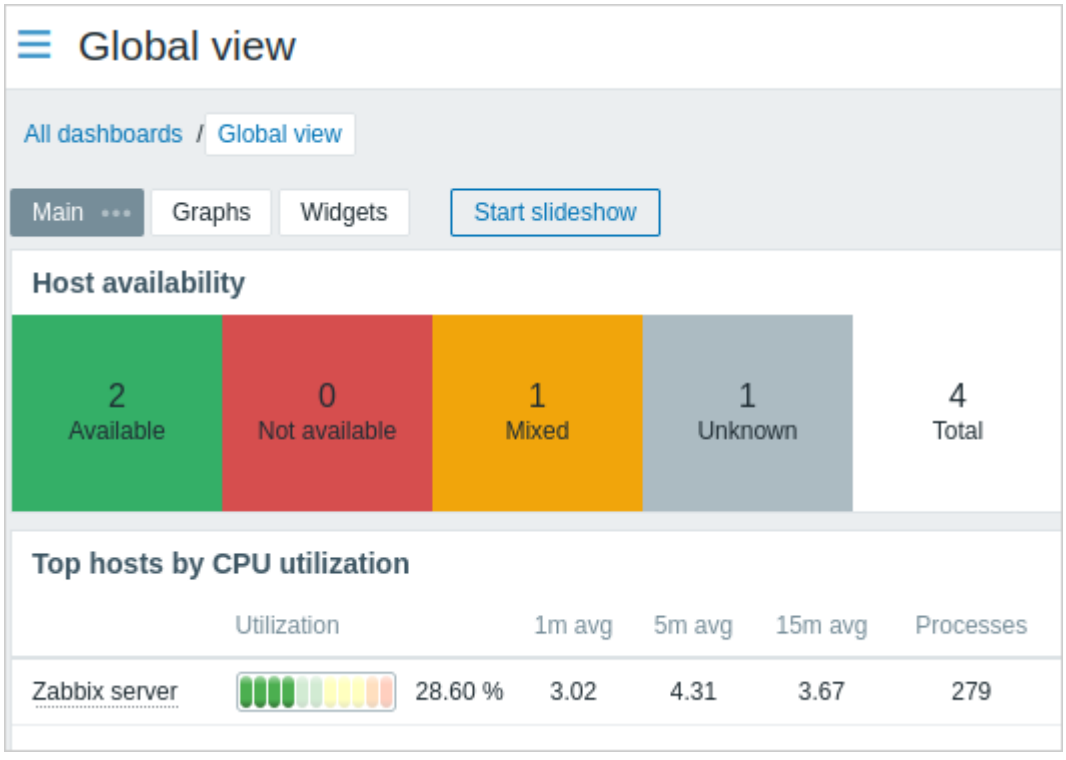
Une zone de recherche globale se trouve sous le logo Zabbix.

Le menu peut être réduit ou complètement masqué :

- Pour le réduire, cliquez sur à côté du logo Zabbix. Dans le menu réduit, seules les icônes sont visibles.



- Pour le masquer, cliquez sur à côté du logo Zabbix. Dans le menu masqué, tout est caché.



Menu réduit

Lorsque le menu est réduit à ses seules icônes, le menu complet réapparaît dès que le curseur de la souris est placé dessus. Notez qu'il réapparaît par-dessus le contenu de la page ; pour déplacer le contenu de la page vers la droite, vous devez cliquer sur le bouton d'agrandissement. Si le curseur de la souris est de nouveau placé en dehors du menu complet, celui-ci se réduira à nouveau après deux secondes.

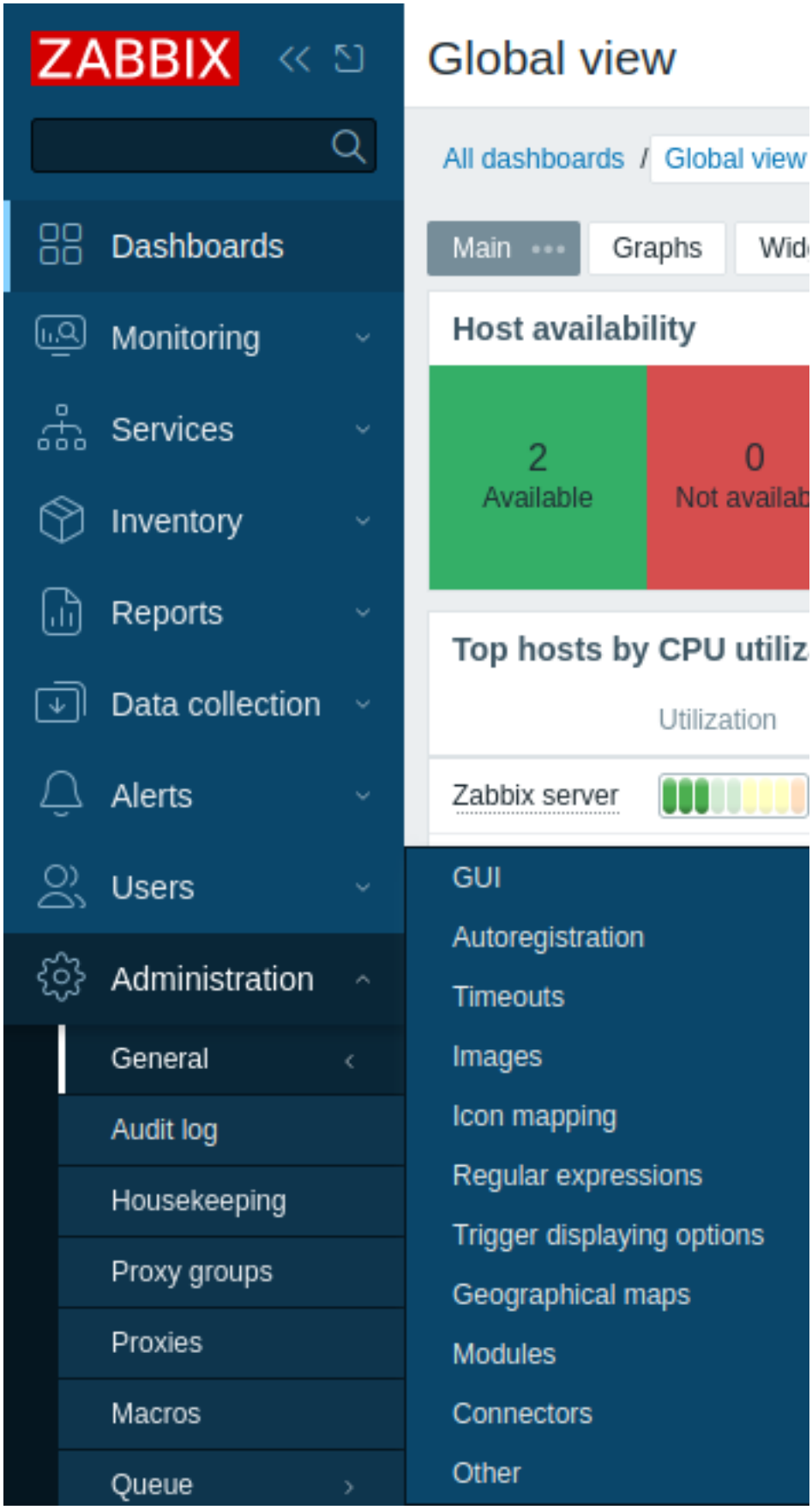
Vous pouvez également faire réapparaître complètement un menu réduit en appuyant sur la touche Tab. Appuyer plusieurs fois sur la touche Tab permet de placer le focus sur l'élément de menu suivant.

Menu masqué

Même lorsque le menu est complètement masqué, un menu complet n’est qu’à un clic de souris, en cliquant sur l’icône burger . Notez qu’il réapparaît au-dessus du contenu de la page ; pour déplacer le contenu de la page vers la droite, vous devez afficher à nouveau le menu en cliquant sur le bouton d’affichage de la barre latérale.

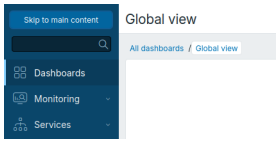
Niveaux de menu

Le menu comporte jusqu’à trois niveaux.



Accessibilité: passer la navigation

Le lien *Passer au contenu principal* permet aux utilisateurs de clavier et de lecteur d'écran de sauter la navigation répétitive de la barre latérale et d'accéder directement au contenu principal.



Il reste masqué lors d'une utilisation normale et n'apparaît que lorsqu'il reçoit le focus, par exemple lorsque la touche Tab est enfoncée après le chargement de la page ou lorsqu'il est utilisé avec un lecteur d'écran.

Menus contextuels

En plus du menu principal, Zabbix fournit des menus contextuels pour hôte, élément et événement pour un accès rapide aux entités fréquemment utilisées, telles que les dernières valeurs, le graphique simple, le formulaire de configuration, les scripts associés ou les liens externes.

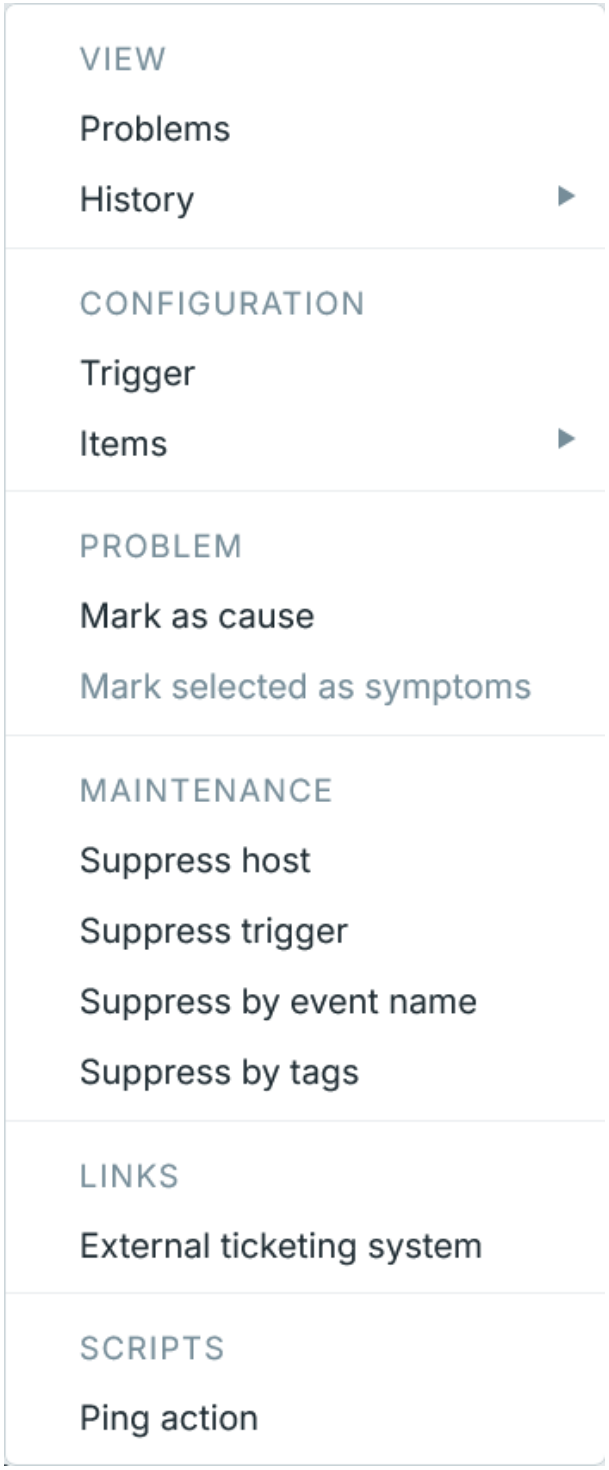
Les menus contextuels sont accessibles en cliquant sur le nom de l'hôte, de l'élément ou du problème/déclencheur dans les emplacements pris en charge.

Menu d'événements

Vue d'ensemble

Le menu des événements contient des raccourcis vers des actions ou des sections de l'interface fréquemment nécessaires pour un événement.

Le menu des événements peut être ouvert en cliquant sur le nom de l'événement.



Contenu

Le menu contextuel de l'événement comporte plusieurs sections : *Affichage*, *Actions*, *Configuration*, *Problème*, *Maintenance*, *Liens* et *Scripts*. Pour les entités qui ne sont pas configurées, les liens sont désactivés et affichés en gris. Les sections *Scripts* et *Liens* sont affichées si leurs entités sont configurées.

La section *Affichage* contient des liens vers :

- **Problèmes** - ouvre la liste des problèmes non résolus du déclencheur sous-jacent ;
- **Historique** - mène au graphique/historique de l'élément dans *Dernières données* pour le ou les éléments sous-jacents. Si un déclencheur utilise plusieurs éléments, des liens seront disponibles pour chacun d'eux.

La section *Actions* est disponible uniquement dans les widgets *Vue d'ensemble des déclencheurs*. Elle contient un lien vers :

- **Mettre à jour le problème** - ouvre l'écran de mise à jour du problème.

La section *Configuration* contient des liens vers la configuration des éléments suivants :

- **Déclencheur** à l'origine du problème ;
- **Éléments** utilisés dans l'expression du déclencheur.

Note:
Notez que la section de configuration est disponible uniquement pour les utilisateurs Admin et Super admin.

La section *Problème* contient les options suivantes :

- **Marquer comme cause** - marque le problème comme cause ;
- **Marquer la sélection comme symptômes** - marque les problèmes sélectionnés comme symptômes de ce problème.

La section *Maintenance* contient des options permettant de démarrer une maintenance ad hoc :

- **Supprimer l'hôte** - démarre une période de maintenance pour l'hôte ;
- **Supprimer le déclencheur** - démarre une période de maintenance pour le déclencheur ;
- **Supprimer par nom d'événement** - démarre une période de maintenance pour le nom de l'événement ;
- **Supprimer par étiquettes** - démarre une période de maintenance pour les événements comportant les étiquettes correspondantes.

Lorsque vous sélectionnez l'option correspondante, le formulaire de configuration d'une période de maintenance s'ouvre.

Une période de maintenance ponctuelle d'une heure (configurable dans les paramètres et le profil de l'utilisateur), commençant à partir de maintenant, est présélectionnée. L'hôte, le déclencheur, le nom de l'événement ou les étiquettes correspondants sont également présélectionnés. Le nom de la maintenance est prérempli au format `Ad-hoc: <event name>`.

La section *Liens* contient des liens permettant :

- d'accéder à une URL de déclencheur configurée ;
- d'accéder aux liens personnalisés configurés dans les Scripts globaux (avec la portée « Action d'événement manuelle » et le type « URL ») ;
- d'accéder à un ticket externe configuré pour le problème (voir l'option *Inclure l'entrée dans le menu de l'événement* lors de la configuration des webhooks).

La section *Scripts* contient des liens permettant d'exécuter un script global (avec la portée *Action d'événement manuelle*). Cette fonctionnalité peut être utile pour exécuter des scripts utilisés pour gérer les tickets de problème dans des systèmes externes.

Emplacements pris en charge

Le menu contextuel de l'événement est accessible en cliquant sur un nom de problème ou d'événement dans différentes sections de l'interface, par exemple :

- Tableaux de bord widgets, tels que le widget *Problems*, le widget *Trigger overview*, etc.
- *Monitoring* > *Problems*
- *Monitoring* > *Problems* > Détails de l'événement
- *Reports* > *Top 100 triggers* (les scripts globaux et l'accès au ticket externe ne sont pas pris en charge à cet emplacement)

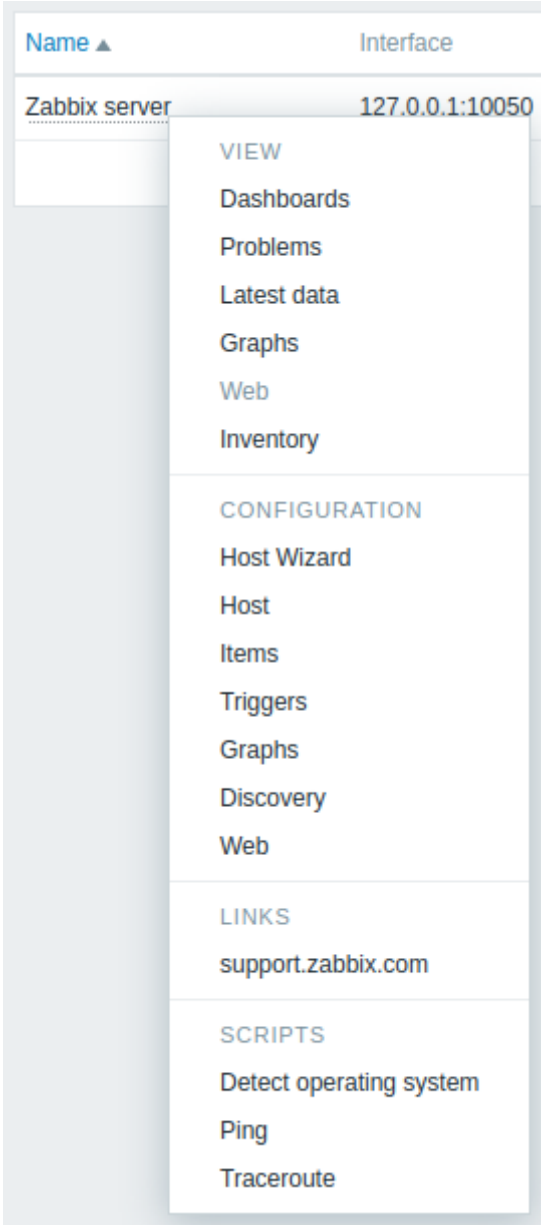
Menu de l'hôte

Vue d'ensemble

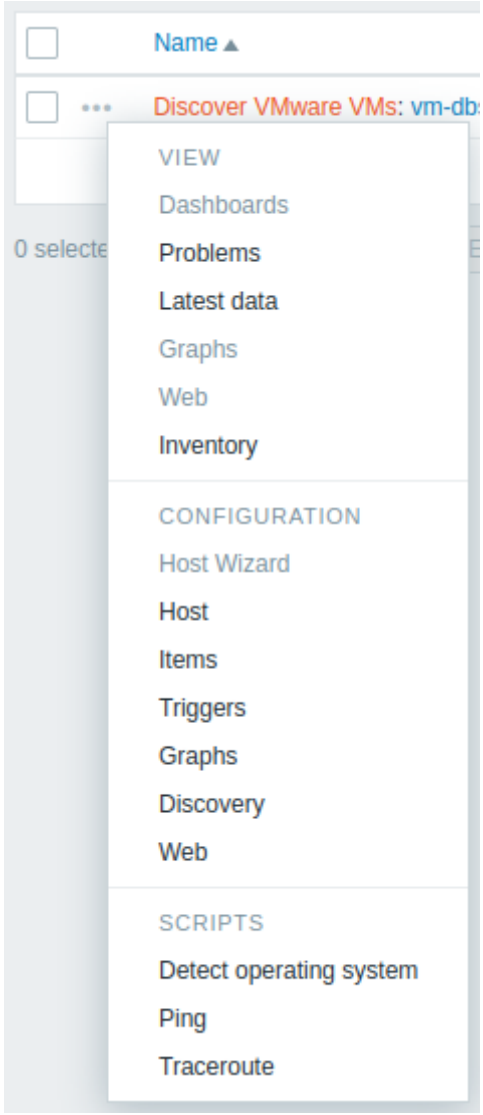
Le menu de l'hôte contient des raccourcis vers des actions ou des sections de l'interface fréquemment nécessaires pour un hôte.

Le menu de l'hôte peut être ouvert en cliquant sur le nom de l'hôte ou sur l'icône à trois points, selon la section de l'interface, par exemple :

Monitoring > *Latest data*



Data collection > *Hosts*



Contenu

Le menu contextuel de l'hôte comporte quatre sections : *Affichage*, *Configuration*, *Liens* et *Scripts*. Pour les entités qui ne sont pas configurées, les liens sont désactivés et affichés en gris. Les sections *Scripts* et *Liens* sont affichées si leurs entités sont configurées.

La section *Affichage* contient des liens vers :

- **Tableaux de bord** - ouvre les widgets et les graphiques.
- **Problèmes** - ouvre la section *Problèmes* avec la liste des problèmes non résolus du déclencheur sous-jacent.
- **Dernières données** - ouvre la section *Dernières données* avec la liste de toutes les dernières données de l'hôte actuel.
- **Graphiques** - ouvre les graphiques simples de l'hôte actuel.
- **Web** - ouvre le lien vers les scénarios Web configurés.
- **Inventaire** - ouvre le lien vers l'inventaire de l'hôte actuel.

La section *Configuration* contient des liens vers :

- **Assistant de configuration de l'hôte** - ouvre l'Assistant de configuration de l'hôte pour l'hôte actuel (désactivé pour les hôtes découverts).
- **Hôte** - formulaire de configuration de l'hôte actuel.
- **Éléments** - liste des éléments de l'hôte actuel.
- **Déclencheurs** - liste des déclencheurs de l'hôte actuel.
- **Graphiques** - graphiques simples de l'hôte actuel.

- **Découverte** - liste des règles de découverte de bas niveau de l'hôte actuel.
- **Web** - liste des scénarios Web de l'hôte actuel.

> **Note:**
> Notez que la section de configuration est disponible uniquement pour les utilisateurs Admin et Super admin.

La section *Liens* contient des liens permettant :

- d'accéder à une URL de déclencheur configurée.
- d'accéder aux liens personnalisés configurés dans les Scripts globaux (avec la portée *Action manuelle sur l'hôte* et le type « URL »).

La section *Scripts* permet d'exécuter les scripts globaux configurés pour l'hôte actuel. Pour être disponibles dans le menu de l'hôte, ces scripts doivent avoir la portée *Action manuelle sur l'hôte*.

Emplacements pris en charge

Le menu de l'hôte est accessible en cliquant sur le nom d'un hôte dans différentes sections de l'interface, par exemple :

- Tableaux de bord widgets, tels que *Problems*, *Top items*, *Trigger overview*, etc.
- *Monitoring* > *Problems*
- *Monitoring* > *Problems* > Détails de l'événement
- *Monitoring* > *Hosts*
- *Monitoring* > *Hosts* > *Web Monitoring*
- *Monitoring* > *Latest data*
- *Monitoring* > *Maps*
- *Inventory* > *Hosts*
- *Reports* > *Top 100 triggers*

Le menu de l'hôte est accessible en cliquant sur l'icône à trois points dans Data collection > Hosts.

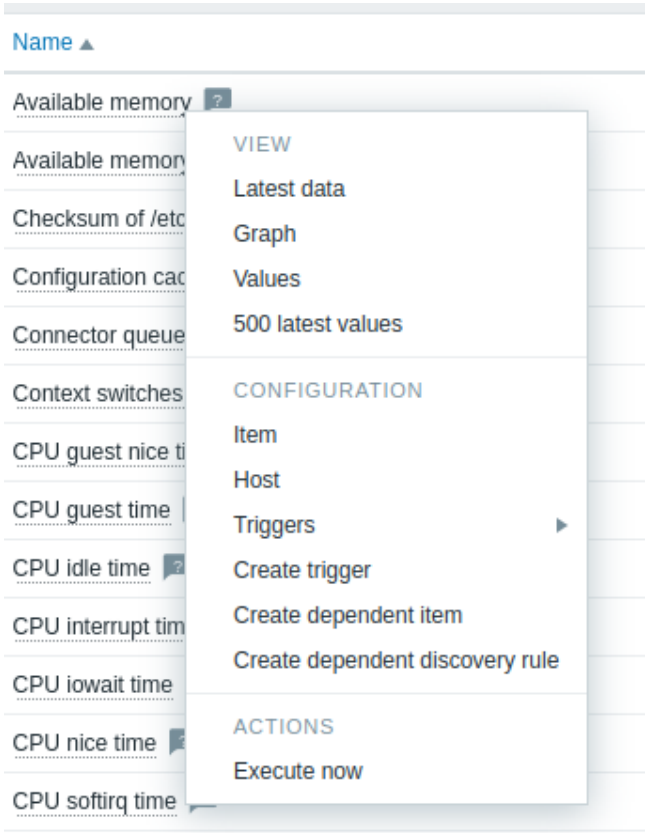
Menu de l'élément

Vue d'ensemble

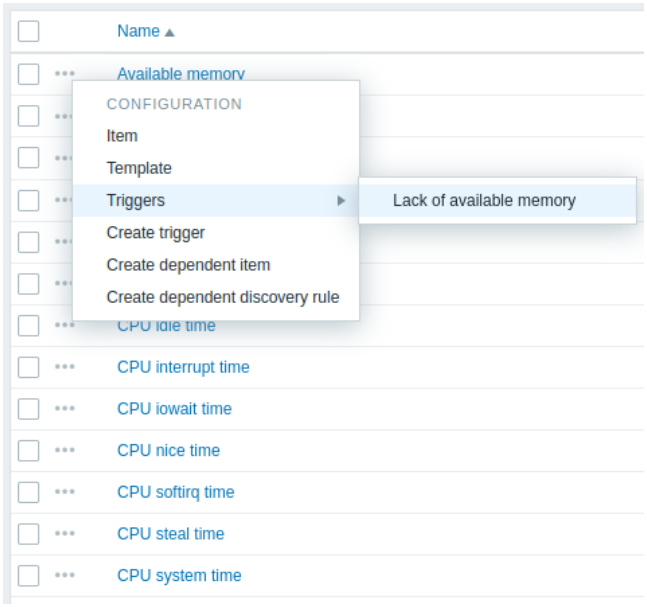
Le menu de l'élément contient des raccourcis vers des actions ou des sections de l'interface fréquemment nécessaires pour un élément.

Le menu de l'élément peut être ouvert en cliquant sur le nom de l'élément ou sur l'icône à trois points, selon la section de l'interface, par exemple :

Monitoring > *Latest data*



Data Collection > *Modèles* > *Items*



Contenu

Le menu de l'élément comporte trois sections : *Affichage*, *Configuration* et *Actions*.

La section *Affichage* contient les options suivantes :

- **Dernières données** - ouvre la section *Dernières données* filtrée par l'hôte et l'élément actuels ;
- **Graphique** - ouvre un graphique simple de l'élément actuel ;
- **Valeurs** - ouvre la liste de toutes les valeurs reçues pour l'élément actuel au cours des 60 dernières minutes ;
- **500 dernières valeurs** - ouvre la liste des 500 dernières valeurs de l'élément actuel.

La section *Configuration* (disponible uniquement pour les utilisateurs de type *Administrateur* et *Super administrateur*) contient les options suivantes :

- **Élément** - ouvre le formulaire de configuration de l'élément actuel ;
- **Modèle** - ouvre le formulaire de configuration du modèle auquel appartient l'élément (disponible uniquement lorsque le menu de l'élément est ouvert depuis *Collecte de données* > *Modèles* > *Éléments*) ;
- **Hôte** - ouvre le formulaire de configuration de l'hôte auquel appartient l'élément ;
- **Déclencheurs** - au survol de la souris, ouvre la liste des déclencheurs de l'élément, le cas échéant ; cliquer sur un déclencheur ouvre son formulaire de configuration ;
- **Créer un déclencheur** - ouvre le formulaire de configuration du déclencheur pour créer un déclencheur pour cet élément ;
- **Créer un élément dépendant** - ouvre le formulaire de configuration de l'élément pour créer un élément dépendant, avec l'élément actuel comme élément maître ;
- **Créer une règle de découverte dépendante** - ouvre le formulaire de configuration de la règle de découverte pour créer une règle de découverte dépendante, avec l'élément actuel comme élément maître.

La section *Actions* contient l'option suivante :

- **Exécuter maintenant** - exécute immédiatement une vérification pour obtenir une nouvelle valeur de l'élément.

Emplacements pris en charge

Le menu de l'élément est accessible en cliquant sur le nom d'un élément dans différentes sections de l'interface, par exemple :

- *Surveillance* > *Dernières données*
- *Collecte de données* > *Hôtes* > *Éléments*
- *Collecte de données* > *Hôtes* > *Règles de découverte* > *Modèles d'éléments*

Le menu de l'élément est accessible en cliquant sur la valeur d'un élément dans le widget de tableau de bord *Top items*.

Sections de l'interface

Structure du menu Le menu de l'interface Zabbix présente la structure suivante :

- Tableaux de bord
- Supervision
  - Problèmes
  - Hôtes
  - Dernières données
  - Cartes
  - Découverte
- APM
  - Traces
  - Métriques
  - Journaux
- Services
  - Services
  - SLA
  - Rapport SLA
- Inventaire
  - Vue d'ensemble
  - Hôtes
- Rapports
  - Informations système
  - Rapports planifiés
  - Rapport de disponibilité
  - 100 principaux déclencheurs
  - Journal d'audit
  - Journal des actions
  - Notifications
- Collecte de données
  - Groupes de modèles
  - Groupes d'hôtes
  - Modèles
  - Hôtes
  - Maintenance
  - Corrélation des événements
  - Découverte
- Alertes
  - Actions
    - Actions de déclencheur
    - Actions de service
    - Actions de découverte
    - Actions d'enregistrement automatique
    - Actions internes
  - Types de média
  - Scripts
- Utilisateurs
  - Groupes d'utilisateurs
  - Rôles utilisateur
  - Utilisateurs
  - Jetons API
  - Authentification
  - Appareils
- Administration
  - Général
    - Interface graphique
    - Enregistrement automatique
    - Délais d'expiration

    - Images
    - Mappage des icônes
    - Expressions régulières
    - Options d'affichage des déclencheurs
    - Cartes géographiques
    - Modules
    - Connecteurs
    - Autre
- Journal d'audit
- Entretien
- Groupes de proxy
- Proxys
- Macros
- File d'attente
    - Vue d'ensemble de la file d'attente
    - Vue d'ensemble de la file d'attente par proxy
    - Détails de la file d'attente

Tableaux de bord

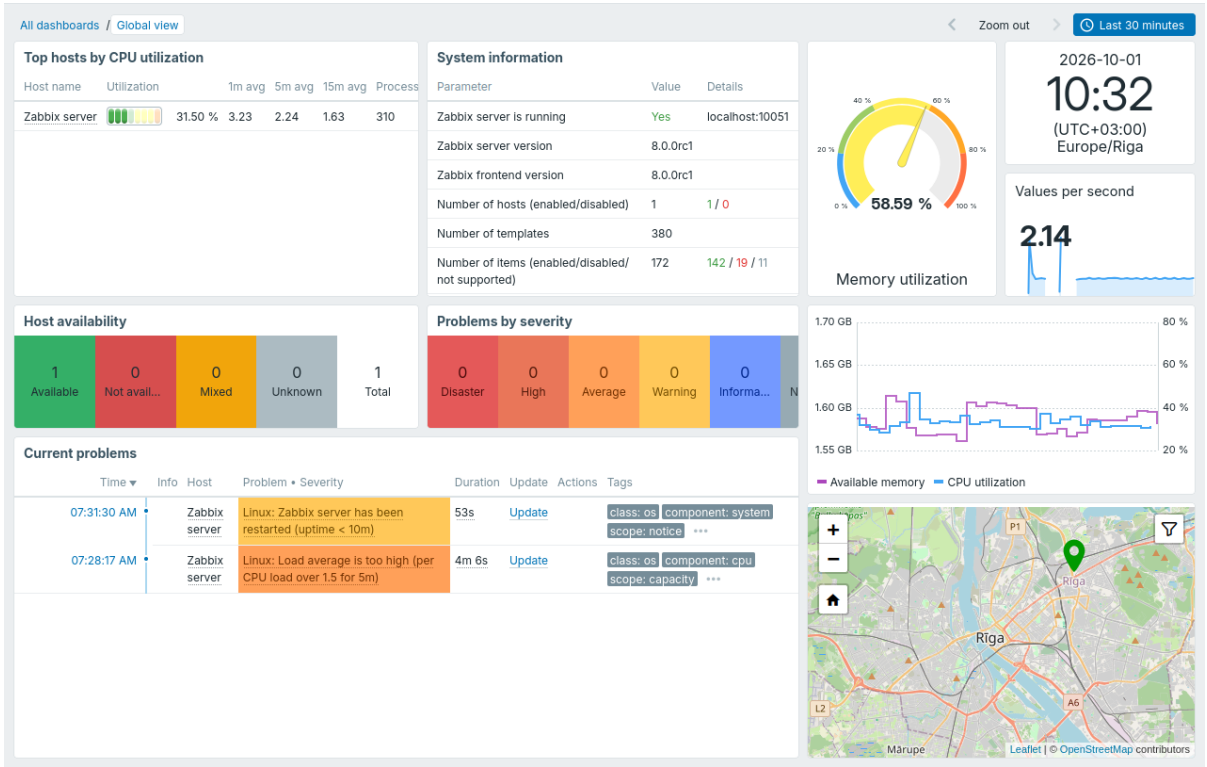
Les tableaux de bord vous donnent une vue d'ensemble de votre environnement surveillé. Vous pouvez les créer et les personnaliser selon vos besoins.

Les éléments constitutifs sont les widgets : un graphique, une carte, une liste des problèmes récents, etc., chacun étant conçu pour afficher un type particulier d'informations.

Les widgets peuvent être organisés sur plusieurs pages de tableau de bord. Ces pages peuvent être affichées sous forme de diaporama. Vous pouvez également accéder à n'importe quelle page en cliquant sur son onglet.

Vous pouvez créer plusieurs tableaux de bord, mais un seul est affiché à la fois.

Zabbix est fourni avec plusieurs tableaux de bord par défaut. L'un d'eux est le tableau de bord *Vue globale*, qui affiche l'état et la santé de vos hôtes surveillés, y compris du serveur Zabbix :

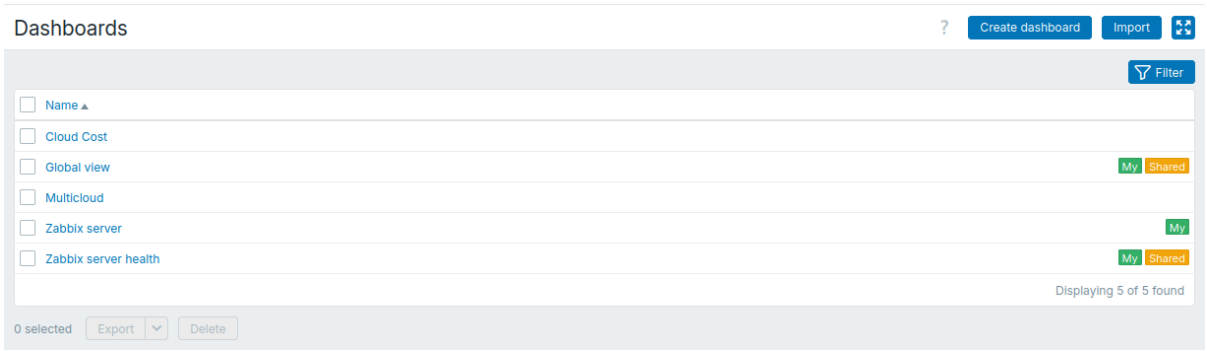


Vous trouverez d'autres tableaux de bord dans le référentiel des tableaux de bord. Ces tableaux de bord sont conçus pour fonctionner avec les modèles Zabbix existants et peuvent être téléchargés et importés dans votre environnement Zabbix. Une description détaillée de chaque tableau de bord, y compris les prérequis, est disponible dans le fichier `README` du tableau de bord.

Liste des tableaux de bord

Pour ouvrir la liste des tableaux de bord, vous pouvez :

- Cliquer sur *Tableaux de bord* dans le menu.
- Cliquer sur *Tous les tableaux de bord* lors de la consultation d'un tableau de bord pour ouvrir la liste des tableaux de bord.



L'accès au menu *Tableaux de bord* dépend du rôle utilisateur et de l'autorisation *Tableau de bord* sous *Accès aux éléments de l'interface*.

Depuis cette page, vous pouvez : - Créer des tableaux de bord. - Supprimer des tableaux de bord. Vous pouvez supprimer un tableau de bord en sélectionnant sa case à cocher, puis en cliquant sur le bouton *Supprimer*. - Importer des tableaux de bord. - Exporter des tableaux de bord.

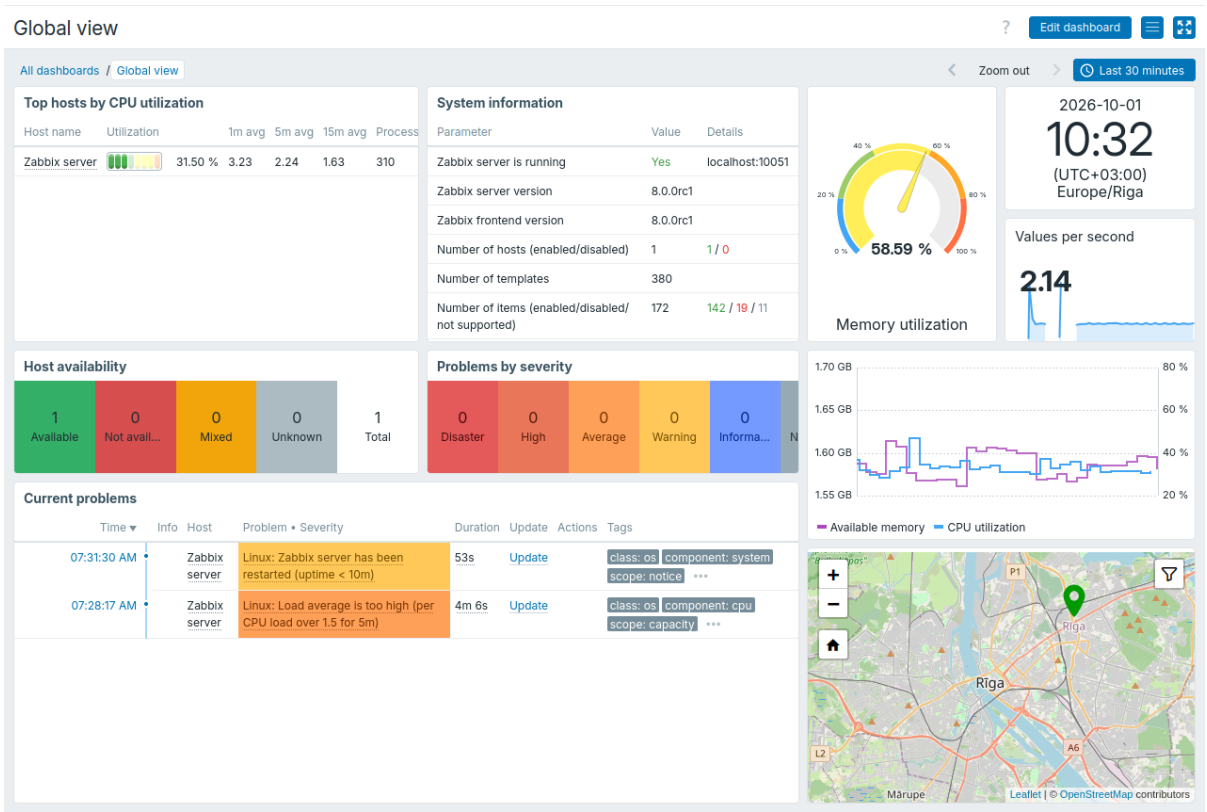
Les tableaux de bord comportent des balises indiquant leur propriétaire et leurs paramètres de partage :

- *Mes tableaux de bord* - le tableau de bord vous appartient et est visible uniquement par vous.
- *Mes tableaux de bord + Partagé* - le tableau de bord vous appartient, mais il est public ou partagé avec des utilisateurs ou des groupes d'utilisateurs spécifiques.
- Les tableaux de bord sans balises appartiennent à d'autres utilisateurs, mais sont publics ou partagés avec vous.

Vous pouvez filtrer la liste par nom de tableau de bord et par propriétaire.

Afficher le tableau de bord

Cliquez sur le nom d'un tableau de bord dans la liste des tableaux de bord pour afficher le tableau de bord :



L'interface utilisateur de Zabbix mémorise le dernier tableau de bord ouvert pour la session suivante.

Note:
Les données que vous pouvez voir sur un tableau de bord dépendent de votre accès aux hôtes.

Pour les widgets qui affichent des graphiques, le sélecteur de période situé au-dessus des widgets définit la plage temporelle. Il affiche la plage temporelle actuelle sous forme d'étiquette. Cliquez sur l'étiquette pour ouvrir ou fermer le sélecteur.

Vous pouvez gérer les tableaux de bord comme suit :

- Modifier le tableau de bord.
- Effectuer les actions suivantes depuis le menu *Actions* du tableau de bord :
  - *Partage* - modifier les paramètres de partage du tableau de bord.
  - *Créer* - créer un nouveau tableau de bord.
  - *Cloner* - créer un nouveau tableau de bord en clonant celui actuel. Vous pouvez modifier les propriétés du tableau de bord avant de l'enregistrer.
  - *Supprimer* - supprimer le tableau de bord.
  - *Exporter* - exporter le tableau de bord.
  - *Créer un nouveau rapport* - créer un rapport planifié. Cette option est désactivée si vous n'avez pas la permission de gérer les rapports planifiés.
  - *Afficher les rapports associés* - afficher la liste des rapports existants pour le tableau de bord actuel. Cette option est désactivée s'il n'existe aucun rapport associé ou si vous n'avez pas la permission de gérer les rapports planifiés.
- Passer en mode kiosque.
- Démarrer ou arrêter le diaporama du tableau de bord.

En mode d'affichage du tableau de bord, les options suivantes sont disponibles pour les widgets :

- Modifier le widget.
- Effectuer les actions suivantes depuis le menu *Actions* du widget ... :
  - *Copier* - copier le widget.
  - *Télécharger l'image* - télécharger le widget sous forme d'image PNG. Cette option est disponible pour les widgets qui prennent en charge cette fonctionnalité (voir les pages consacrées à chaque widget).
  - *Intervalle d'actualisation* - sélectionner la fréquence d'actualisation du contenu du widget.

Pour les tableaux de bord comportant plusieurs pages, les actions suivantes sont disponibles dans le menu *Actions* de la page  en mode d'affichage du tableau de bord :

- *Copier* - copier la page.
- *Propriétés* - personnaliser les propriétés de la page, notamment son nom et l'intervalle du diaporama.

Diaporama

Si la case *Démarrer le diaporama* est sélectionnée dans les propriétés du tableau de bord, le diaporama démarre automatiquement lorsque le tableau de bord comporte plusieurs pages. Sinon, vous pouvez démarrer le diaporama en cliquant sur le bouton *Démarrer le diaporama*.



Les pages changent selon les intervalles spécifiés dans les propriétés du tableau de bord et des pages individuelles. Si la propriété *Intervalle du diaporama* est définie pour une page individuelle, l'intervalle spécifié pour cette page est appliqué. Sinon, l'*Intervalle de diaporama par défaut* du tableau de bord est appliqué.

Pour arrêter le diaporama, cliquez sur *Arrêter le diaporama*.

En mode kiosque, utilisez les boutons suivants :

-  pour démarrer le diaporama ;
-  pour arrêter le diaporama ;
-  pour accéder à la page précédente ;
-  pour accéder à la page suivante.

Partager un tableau de bord

Les tableaux de bord peuvent être publics ou privés.

Les tableaux de bord publics sont visibles par tous les utilisateurs. Les tableaux de bord privés sont visibles uniquement par leurs propriétaires et par les utilisateurs ou groupes d'utilisateurs avec lesquels ils sont partagés.

Pour modifier les paramètres de partage d'un tableau de bord, cliquez sur *Partage* dans le menu *Actions* du tableau de bord lorsque vous consultez le tableau de bord :

Paramètre	Description
Type	Sélectionnez le type de tableau de bord : **Privé** - le tableau de bord est visible uniquement par les groupes d'utilisateurs et les utilisateurs sélectionnés. **Public** - le tableau de bord est visible par tous les utilisateurs.
Liste des partages avec des groupes d'utilisateurs	Sélectionnez les groupes d'utilisateurs qui peuvent accéder au tableau de bord. Vous pouvez accorder un accès en lecture seule ou en lecture-écriture.
Liste des partages avec des utilisateurs	Sélectionnez les utilisateurs qui peuvent accéder au tableau de bord. Vous pouvez accorder un accès en lecture seule ou en lecture-écriture.

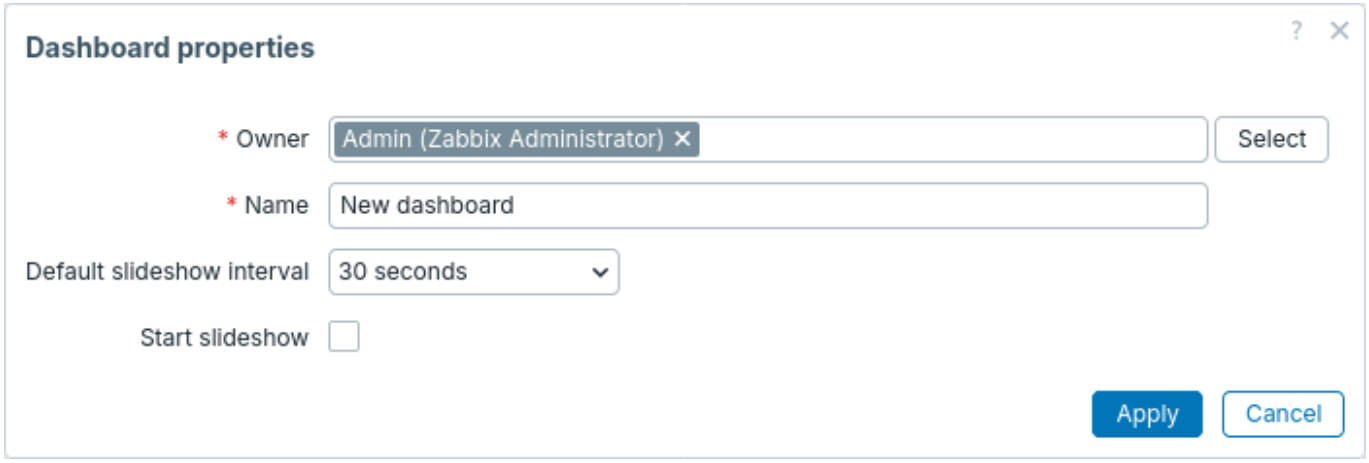
Consultez Autorisations sur les tableaux de bord pour savoir comment les options de partage limitent les actions disponibles pour les utilisateurs.

Créer un tableau de bord

Vous pouvez créer un tableau de bord de deux façons :

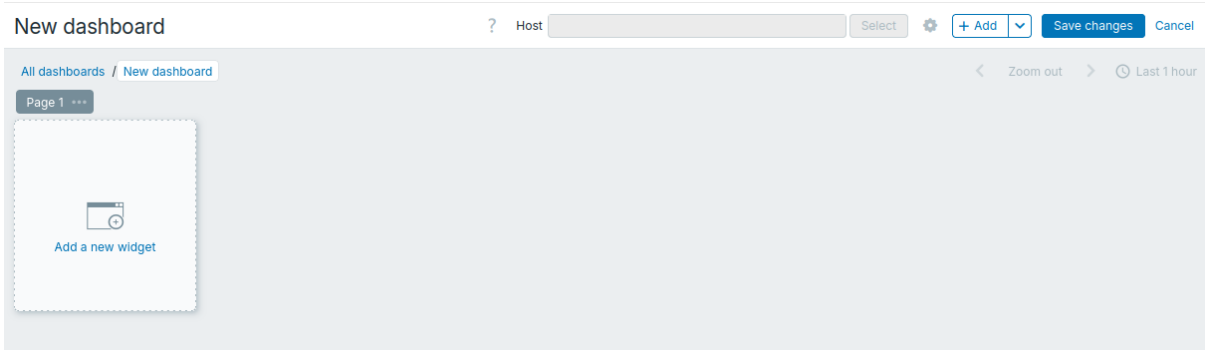
- Dans la liste des tableaux de bord, cliquez sur le bouton *Créer un tableau de bord*.
- En mode affichage du tableau de bord, cliquez sur *Créer* dans le menu *Actions* du tableau de bord.

Indiquez ensuite les propriétés du tableau de bord :



Paramètre	Description
Propriétaire	Sélectionnez l'utilisateur qui sera propriétaire du tableau de bord.
Nom	Saisissez le nom du tableau de bord.
Intervalle par défaut du diaporama	Sélectionnez la durée d'affichage de chaque page du tableau de bord avant que le diaporama ne passe à la page suivante.
Démarrer le diaporama	Cochez cette case pour démarrer automatiquement le diaporama lorsque le tableau de bord comporte plusieurs pages.

Cliquez ensuite sur *Appliquer*. Le nouveau tableau de bord s'ouvre en mode édition :



Vous pouvez alors ajouter des widgets et des pages au tableau de bord.

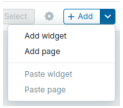
Notez que la largeur minimale d'un tableau de bord est de 1200 pixels ; dans les fenêtres plus étroites, une barre de défilement horizontale apparaît. Un tableau de bord plus large remplit toujours la fenêtre du navigateur ; ses widgets ne peuvent pas dépasser les bords de la fenêtre.

Cliquez sur le bouton *Enregistrer les modifications* pour enregistrer le tableau de bord, ou sur *Annuler* pour revenir à l'écran précédent.

Ajouter des widgets

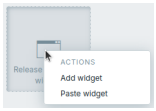
En mode d'édition du tableau de bord, vous pouvez ajouter des widgets de deux façons :

- Cliquez sur le bouton *Ajouter*. Vous pouvez également cliquer sur la flèche déroulante du bouton *Ajouter* et sélectionner *Ajouter un widget*.



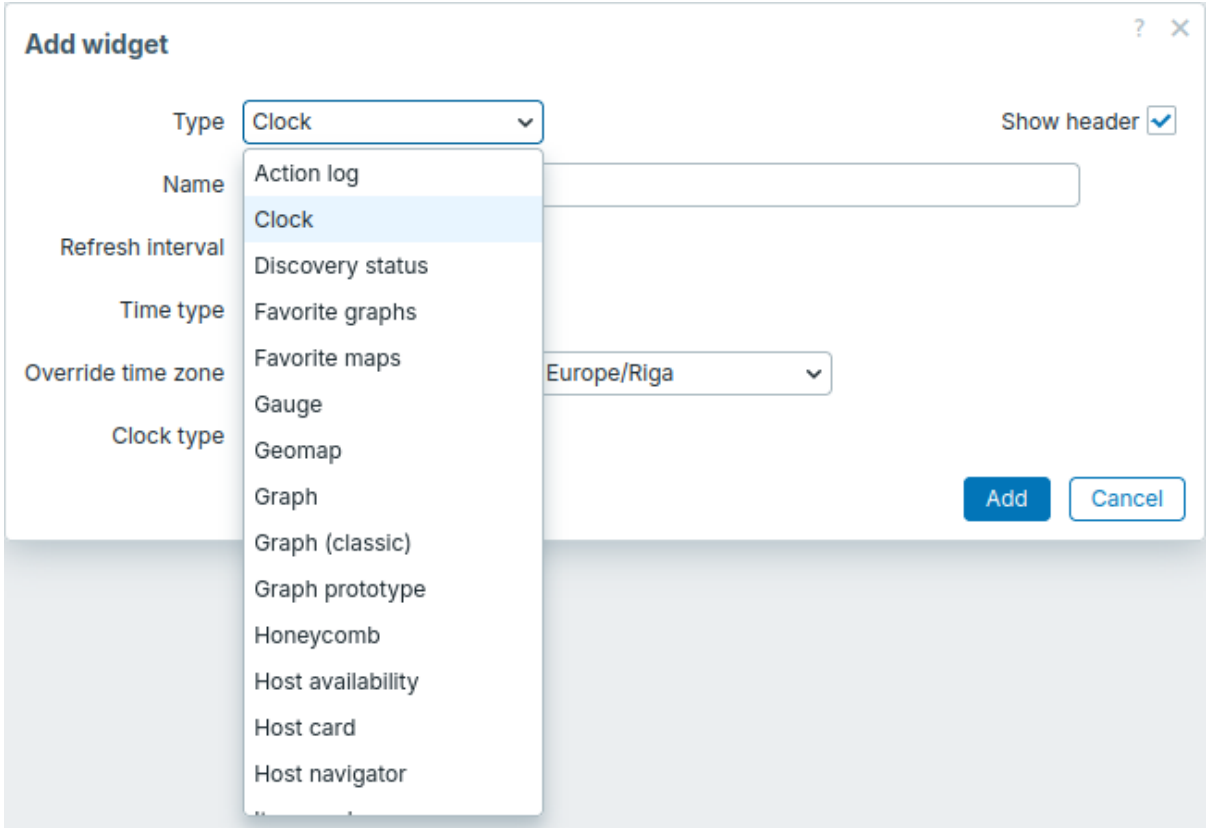
L'espace réservé du widget est ajouté à sa taille par défaut dans le premier espace vide disponible du tableau de bord, avec le formulaire de configuration du widget ouvert.

- Déplacez le pointeur de la souris sur une zone vide du tableau de bord. Cliquez ensuite sur l'espace réservé du widget pour ouvrir le formulaire de configuration du widget. Vous pouvez également cliquer sur l'espace réservé et le faire glisser pour le redimensionner, puis relâcher le bouton pour ouvrir le formulaire de configuration du widget ; si un widget est copié dans le presse-papiers, vous serez d'abord invité à choisir *Ajouter un widget* ou *Coller le widget*.



Suivez ensuite les étapes suivantes :

1. Dans le formulaire de configuration du widget, sélectionnez le type de widget.
2. Configurez les paramètres du widget.
3. Cliquez sur *Ajouter* pour ajouter le widget au tableau de bord.



Le widget est ajouté à sa taille par défaut ou redimensionné pour s'adapter à l'espace disponible. Vous pouvez redimensionner et repositionner les widgets comme décrit dans la section *Modifier les widgets*.

Vous pouvez également copier et coller des widgets pour créer de nouveaux widgets avec la même configuration, soit dans le même tableau de bord, soit entre des tableaux de bord ouverts dans différents onglets du navigateur.

Copier/coller des widgets

Pour copier et coller un widget :

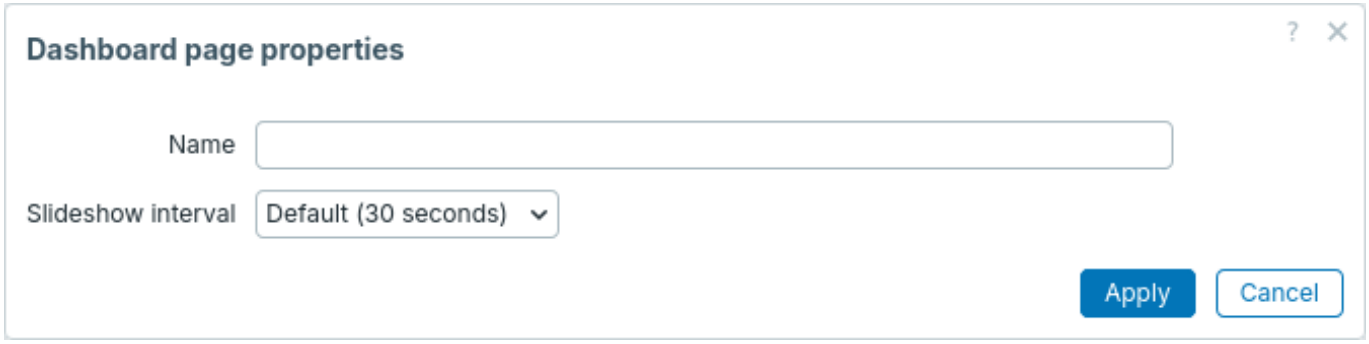
1. Dans le menu *Actions* du widget ⋯, sélectionnez *Copier*.
2. En mode de modification du tableau de bord, cliquez sur la flèche déroulante du bouton *Ajouter* et sélectionnez *Coller le widget*. Vous pouvez également cliquer sur une zone vide du tableau de bord et sélectionner *Coller le widget*.

Vous pouvez aussi coller un widget copié par-dessus un widget existant à l'aide de l'option *Coller* dans son menu *Actions*.

Ajouter des pages

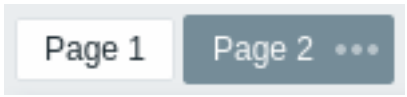
Pour ajouter une nouvelle page à un tableau de bord :

1. En mode d'édition du tableau de bord, cliquez sur la flèche déroulante du bouton *Ajouter* et sélectionnez *Ajouter une page*.
2. Renseignez les propriétés de la page :
   - *Nom* - saisissez le nom de la page. Si vous laissez le nom vide, la page est ajoutée avec le nom *Page N*, où « N » correspond au numéro de page incrémentiel.
   - *Intervalle du diaporama* - sélectionnez la durée d'affichage de la page avant que le diaporama ne passe à la page suivante.



3. Cliquez sur *Appliquer*.

Un nouvel onglet (par exemple, *Page 2*) est ajouté pour la nouvelle page.



Vous pouvez également copier et coller des pages pour créer de nouvelles pages avec les mêmes propriétés, soit dans le même tableau de bord, soit entre des tableaux de bord ouverts dans des onglets différents.

Copier/coller des pages

Pour copier et coller une page :

1. Dans le menu *Actions* de la page ⋯, sélectionnez *Copier*.
2. En mode modification du tableau de bord, cliquez sur la flèche déroulante du bouton *Ajouter* et sélectionnez *Coller la page*.

Modifier un tableau de bord

Pour modifier un tableau de bord, cliquez sur le bouton *Modifier le tableau de bord* dans le coin supérieur droit. Vous pouvez également cliquer sur le bouton *Modifier* ◦ d'un widget. Le tableau de bord s'ouvre en mode modification.

À partir de là, vous pouvez gérer les tableaux de bord comme suit :

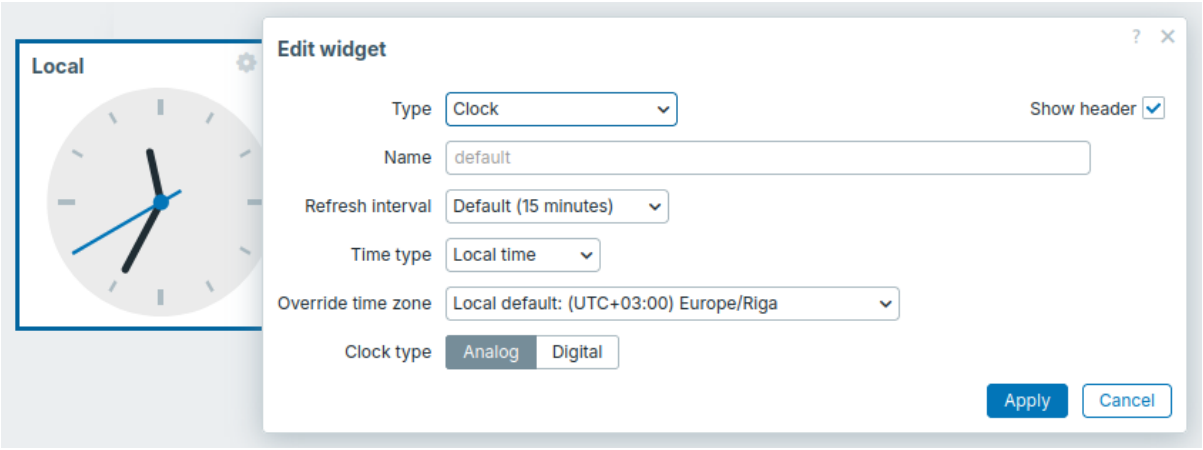
- Modifiez les propriétés du tableau de bord en cliquant sur le bouton représentant une roue dentée ◦.
- Modifiez les widgets du tableau de bord.
- Accédez aux actions suivantes du tableau de bord en cliquant sur la flèche déroulante du bouton *Ajouter* :
  - *Ajouter un widget* - ajoutez un widget au tableau de bord.
  - *Ajouter une page* - ajoutez une page au tableau de bord.
  - *Coller le widget* - collez un widget copié. Cette option est grisée si aucun widget n'a été copié. Un seul widget peut être copié à la fois.
  - *Coller la page* - collez une page copiée. Cette option est grisée si aucune page n'a été copiée. Une seule page peut être copiée à la fois.

Après avoir modifié le tableau de bord, cliquez sur le bouton *Enregistrer les modifications* pour enregistrer le tableau de bord, ou sur *Annuler* pour revenir à l'écran précédent.

Modifier les widgets

Un widget peut être modifié en mode d'édition du tableau de bord en cliquant sur son bouton *Modifier* . Vous pouvez également, lorsque vous consultez un tableau de bord, survoler un widget et cliquer sur le bouton *Modifier*.

Pendant la modification d'un widget, celui-ci est mis en évidence et les changements sont prévisualisés en temps réel.



Vous pouvez repositionner le widget et son formulaire de configuration en faisant glisser leurs barres de titre, et redimensionner le widget en faisant glisser ses poignées de redimensionnement.



Les actions suivantes sont disponibles dans le menu *Actions* du widget ... lors de la modification d'un widget :

- *Copier* - copier le widget.
- *Coller* - coller un widget copié par-dessus ce widget. Cette option est grisée si aucun widget n'a été copié.
- *Supprimer* - supprimer le widget.

Si les paramètres obligatoires du widget ne sont pas configurés, le widget passe à l'état *Le widget n'est pas entièrement configuré*.

Après la modification, cliquez sur *Appliquer* dans le formulaire de configuration du widget, puis sur *Enregistrer les modifications* dans le coin supérieur droit du tableau de bord pour appliquer vos modifications.

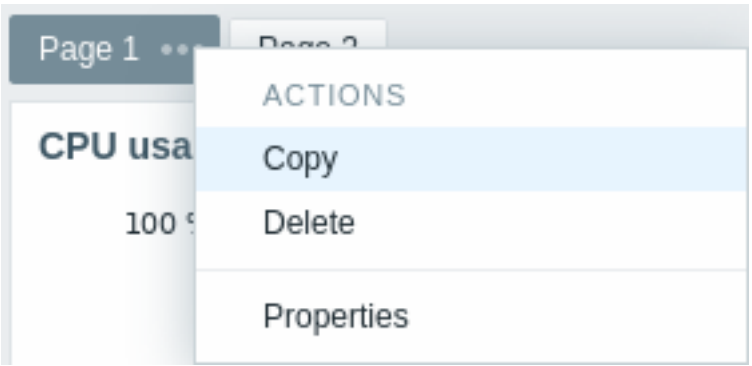
Modifier les pages

Les pages peuvent être modifiées en mode d'édition du tableau de bord.

Les pages peuvent être réorganisées par glisser-déposer des onglets de page. La réorganisation ne renomme pas les pages, sauf celles dont le nom est attribué automatiquement (*Page N*), qui sont renumérotées.

Les actions suivantes sont disponibles dans le menu *Actions* de la page lors de la modification d'une page :

- *Copier* - copier la page.
- *Supprimer* - supprimer la page.
- *Propriétés* - personnaliser les propriétés de la page.



Autorisations des tableaux de bord

Les autorisations relatives aux tableaux de bord pour les utilisateurs standard et les utilisateurs de type *Admin* sont limitées comme décrit ci-dessous, à condition que l'accès aux tableaux de bord et l'option *Créer et modifier les tableaux de bord* soient activés pour leur rôle utilisateur attribué :

- Ils peuvent consulter et cloner un tableau de bord public, même si aucun utilisateur ni groupe d'utilisateurs n'est ajouté aux préférences de partage du tableau de bord.
- Ils peuvent consulter et cloner un tableau de bord privé s'ils disposent au moins des droits de *Lecture* sur celui-ci, comme indiqué dans les préférences de partage.
- Ils peuvent modifier et supprimer un tableau de bord uniquement s'ils disposent des droits de *Lecture-écriture* sur celui-ci, comme indiqué dans les préférences de partage.
- Ils ne peuvent pas modifier le propriétaire du tableau de bord.

Widgets du tableau de bord

Vue d'ensemble

Zabbix propose une variété de widgets de tableau de bord qui visualisent et résument les données de supervision, vous offrant une vue claire des performances et de la disponibilité du système.

Vous trouverez ci-dessous une vue d'ensemble de tous les widgets, regroupés selon leur objectif principal.

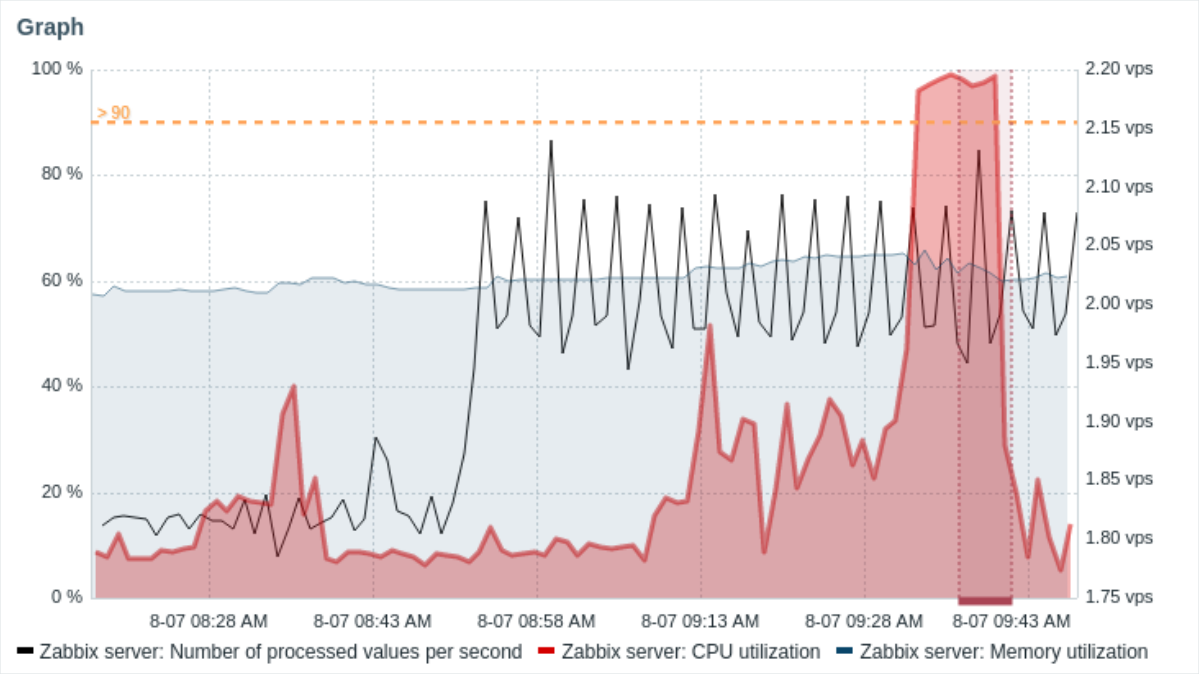
Pour plus de détails sur la configuration des widgets, consultez Paramètres des widgets de tableau de bord.

Visualisation des données

Ces widgets présentent les données sous différents formats visuels, vous aidant à surveiller les données et les problèmes de différentes manières.

Graphique

Le widget Graphique affiche les données numériques des éléments sous forme de graphique vectoriel. Il peut vous aider à suivre les métriques, à repérer les problèmes et à comparer les valeurs au fil du temps et entre les hôtes.

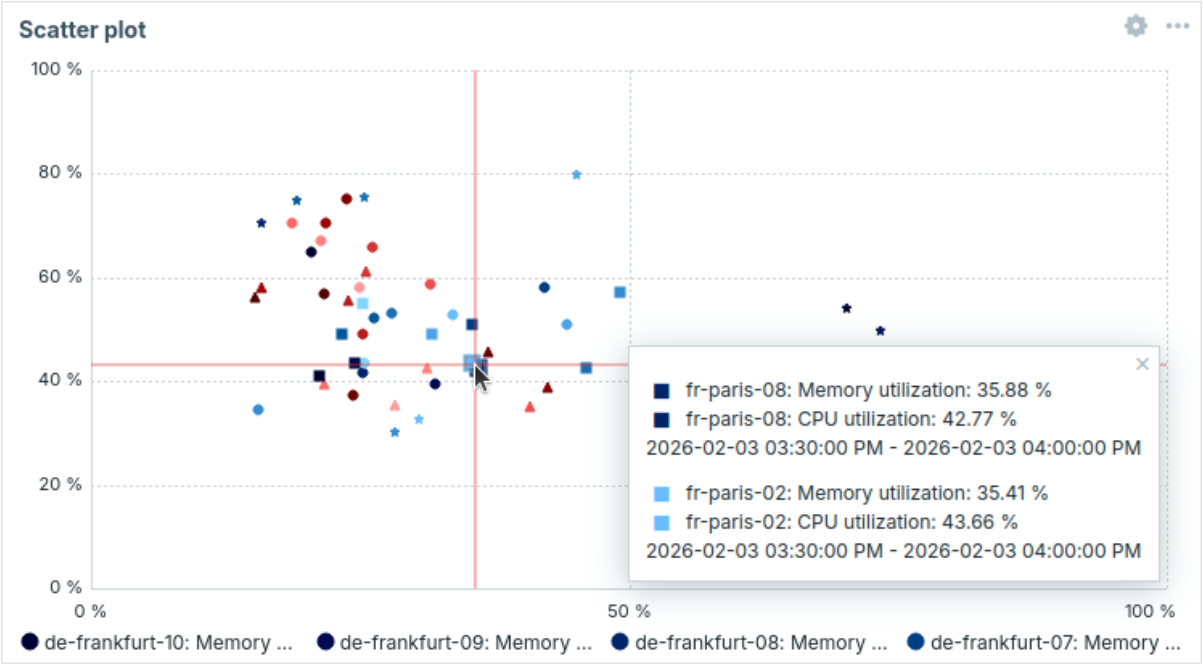


Voir aussi :

- Graphique (classique) - l'ancien widget Graphique qui affiche les données numériques des éléments sous forme de graphique basé sur une image.
- Prototype de graphique - affiche une grille de graphiques personnalisés créés automatiquement à partir de prototypes de graphiques ou de prototypes d'éléments de règle de découverte de bas niveau.
- Graphiques favoris - affiche une liste alphabétique de raccourcis vers les graphiques marqués comme favoris par l'utilisateur actuel.

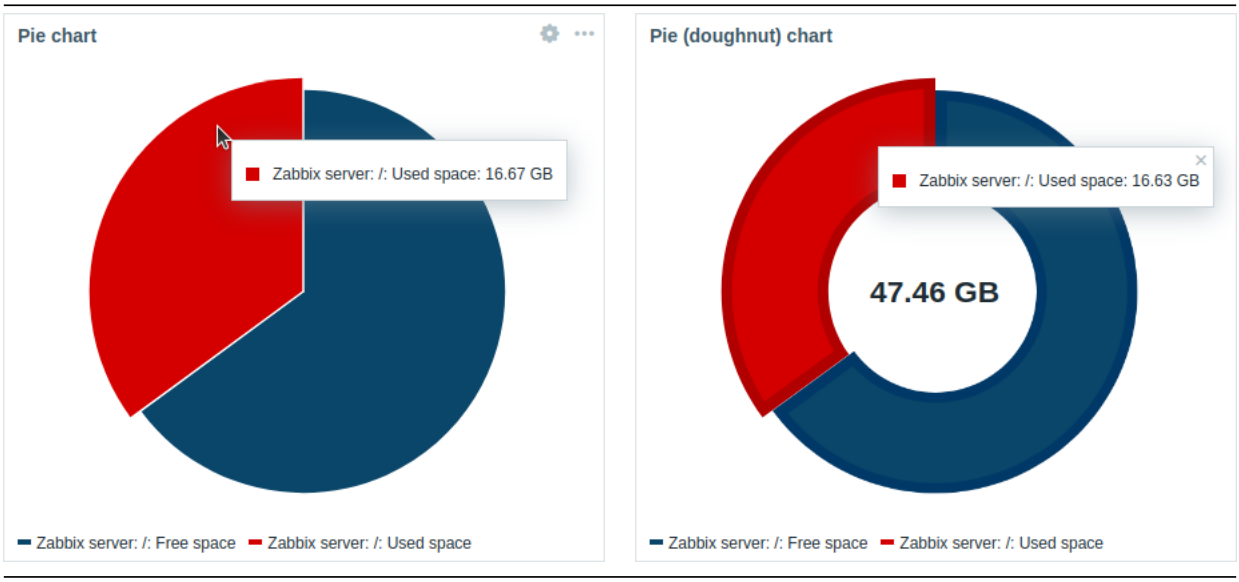
Nuage de points

Le widget Scatter plot affiche la relation entre deux métriques en traçant des points de données individuels le long des axes X et Y. Cela aide à révéler des motifs, des regroupements, des corrélations et des valeurs aberrantes dans l'ensemble de données.



Camembert

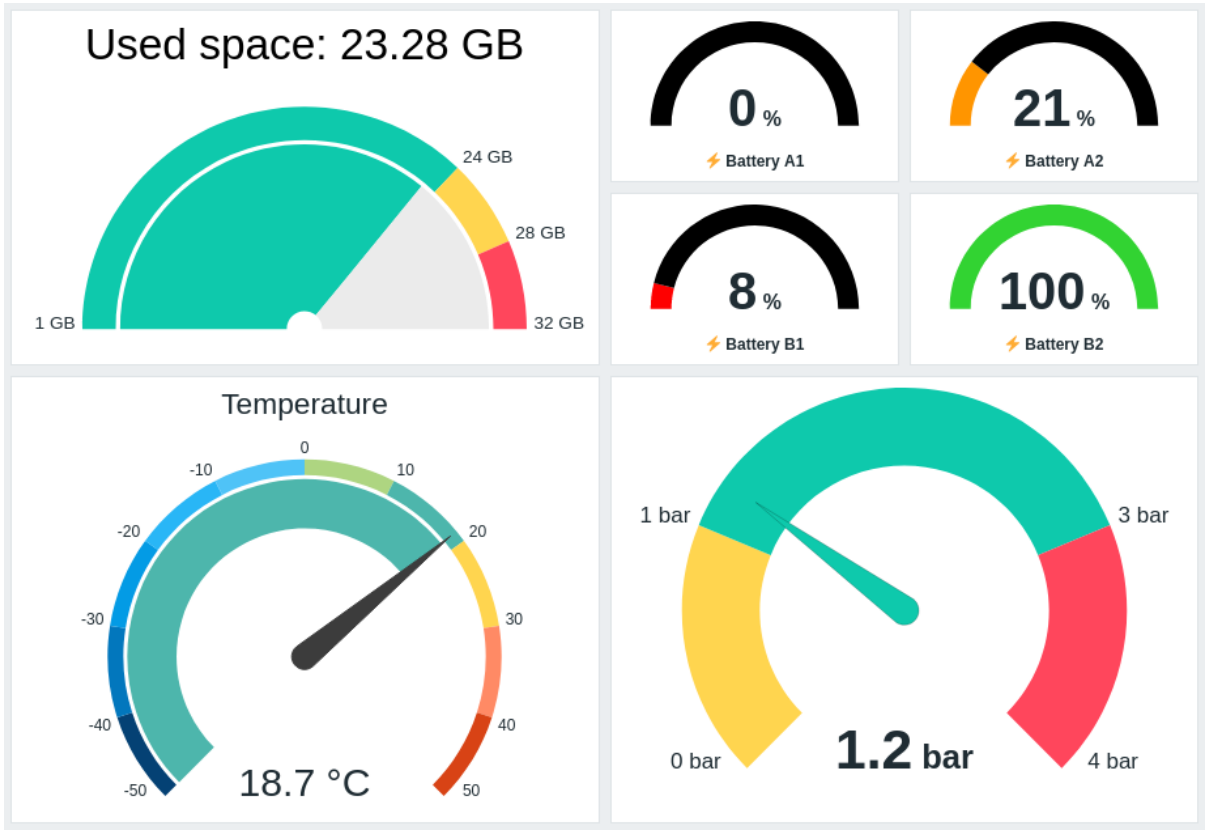
Le widget Camembert affiche des données numériques d'éléments sous la forme d'un graphique vectoriel en secteurs ou en anneau. Ce widget peut vous aider à visualiser comment les éléments ou les hôtes contribuent à l'ensemble du jeu de données.



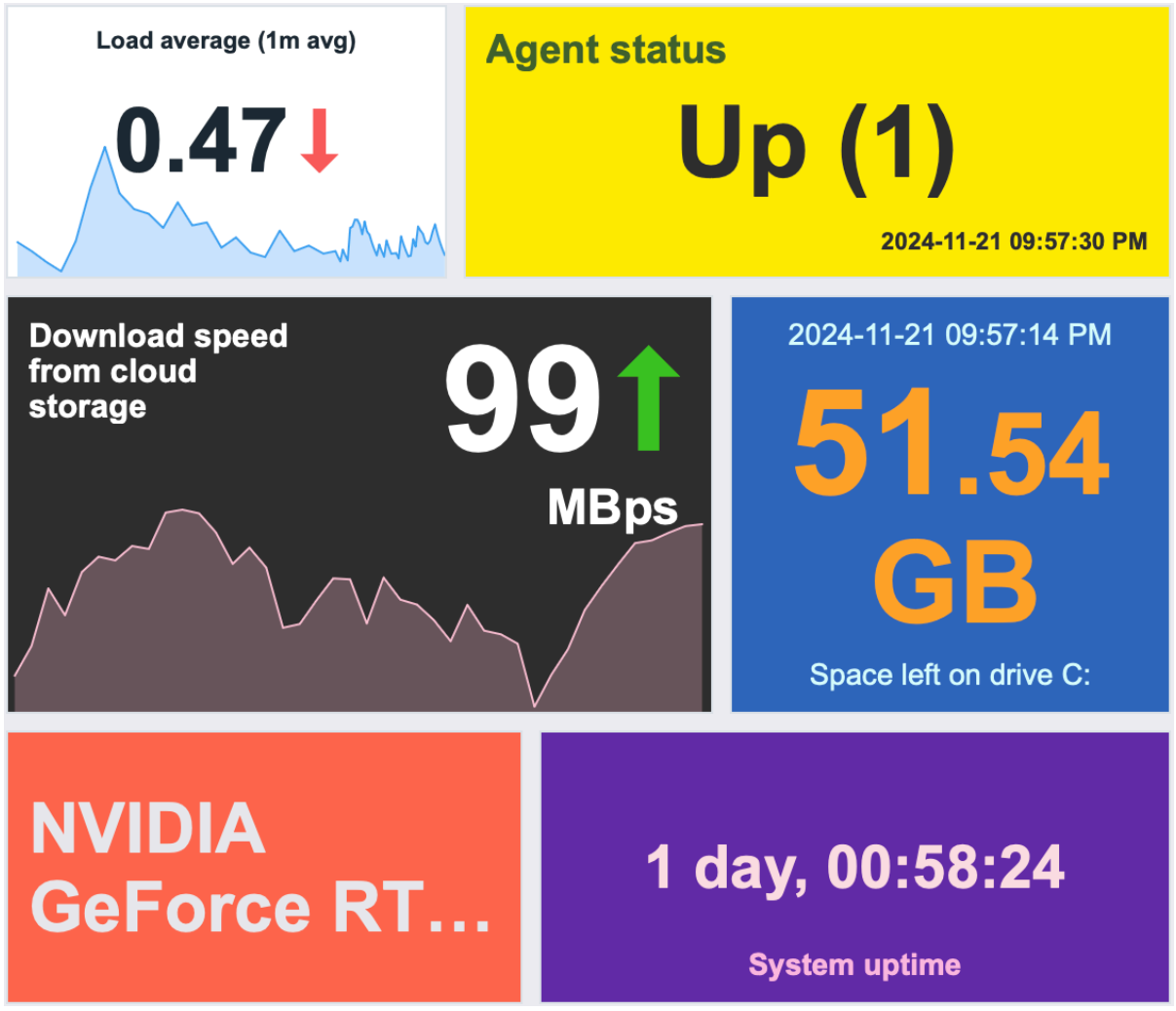
Jauge/Valeur d'élément

Les widgets Jauge/Valeur d'élément affichent la valeur d'un seul élément, ce qui est utile pour surveiller des métriques importantes, visualiser des seuils et détecter des changements soudains dans les données.

Widget Jauge :

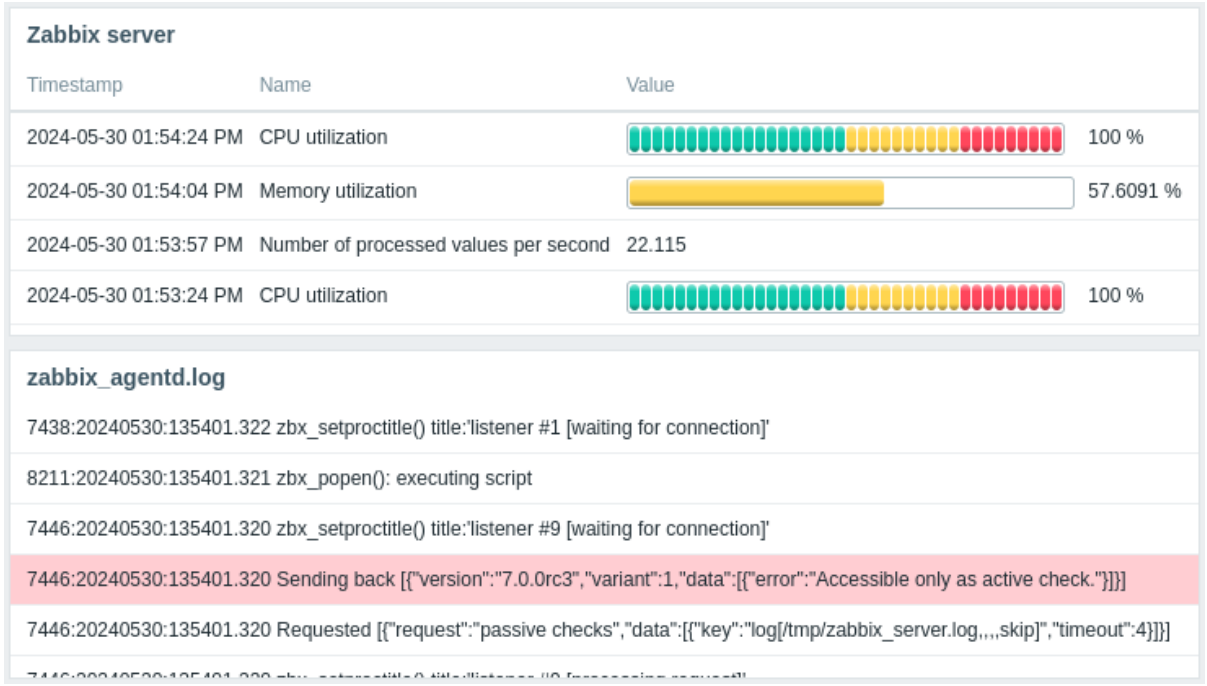


Widget Valeur d’élément :



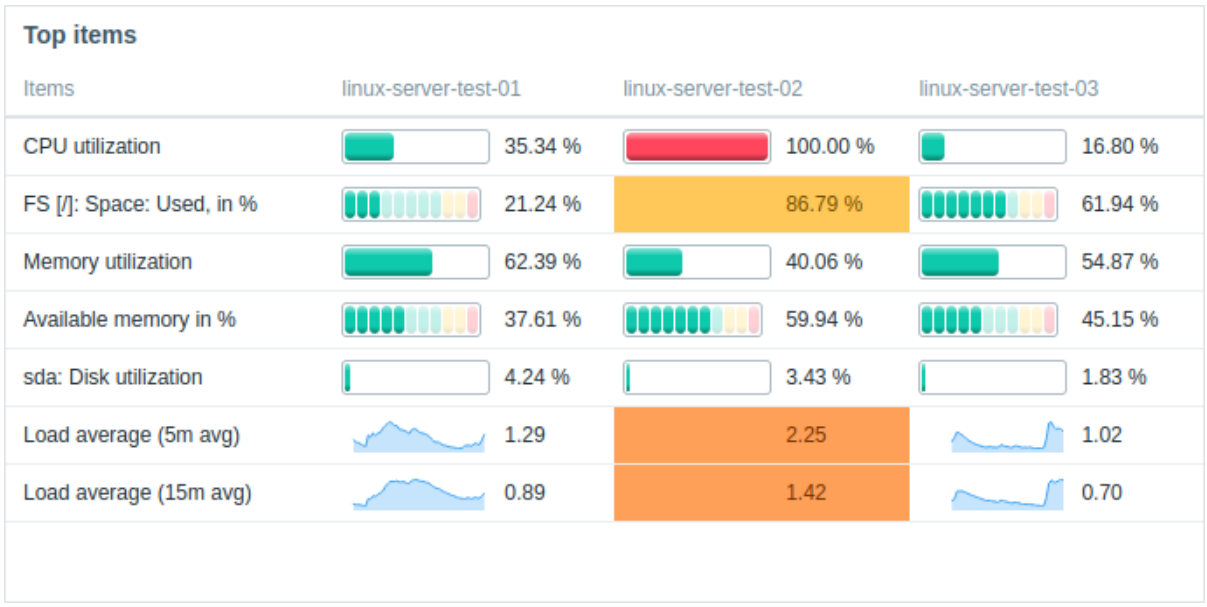
Historique des éléments

Le widget Item history affiche les données les plus récentes pour tous les types d'éléments (numériques, texte, etc.) sous forme de tableau. Il peut afficher des barres de progression et des images (utile pour les éléments de navigateur) et peut mettre en évidence des valeurs (utile pour la surveillance des fichiers journaux).



Top des éléments

Le widget Top items affiche les valeurs les plus élevées/les plus basses pour les éléments sélectionnés, offrant un aperçu rapide de leurs performances.

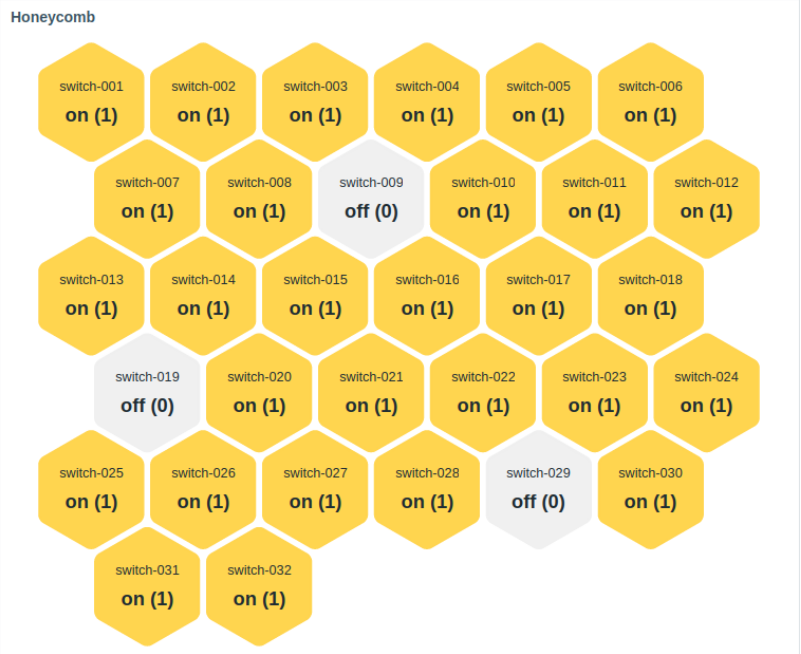


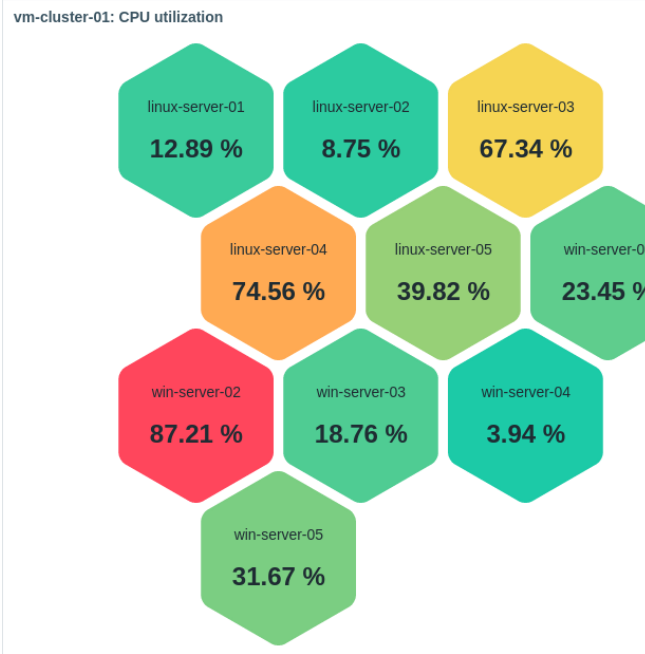
Voir aussi :

- Top hosts - affiche les valeurs les plus élevées/les plus basses pour les éléments sélectionnés provenant de plusieurs hôtes.
- Top triggers - affiche les déclencheurs avec le plus grand nombre de problèmes.

Nid d'abeilles

Le widget Honeycomb affiche les hôtes ou les éléments sous forme de cellules hexagonales dans une grille. Cette disposition permet de détecter facilement les hôtes/éléments en problème, de repérer des groupes de problèmes, de comparer des groupes d'un coup d'œil, et plus encore.



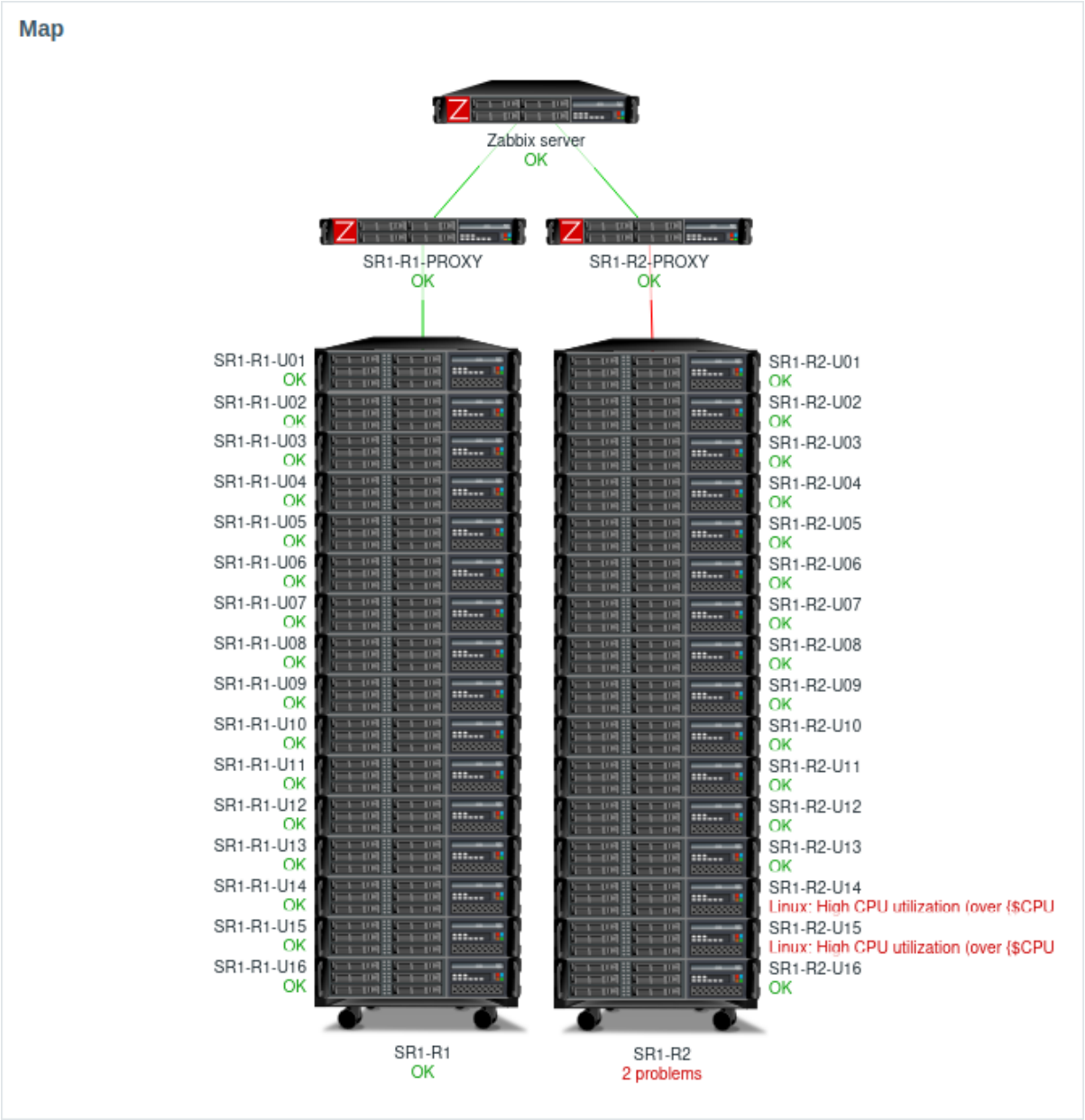


Cartes

Les widgets de carte visualisent la topologie du réseau ou l'emplacement des hôtes, ce qui aide à évaluer l'état global ainsi que l'état par emplacement.

Carte

Le widget Map peut afficher une carte réseau ou une visualisation similaire, vous offrant une vue d'ensemble dynamique de votre réseau.

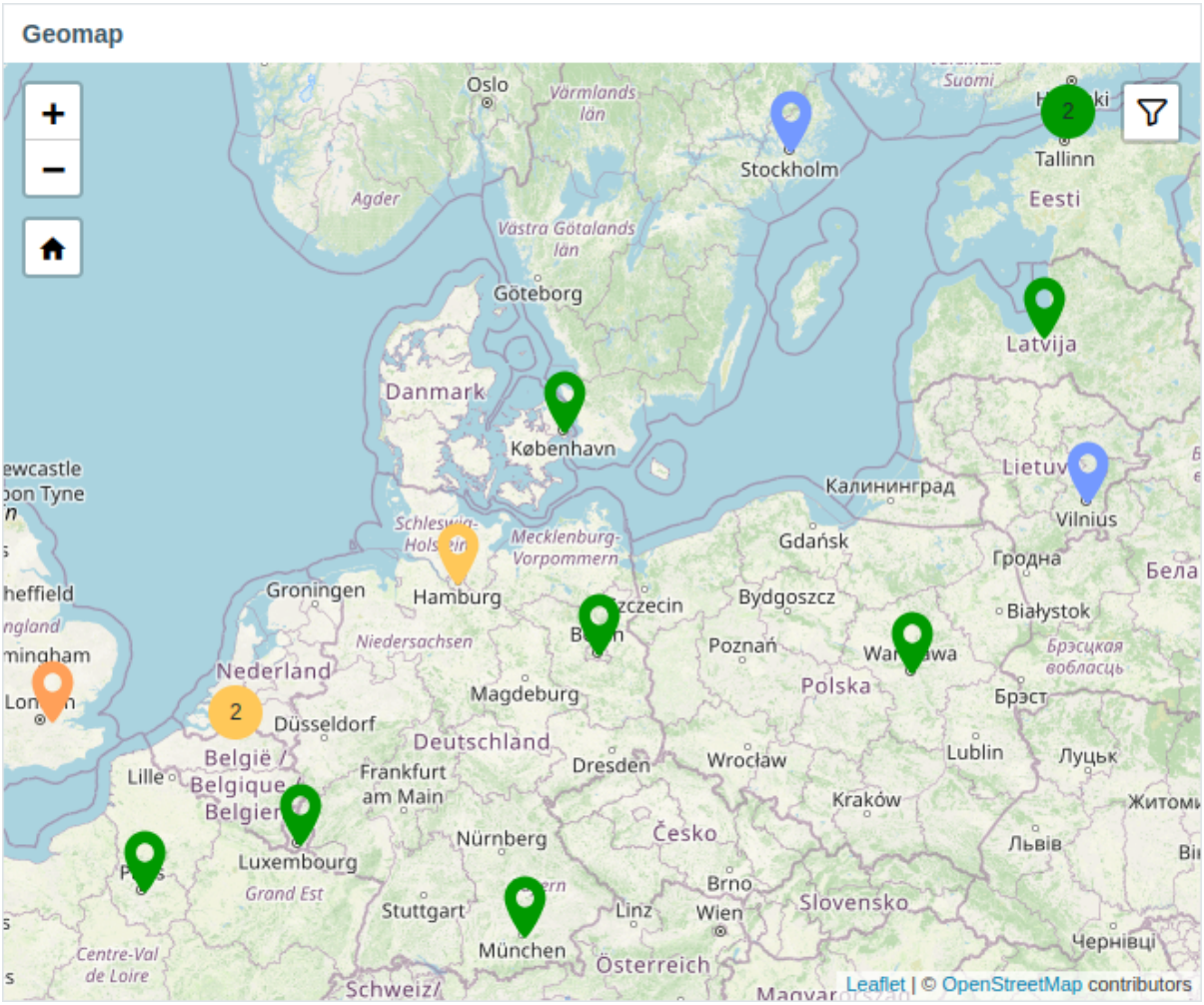


Voir aussi :

- Arborescence de navigation des cartes - affiche une hiérarchie des cartes existantes ainsi que le nombre de problèmes pour chaque carte et groupe de cartes.
- Cartes favorites - affiche une liste de raccourcis vers les cartes marquées comme favorites par l'utilisateur actuel, triées par ordre alphabétique.

Carte géographique

Le widget Geomap affiche les hôtes sous forme de marqueurs sur une carte interactive basée sur Leaflet, la couleur des marqueurs indiquant le problème le plus grave de chaque hôte.

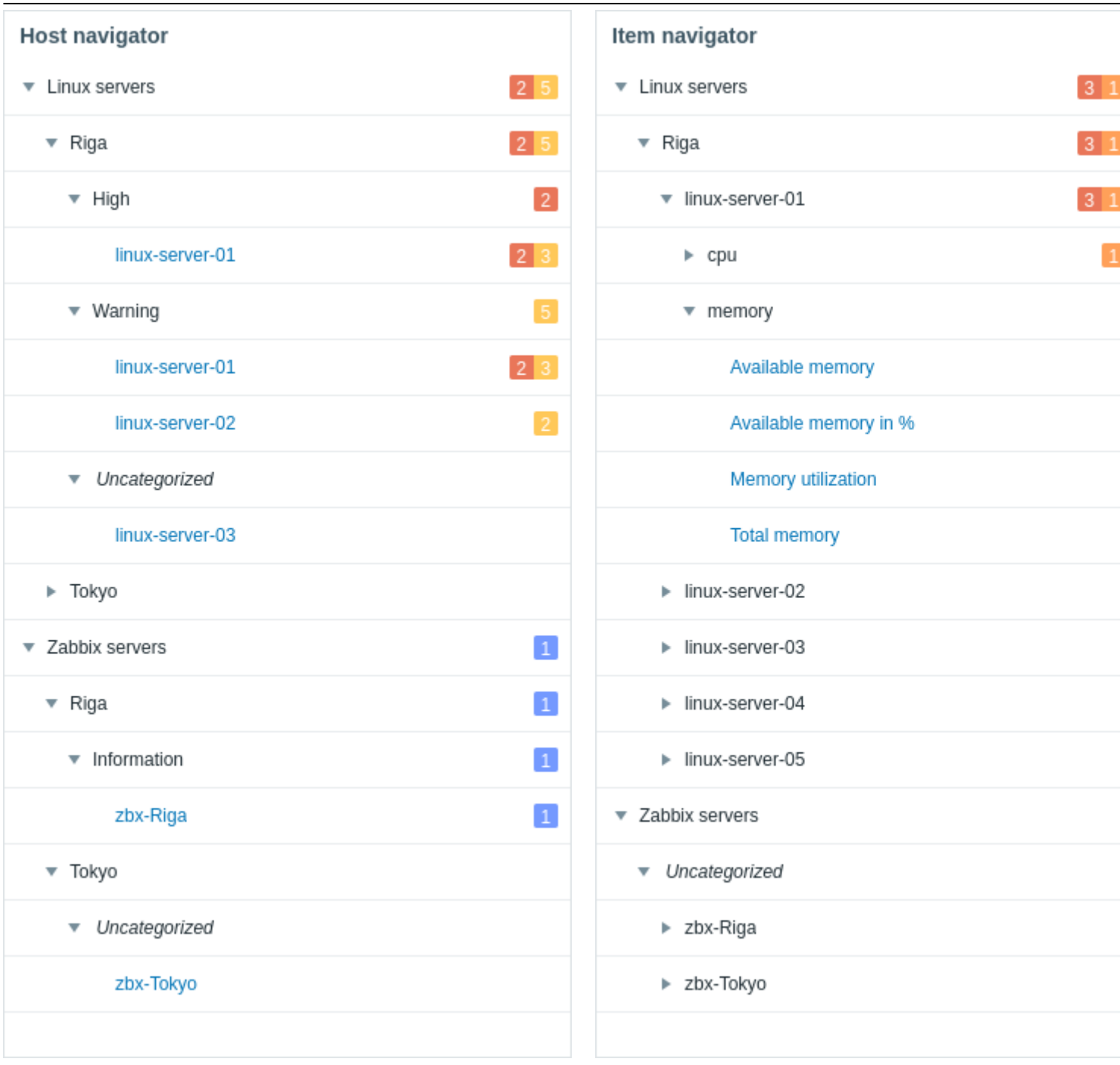


Détails de l’hôte/de l’élément

Ces widgets fournissent des listes et des vues détaillées des hôtes ou des éléments.

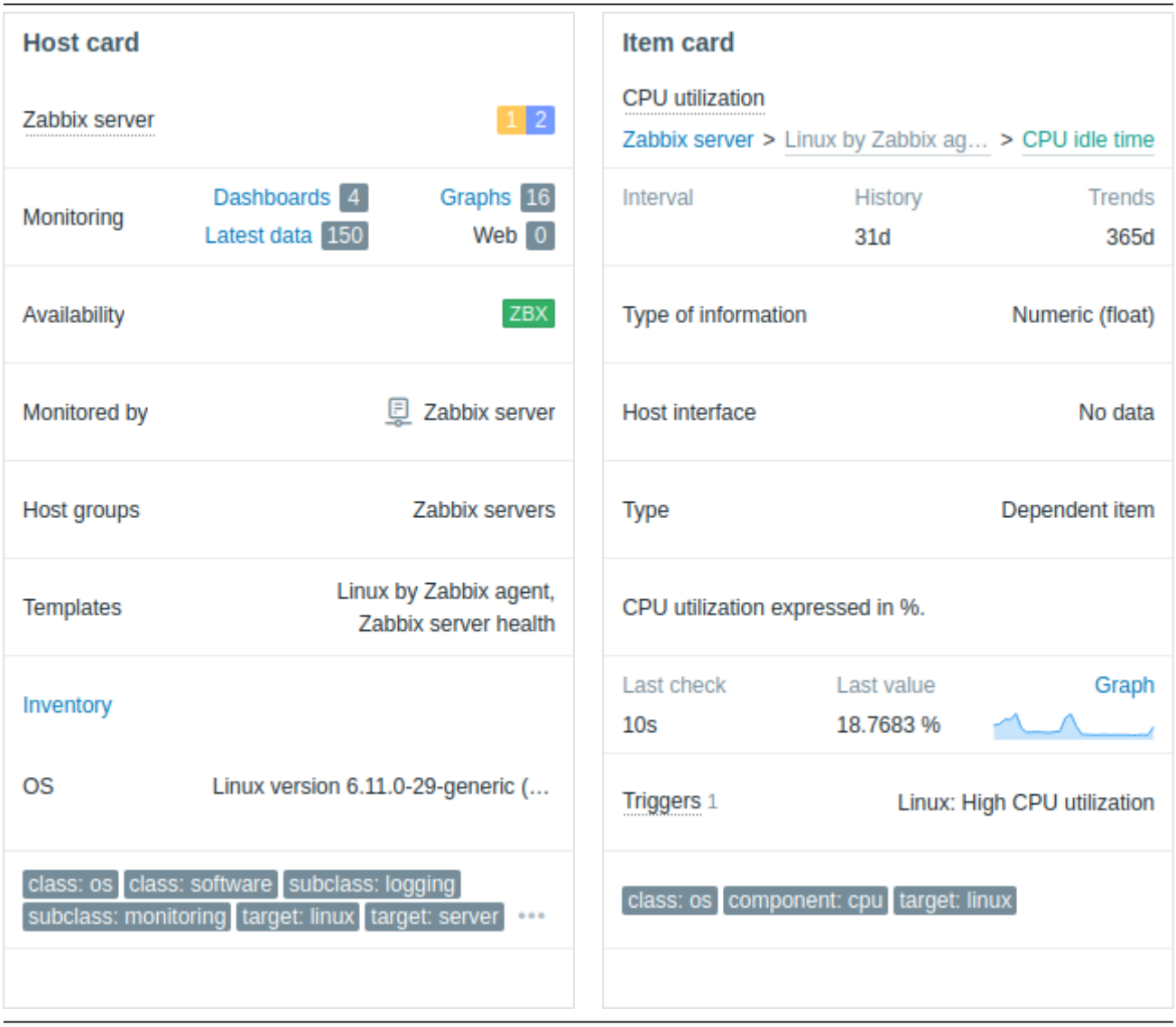
Navigateur d’hôte/d’élément

Les widgets Navigateur d’hôte/Navigateur d’élément affichent une liste d’hôtes/d’éléments selon diverses options de filtrage et de regroupement. Ces widgets sont particulièrement utiles pour contrôler ce que les autres widgets affichent en fonction de l’hôte/de l’élément sélectionné.



Carte d'hôte/élément

Les widgets Carte d'hôte/Carte d'élément affichent les détails d'un seul hôte/élément, ce qui vous permet d'évaluer rapidement l'état et la configuration d'un hôte ou d'un élément (disponibilité de l'hôte, problèmes, inventaire, dernières données de l'élément, erreurs, description, etc.).

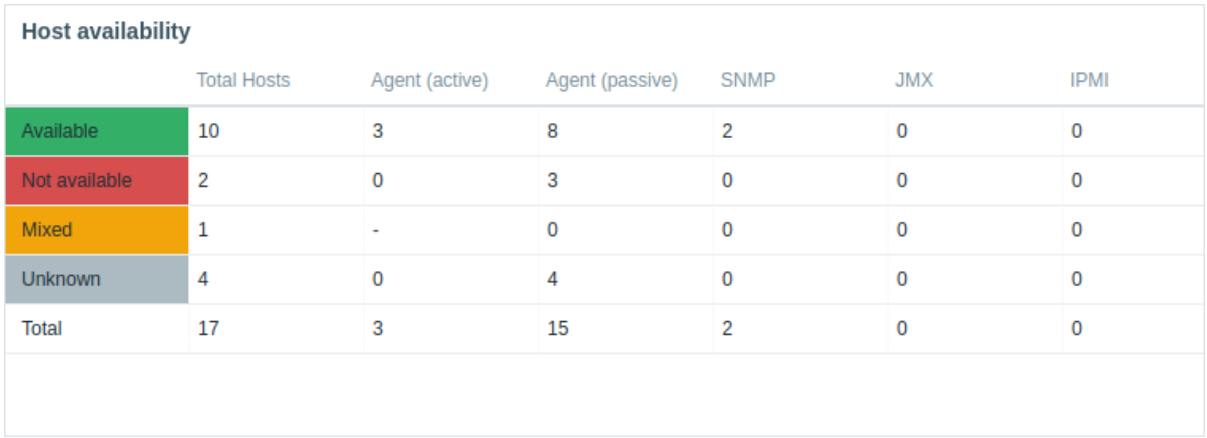


Statut et rapports

Ces widgets affichent des données d'état (par ex. disponibilité, déclencheurs, problèmes) ainsi que des rapports.

Disponibilité de l'hôte

Le widget Disponibilité de l'hôte affiche la disponibilité des hôtes dans les groupes d'hôtes sélectionnés, vous permettant de surveiller le nombre d'hôtes qui sont opérationnels ou indisponibles.

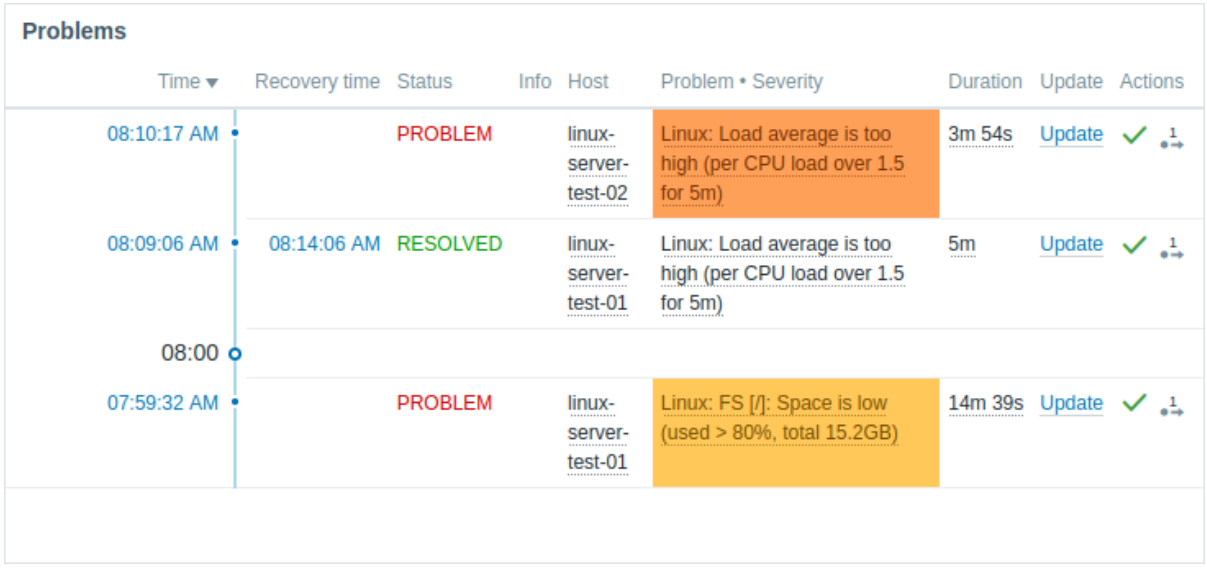


Host availability

	Total Hosts	Agent (active)	Agent (passive)	SNMP	JMX	IPMI
Available	10	3	8	2	0	0
Not available	2	0	3	0	0	0
Mixed	1	-	0	0	0	0
Unknown	4	0	4	0	0	0
Total	17	3	15	2	0	0

Problèmes

Le widget Problems affiche les problèmes actuels filtrés selon différents paramètres (groupes d'hôtes, hôtes, noms de problèmes, etc.), vous offrant une vue claire de ce qui nécessite votre attention.



Voir aussi :

- Problems by severity - affiche le nombre de problèmes causes regroupés par gravité.
- Problem hosts - affiche le nombre de problèmes causes par groupe d'hôtes et indique la gravité de problème la plus élevée dans chaque groupe.

Aperçu des déclencheurs

Le widget Aperçu des déclencheurs affiche les états actuels des déclencheurs des hôtes sous la forme d'un tableau de blocs colorés. Il fournit un aperçu visuel rapide de l'état de santé et de l'activité des déclencheurs sur différents hôtes.



Journal des actions

Le widget Action log affiche les détails des opérations (notifications, commandes distantes) exécutées dans le cadre d'une action. Le journal détaillé facilite l'audit, le dépannage et le suivi de l'exécution de chaque action.

Action log

Time ▾	Action	Media type	Recipient	Message	Status	Info
2025-08-15 07:58:30 AM	Report problems to Zabbix administrators	Email	Admin (Zabbix Administrator) *admin@example.com*	**Resolved in 1m 59s: Linux: Load average is too high (per CPU load over 1.5 for 5m)** Problem has been resolved at 10:58:30 on 2025.08.15 Problem name: Linux: Load average is too high (per CPU load over 1.5 for 5m) Problem duration: 1m 59s Host: Linux server Severity: Average Original problem ID: 45	Sent	
2025-08-15 07:56:30 AM	Report problems to Zabbix administrators	Email	Admin (Zabbix Administrator) *admin@example.com*	**Problem: Linux: Load average is too high (per CPU load over 1.5 for 5m)** Problem started at 10:56:30 on 2025.08.15 Problem name: Linux: Load average is too high (per CPU load over 1.5 for 5m) Host: Linux server Severity: Average Operational data: Load averages(1m 5m 15m): (6.256836 4.808105 3.398438), # of CPUs: 2 Original problem ID: 45	Sent	
2025-08-15 07:42:30 AM	Report problems to Zabbix administrators		Admin (Zabbix Administrator)		Failed	i

Rapport SLA

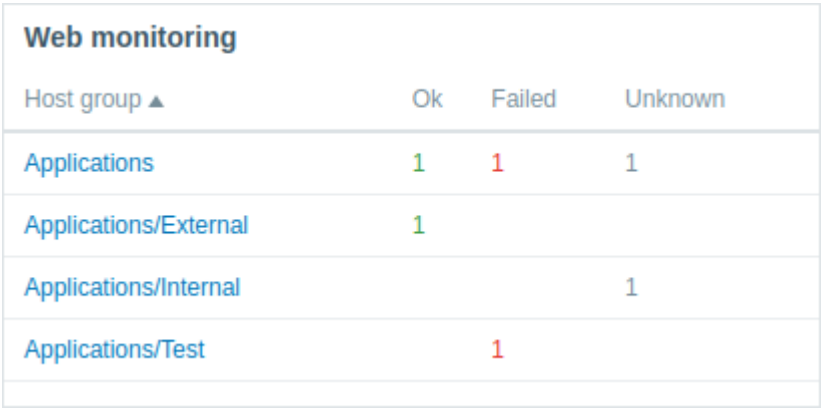
Le widget SLA report affiche des rapports SLA, vous aidant à surveiller les performances des services par rapport aux objectifs définis.

SLA report

Day	SLO	SLI	Uptime	Downtime	Error budget	Excluded downtimes
2025-09-04	99.9%	99.8437	5h 40m 50s	28s	-8s	
2025-09-03	99.9%	99.9479	23h 59m 15s	45s	41s	
2025-09-02	99.9%	99.9190	23h 58m 50s	1m 10s	16s	
2025-09-01	99.9%	99.9251	22h 58m 58s	1h 1m 2s	21s	2025-09-01 01:00 AM Maintenance: 1h
2025-08-31	99.9%	99.6863	23h 55m 29s	4m 31s	-3m 5s	
2025-08-30	99.9%	99.9190	23h 58m 50s	1m 10s	16s	
2025-08-29	99.9%	99.7095	23h 55m 49s	4m 11s	-2m 45s	
2025-08-28	99.9%	99.9421	23h 59m 10s	50s	36s	
2025-08-27	99.9%	99.7188	23h 55m 57s	3m 15s	-2m 37s	
2025-08-26	99.9%	99.7755	23h 56m 46s	3m 14s	-1m 48s	
2025-08-25	99.9%	99.9094	22h 58m 45s	1h 1m 15s	8s	2025-08-25 01:00 AM Maintenance: 1h

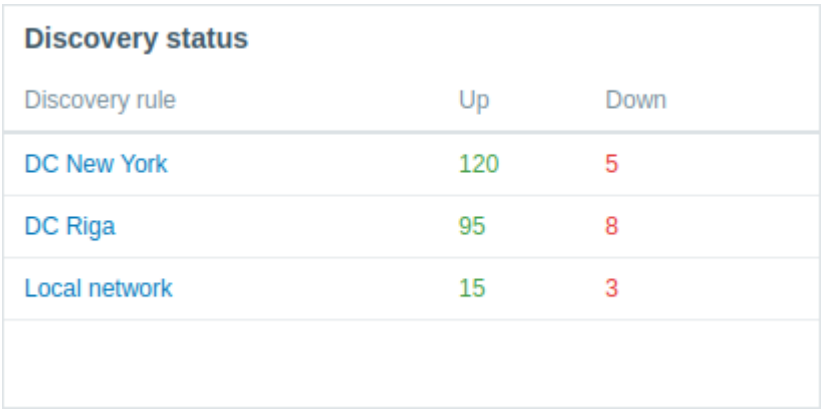
Surveillance web

Le widget Web monitoring affiche l'état des scénarios web actifs.



Statut de découverte

Le widget Discovery status affiche l'état des appareils découverts par les règles de découverte réseau activées. Il vous aide à surveiller si la découverte réseau fonctionne correctement ainsi que le nombre d'appareils opérationnels ou indisponibles.



Utilitaires de tableau de bord et informations système

Ces widgets fournissent des informations complémentaires et des outils pour améliorer vos tableaux de bord.

Informations système

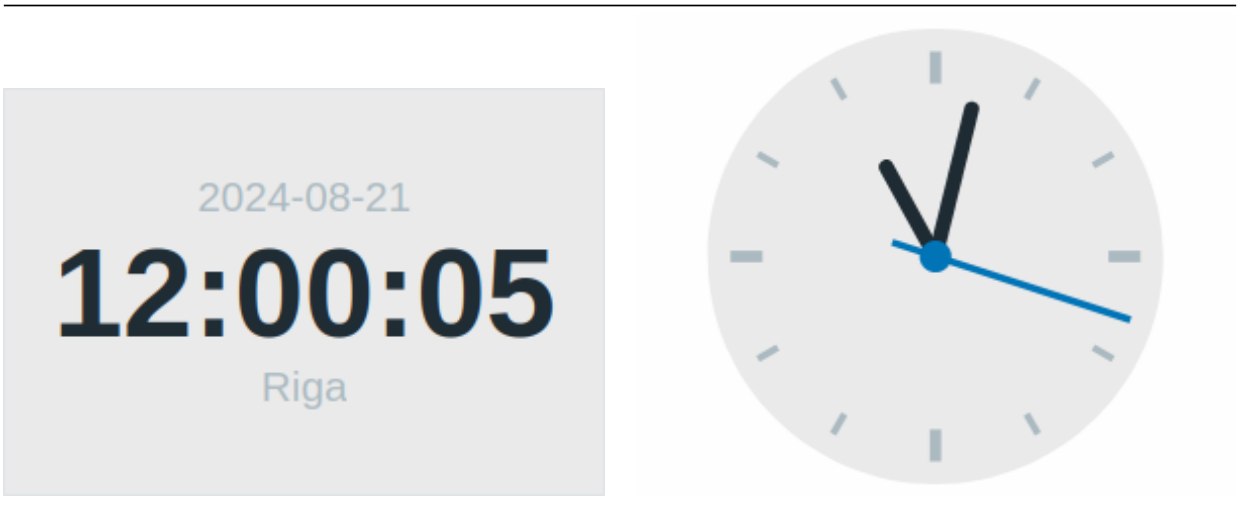
Le widget Informations système affiche un résumé des principales données du serveur Zabbix et du système, ou des détails sur les nœuds de haute disponibilité.

System information

Parameter	Value	Details
Zabbix server is running	Yes	localhost:10051
Zabbix server version	8.0.0	Up to date
Zabbix frontend version	8.0.0	Up to date
Software update last checked	2025-08-30	
Latest release	8.0.0	Release notes
Number of hosts (enabled/disabled)	2	2 / 0
Number of templates	312	
Number of items (enabled/disabled/not supported)	176	165 / 0 / 11
Number of triggers (enabled/disabled [problem/ok])	89	89 / 0 [4 / 85]
Number of users (online)	2	1
Required server performance, new values per second	2.68	
Global scripts on Zabbix server	Disabled	
High availability cluster	Disabled	

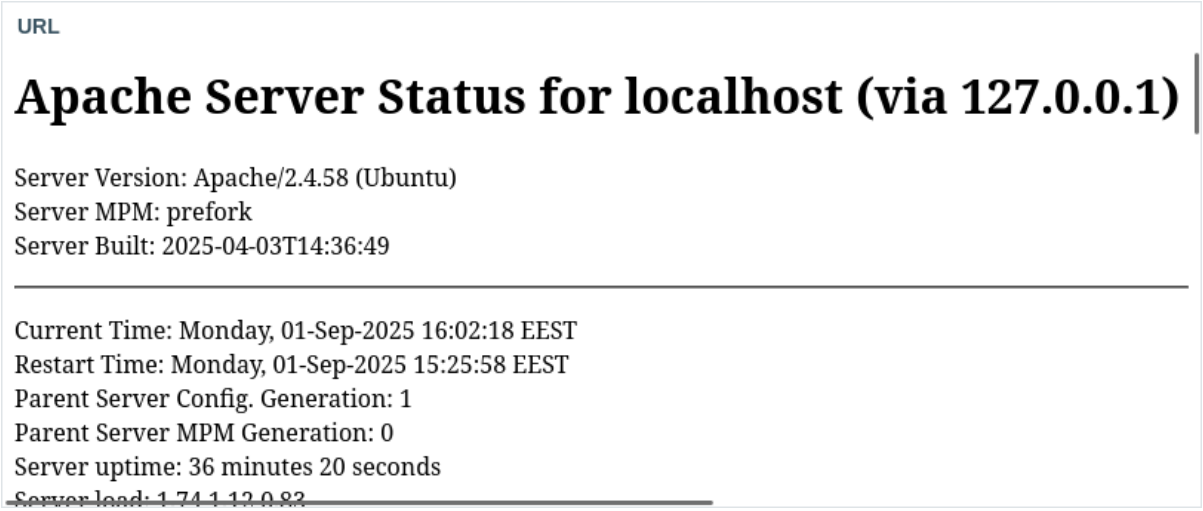
Horloge

Le widget Clock affiche l'heure locale, celle du serveur ou celle de l'hôte au format analogique ou numérique.



URL

Le widget URL affiche le contenu récupéré depuis une URL.



Paramètres du widget du tableau de bord

Vue d'ensemble

Cette page décrit les paramètres communs à tous les widgets du tableau de bord, ainsi que les paramètres dynamiques qui permettent à plusieurs widgets d'échanger des données de configuration entre eux ou avec le tableau de bord.

Pour voir les paramètres spécifiques à chaque widget, veuillez utiliser la barre latérale pour accéder à la page du widget correspondant.

Paramètres communs

Les paramètres suivants sont communs à chaque widget :

Nom	Saisissez un nom pour le widget.
Intervalle d'actualisation	Configurez l'intervalle d'actualisation par défaut. Les intervalles d'actualisation par défaut des widgets vont de *Aucune actualisation* à *15 minutes*, selon le type de widget. Par exemple : - *Aucune actualisation* pour le widget *URL* ; - *1 minute* pour le widget *Journal des actions* ; - *15 minutes* pour le widget *Horloge*. Les intervalles d'actualisation peuvent être définis sur une valeur par défaut pour tous les utilisateurs. Passez le tableau de bord en mode édition, cliquez sur le bouton modifier un widget, puis sélectionnez l'intervalle d'actualisation souhaité dans la liste déroulante. Chaque utilisateur peut également définir son propre intervalle d'actualisation pour les widgets. En mode affichage du tableau de bord, cliquez sur le bouton représentant trois points d'un widget, puis sélectionnez l'intervalle d'actualisation souhaité dans la liste déroulante. Notez que l'intervalle d'actualisation personnalisé d'un utilisateur est prioritaire sur le paramètre du widget et est conservé même lorsque le paramètre du widget est modifié.
Afficher l'en-tête	Cochez la case pour afficher en permanence l'en-tête du widget. Lorsque cette case n'est pas cochée, l'en-tête est masqué afin d'économiser de l'espace et devient visible uniquement au survol du widget avec la souris (en modes affichage et édition). L'en-tête est également semi-visible lorsqu'un widget est déplacé vers un nouvel emplacement.

Paramètres dynamiques

Plusieurs paramètres de widget (par exemple, *Hôtes*, *Remplacer l'hôte*, *Période de temps*) prennent en charge des sources de données dynamiques. Au lieu de configurer des valeurs statiques, vous pouvez lier ces paramètres à d'autres widgets ou au tableau de bord lui-même. Les widgets sont alors automatiquement mis à jour en fonction des sélections ou des diffusions provenant d'autres parties du tableau de bord.

Compatibilité des widgets

Certains widgets peuvent diffuser des données de configuration à d'autres widgets, certains peuvent recevoir des données, et certains peuvent faire les deux. Par exemple :

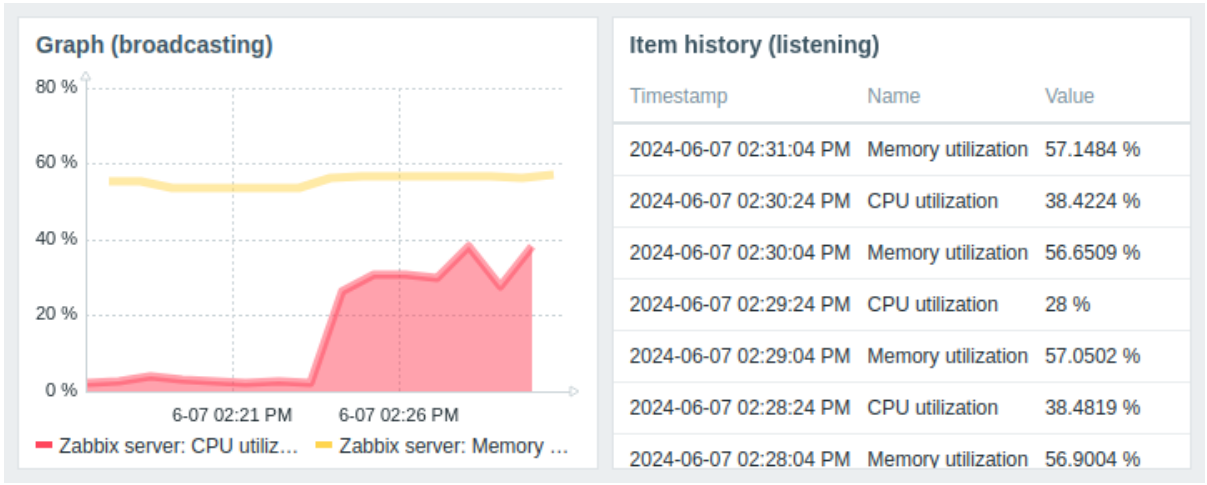
- Le widget *Journal des actions* peut uniquement récupérer les données de période de temps des widgets *Graphique*, *Graphique (classique)* et *Prototype de graphique*.
- Le widget *Geomap* peut diffuser des données d'hôte aux widgets qui les reçoivent (*Honeycomb*, *Top items*, etc.) et peut également recevoir des données de groupe d'hôtes et d'hôte depuis les widgets qui les diffusent (*Honeycomb*, *Problem hosts*, etc.).
- Le widget *Horloge* ne peut ni diffuser ni recevoir de données.

Le tableau suivant présente les capacités de diffusion et de réception de chaque widget.

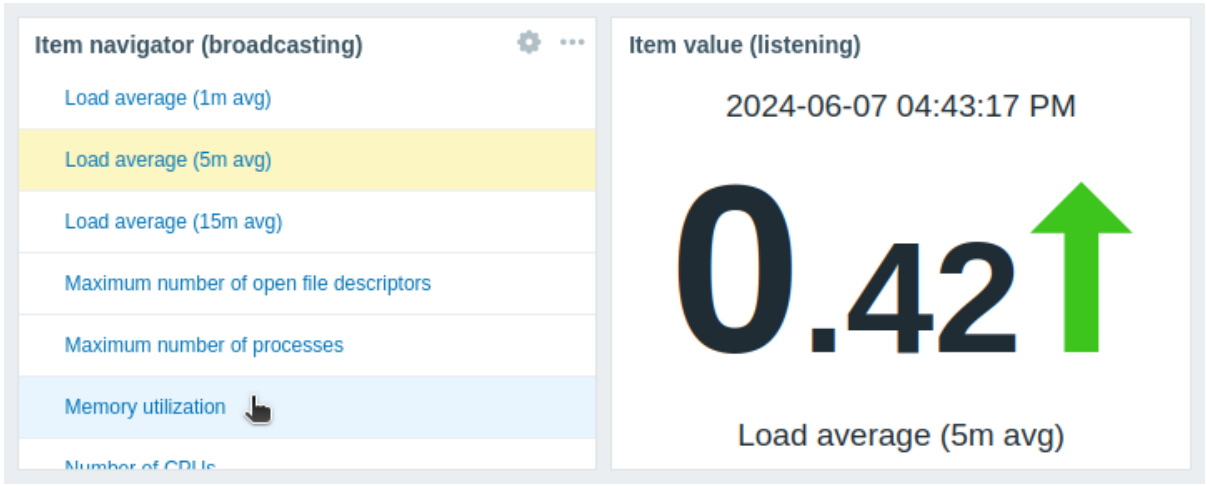
Widget	Diffuse	Reçoit
Journal des actions	-	Période de temps
Horloge	-	-
Statut de découverte	-	-
Graphiques favoris	-	-
Cartes favorites	-	-
Jauge	-	Hôtes, Éléments
Geomap	Hôtes	Groupes d'hôtes, Hôtes
Graphique	Période de temps, Éléments	Hôtes, Éléments, Période de temps
Graphique (classique)	Période de temps	Hôtes, Éléments, Graphiques, Période de temps
Prototype de graphique	Période de temps	Hôtes, Période de temps
Honeycomb	Hôtes, Éléments	Groupes d'hôtes, Hôtes
Disponibilité des hôtes	-	Groupes d'hôtes
Carte d'hôte	-	Hôtes
Navigateur d'hôtes	Hôtes	Groupes d'hôtes
Carte d'élément	-	Hôtes, Éléments, Période de temps
Historique des éléments	Éléments	Hôtes, Période de temps
Navigateur d'éléments	Éléments	Groupes d'hôtes, Hôtes
Valeur d'élément	-	Hôtes, Éléments, Période de temps
Carte	Groupes d'hôtes, Hôtes	Cartes
Arborescence de navigation des cartes	Cartes	-
Diagramme circulaire	-	Éléments, Période de temps
Hôtes à problème	Groupes d'hôtes	Groupes d'hôtes, Hôtes
Problèmes	Événements	Groupes d'hôtes, Hôtes
Problèmes par sévérité	Groupes d'hôtes	Groupes d'hôtes, Hôtes
Nuage de points	Période de temps, Éléments	Hôtes, Éléments, Période de temps
Rapport SLA	-	-
Informations système	-	-
Top hosts	Hôtes	Groupes d'hôtes, Hôtes
Top items	-	Groupes d'hôtes, Hôtes
Top triggers	-	Période de temps
Vue d'ensemble des déclencheurs	-	Groupes d'hôtes, Hôtes
URL	-	Hôtes
Supervision web	Groupes d'hôtes	Groupes d'hôtes, Hôtes

Les widgets diffèrent dans la manière dont ils **diffusent** des données vers d'autres widgets.

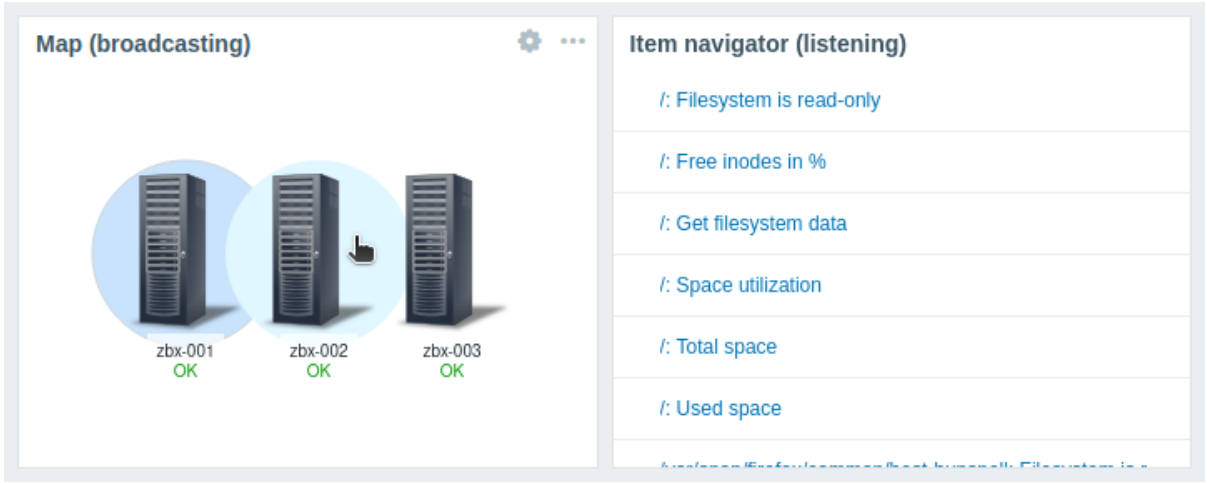
Les widgets qui prennent en charge la diffusion commencent à le faire automatiquement dès leur création. Par exemple, le widget *Graph* diffuse immédiatement les données de période de temps aux widgets à l'écoute.



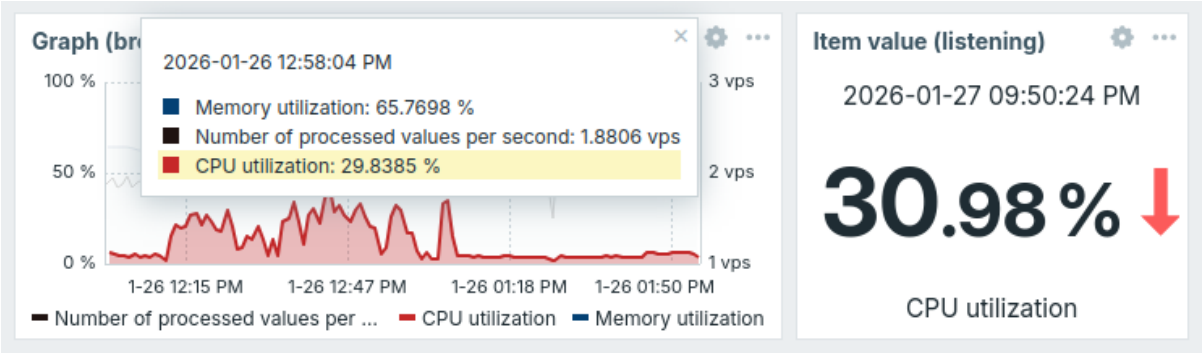
Les widgets qui prennent en charge la sélection d'entités diffusent automatiquement les données de la première entité disponible. Par exemple, le widget *Item navigator* diffuse les données du premier élément de sa liste d'éléments. Lorsqu'un autre élément est sélectionné, il diffuse les données de cet élément. Au survol de la souris, l'élément est mis en évidence en bleu clair ; lorsqu'il est sélectionné, il est mis en évidence en jaune.



Le widget *Map* se comporte de manière similaire, en diffusant les données du premier élément le plus proche du coin supérieur gauche de la zone visible du widget (pour le widget *Geomap*, le plus proche du centre). Lorsqu'un autre élément est sélectionné, il diffuse les données de cet élément. Au survol de la souris, l'élément est mis en évidence en bleu clair ; lorsqu'il est sélectionné, il est mis en évidence en bleu foncé.

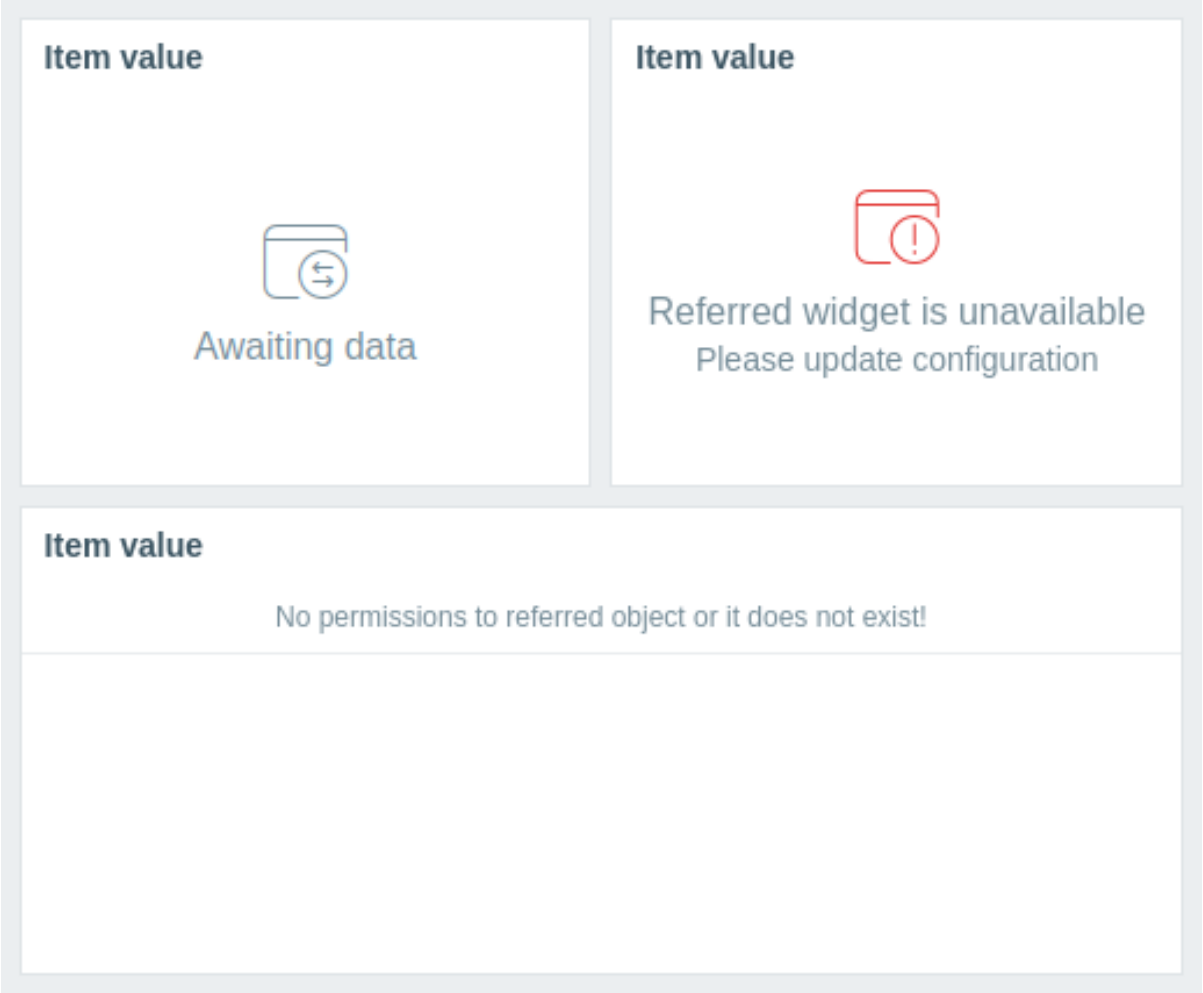


Les widgets tels que *Graph* et *Scatter plot* peuvent diffuser des données d'élément lorsqu'un élément est sélectionné dans leurs infobulles :



Les widgets ont également des comportements spécifiques lorsqu'ils **reçoivent** des données d'autres widgets :

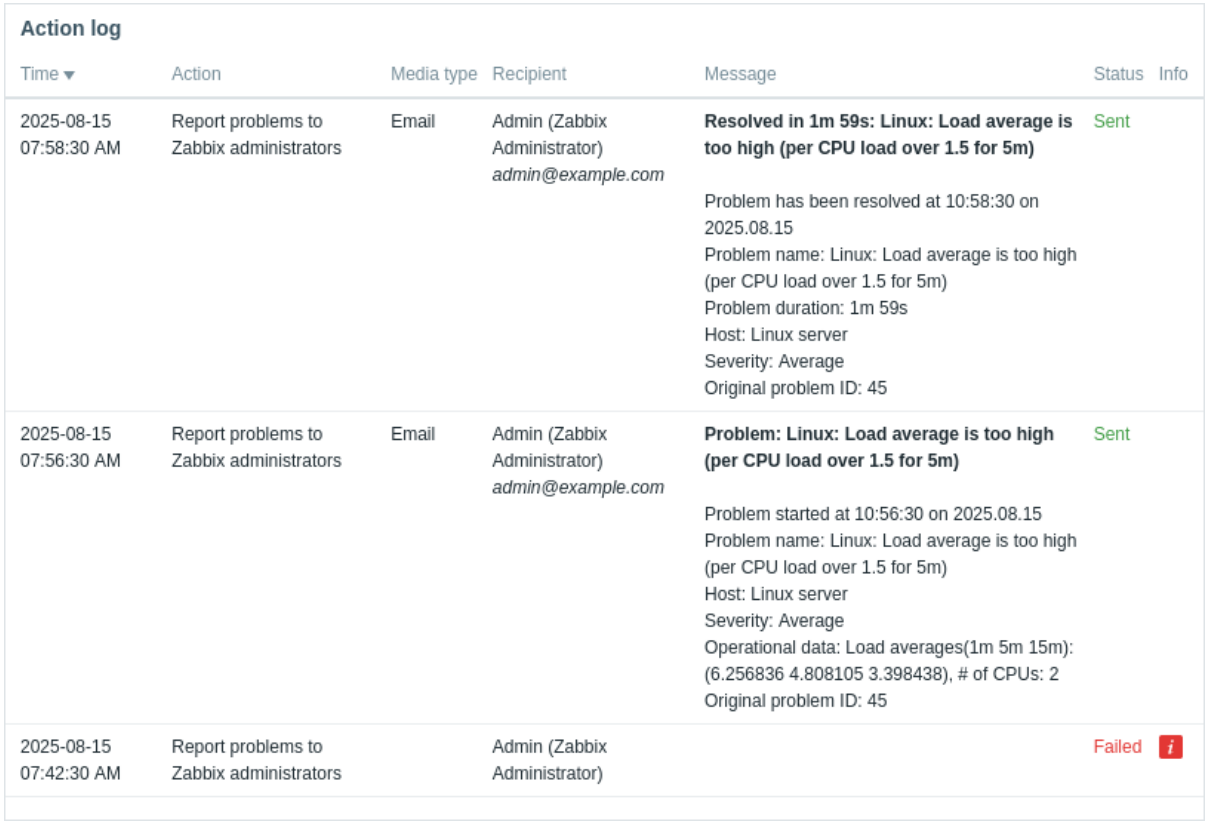
- Si le widget source de données ne diffuse pas de données, le widget à l'écoute passe à l'état *Awaiting data*.
- Si le widget source de données a été supprimé, remplacé par un widget incompatible ou déplacé vers une autre page du tableau de bord, le widget à l'écoute passe à l'état *Referred widget is unavailable*.
- Si l'hôte spécifié dans la source de données (widget ou tableau de bord) ne possède pas l'entité configurée dans le widget à l'écoute (élément, graphique, carte, etc.) ou si l'utilisateur n'a pas les autorisations nécessaires pour accéder à l'hôte, le widget à l'écoute affiche le message suivant : *"No permissions to referred object or it does not exist!"*



Journal des actions

Aperçu

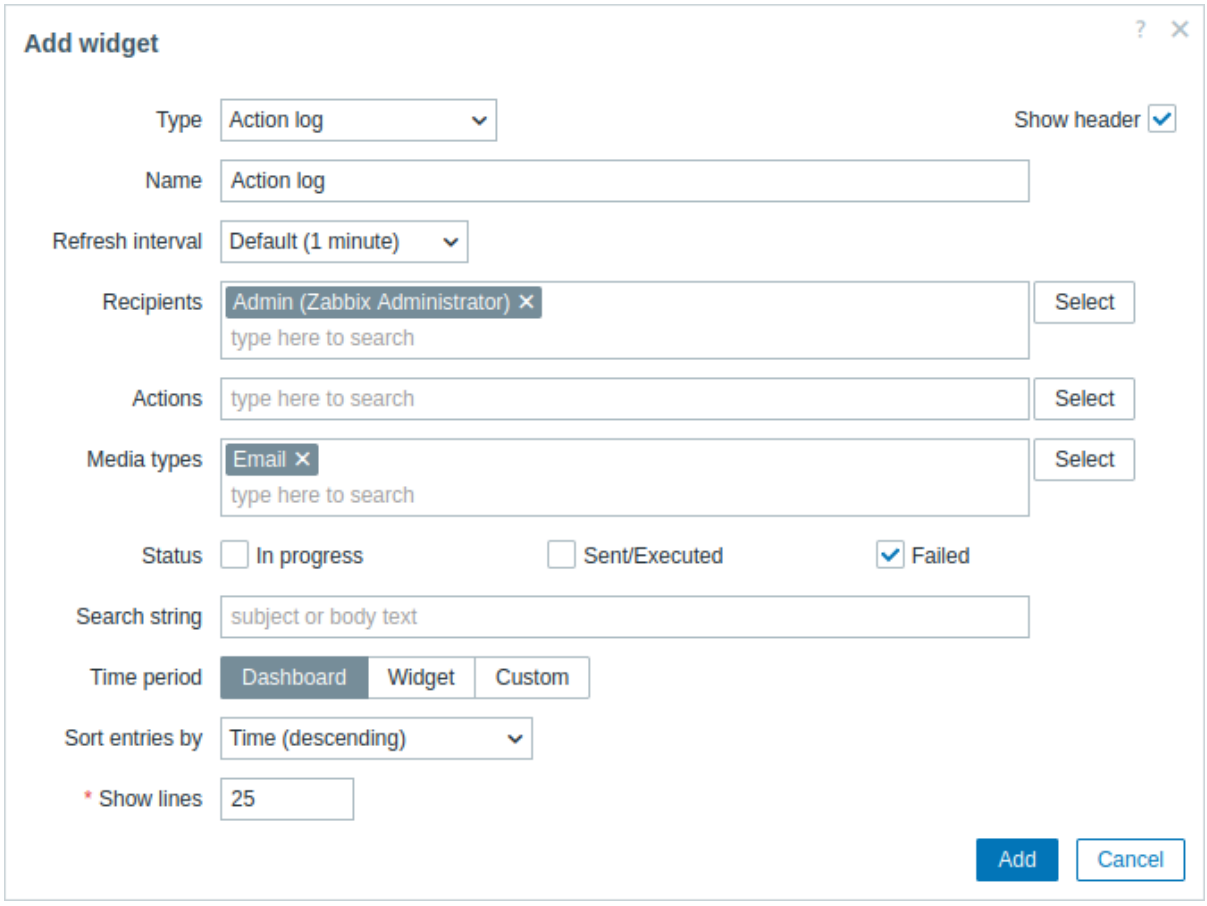
Le widget *Journal des actions* affiche les détails des opérations (notifications, commandes distantes) exécutées dans le cadre d'une action. Le journal détaillé facilite l'audit, le dépannage et le suivi de l'exécution de chaque action.



Il affiche les mêmes données que *Rapports* > *Journal des actions* et peut afficher jusqu'à 1000 enregistrements.

Configuration

Pour configurer, sélectionnez *Journal des actions* comme type :



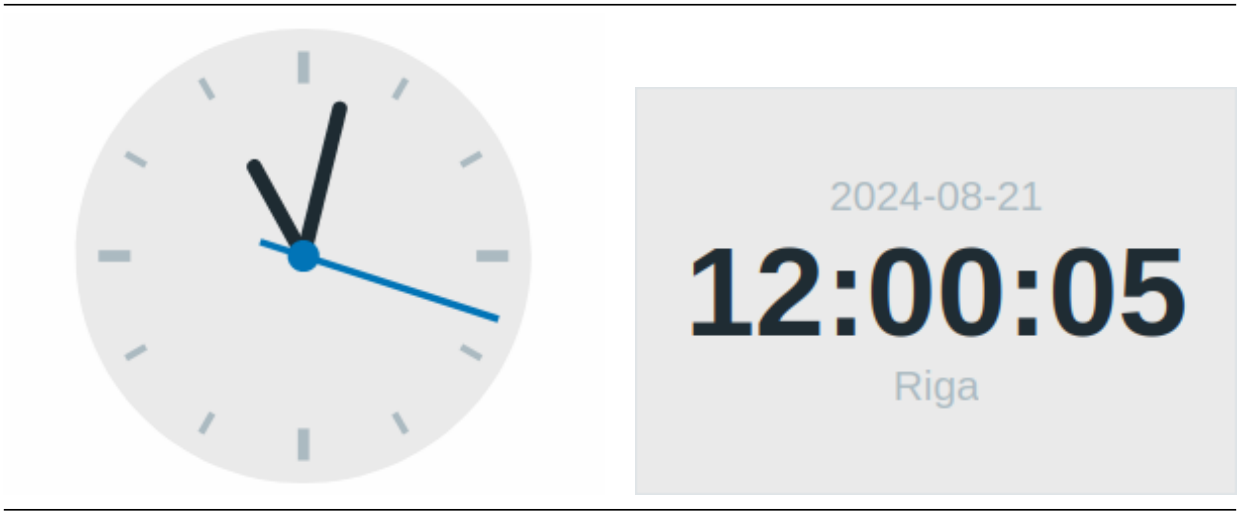
En plus des paramètres communs à tous les widgets, vous pouvez définir les options spécifiques suivantes :

Destinataires	Filtrer les entrées par destinataires. Ce champ prend en charge l'autocomplétion ; ainsi, lorsque vous commencez à saisir le nom d'un destinataire, une liste déroulante des destinataires correspondants est proposée. Si aucun destinataire n'est sélectionné, les détails des opérations d'action pour tous les destinataires seront affichés.
Actions	Filtrer les entrées par actions. Ce champ prend en charge l'autocomplétion ; ainsi, lorsque vous commencez à saisir le nom d'une action, une liste déroulante des actions correspondantes est proposée. Si aucune action n'est sélectionnée, les détails des opérations d'action pour toutes les actions seront affichés.
Types de média	Filtrer les entrées par types de média. Ce champ prend en charge l'autocomplétion ; ainsi, lorsque vous commencez à saisir le nom d'un type de média, une liste déroulante des types de média correspondants est proposée. Si aucun type de média n'est sélectionné, les détails des opérations d'action pour tous les types de média seront affichés.
Statut	Cochez la case pour filtrer les entrées selon le statut correspondant : **En cours** - les opérations d'action en cours sont affichées ; **Envoyé/Exécuté** - les opérations d'action ayant envoyé une notification ou ayant été exécutées sont affichées ; **Échoué** - les opérations d'action ayant échoué sont affichées.
Chaîne de recherche	Filtrer les entrées selon le contenu du message/de la commande distante. Si vous saisissez une chaîne ici, seules les opérations d'action dont le message/la commande distante contient la chaîne saisie seront affichées. Les macros ne sont pas résolues.
Période	Filtrer les entrées par période. Sélectionnez la source de données pour la période : **Tableau de bord** - utiliser le sélecteur de période du tableau de bord ; **Widget** - utiliser un widget compatible (défini dans le paramètre *Widget*) ; **Personnalisée** - utiliser une période personnalisée définie dans les paramètres *De* et *À* ; si elle est définie, une icône d'horloge sera affichée dans le coin supérieur droit du widget, indiquant l'heure définie au survol de la souris. Notez que, quelle que soit la configuration de la *Période* du widget, les widgets compatibles peuvent toujours l'utiliser comme source de données pour la période.
Widget	Saisissez ou sélectionnez un widget compatible comme source de données pour la période. Ce paramètre est disponible si *Période* est défini sur "Widget".
De	Saisissez ou sélectionnez le début de la période. La syntaxe de temps relatif (`now`, `now/d`, `now/w-1w`, etc.) est prise en charge. Ce paramètre est disponible si *Période* est défini sur "Personnalisée".
À	Saisissez ou sélectionnez la fin de la période. La syntaxe de temps relatif (`now`, `now/d`, `now/w-1w`, etc.) est prise en charge. Ce paramètre est disponible si *Période* est défini sur "Personnalisée".
Trier les entrées par	Trier les entrées par : **Heure** (ordre décroissant ou croissant) ; **Type** (ordre décroissant ou croissant) ; **Statut** (ordre décroissant ou croissant) ; **Destinataire** (ordre décroissant ou croissant).
Afficher les lignes	Définissez combien de lignes du journal des actions seront affichées dans le widget.

Horloge

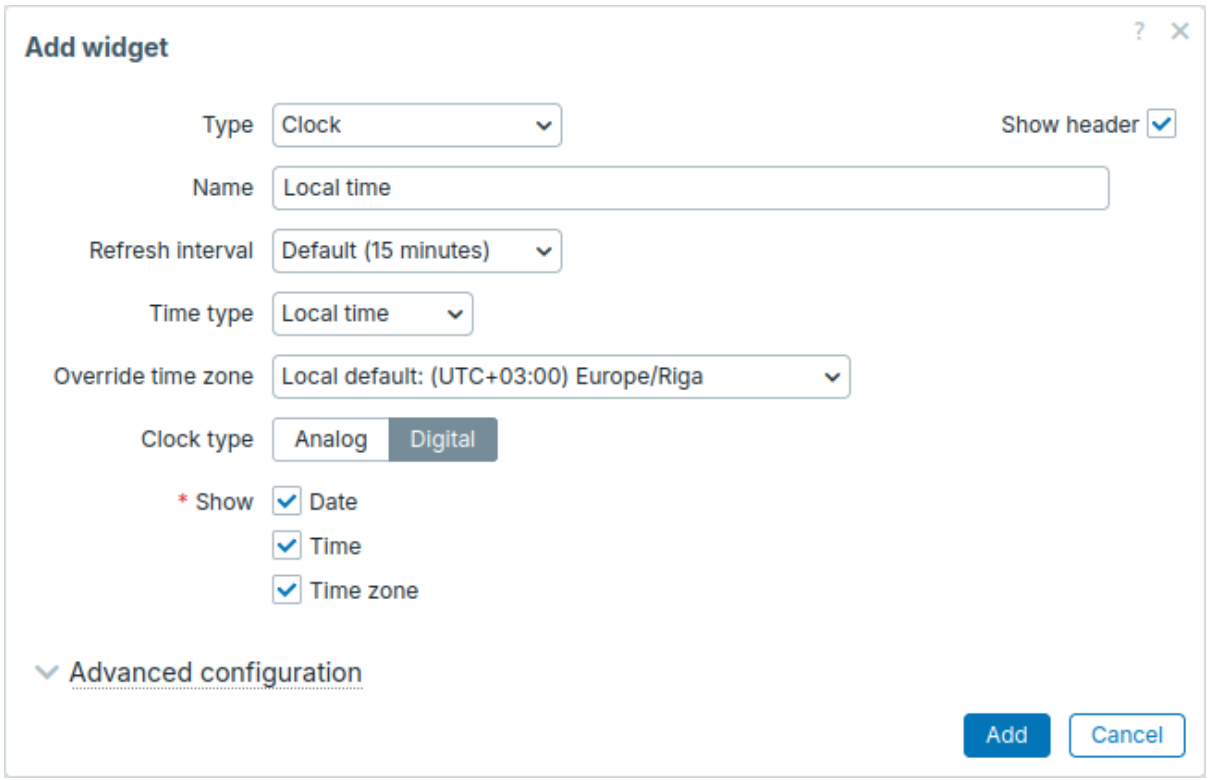
Vue d'ensemble

Le widget *Clock* affiche l'heure locale, du serveur ou de l'hôte au format analogique ou numérique.



Configuration

Pour configurer le widget, sélectionnez *Horloge* comme type :



En plus des paramètres communs à tous les widgets, vous pouvez définir les options spécifiques suivantes :

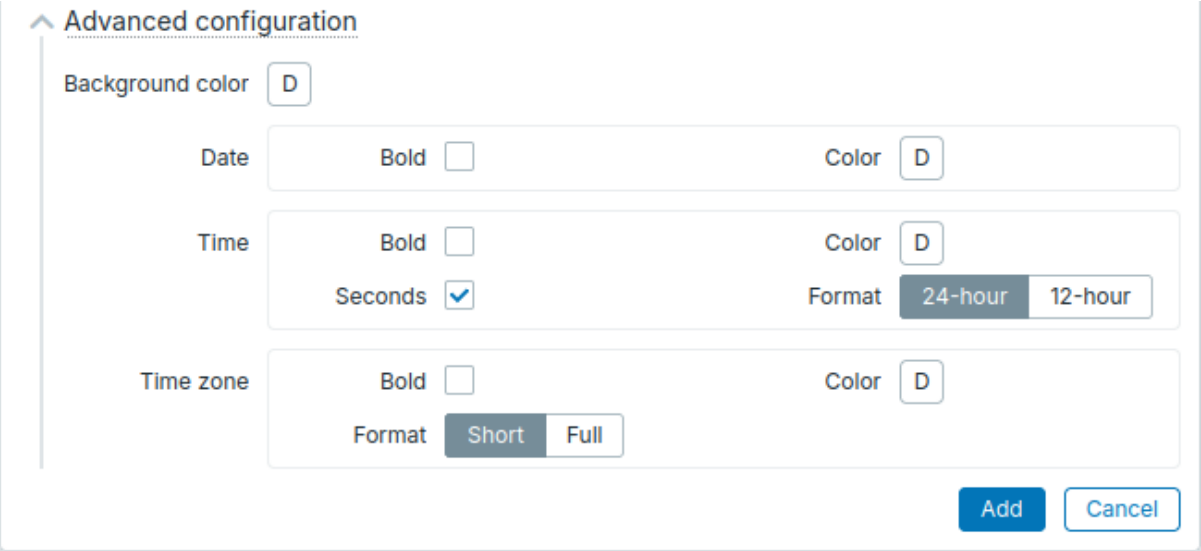
Type d'heure	Sélectionnez la source de l'heure : **Heure locale** - heure système de l'appareil exécutant le navigateur ; **Heure du serveur** - fuseau horaire défini globalement ou pour l'utilisateur Zabbix ; **Heure de l'hôte** - heure récupérée depuis un élément d'un hôte.
Élément	Sélectionnez l'élément dont la clé `system.localtime[local]` permet d'afficher l'heure de l'hôte. Ce paramètre est disponible si le *Type d'heure* est défini sur « Heure de l'hôte ».
Type d'horloge	Sélectionnez le type d'horloge (analogique ou numérique).
Remplacer le fuseau horaire	Sélectionnez le fuseau horaire. Ce paramètre est disponible pour les types d'heure *Heure locale* et *Heure du serveur*.
Afficher	Sélectionnez une ou plusieurs unités d'information (date, heure, fuseau horaire) à afficher dans l'horloge numérique. La taille des unités s'adapte automatiquement aux unités affichées, à la taille du widget et à la mise à l'échelle de l'affichage. Ce paramètre est disponible si le *Type d'horloge* est défini sur « Numérique ».

Configuration avancée	Cliquez sur l'en-tête *Configuration avancée* pour afficher les options de configuration avancée de l'horloge numérique. Cette section est disponible si le *Type d'horloge* est défini sur « Numérique ».

Configuration avancée

Les options de configuration avancée sont disponibles dans la section réductible *Configuration avancée*, et uniquement pour les éléments sélectionnés dans le champ *Afficher* (voir ci-dessus).

De plus, la configuration avancée permet de modifier la couleur d'arrière-plan de l'ensemble du widget.

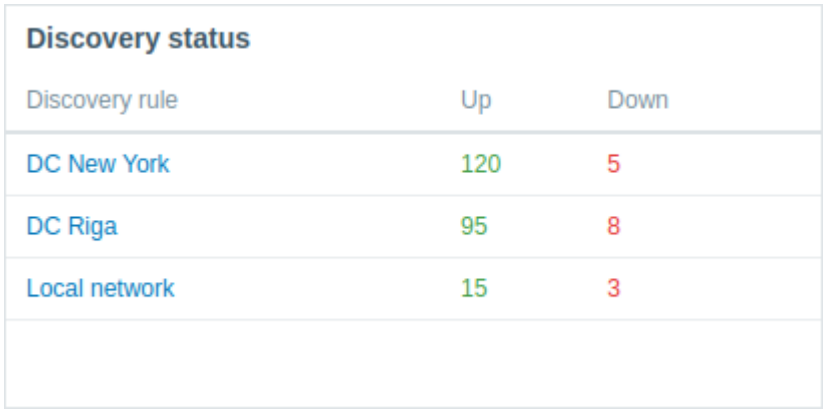


Couleur d'arrière-plan		Sélectionnez la couleur d'arrière-plan dans le sélecteur de couleurs. D correspond à la couleur par défaut (qui dépend du thème de l'interface). Pour rétablir la valeur par défaut, cliquez sur le bouton *Utiliser la valeur par défaut* dans le sélecteur de couleurs.
Date		Cette section est disponible lorsque la case *Date* est sélectionnée pour le paramètre *Afficher*.
	Gras	Cochez la case pour afficher la date en caractères gras.
	Couleur	Sélectionnez la couleur de la date dans le sélecteur de couleurs. D correspond à la couleur par défaut (qui dépend du thème de l'interface). Pour rétablir la valeur par défaut, cliquez sur le bouton *Utiliser la valeur par défaut* dans le sélecteur de couleurs.
Heure		Cette section est disponible lorsque la case *Heure* est sélectionnée pour le paramètre *Afficher*.
	Gras	Cochez la case pour afficher l'heure en caractères gras.
	Couleur	Sélectionnez la couleur de l'heure dans le sélecteur de couleurs. D correspond à la couleur par défaut (qui dépend du thème de l'interface). Pour rétablir la valeur par défaut, cliquez sur le bouton *Utiliser la valeur par défaut* dans le sélecteur de couleurs.
	Secondes	Cochez la case pour afficher les secondes. Sinon, seules les heures et les minutes seront affichées.
	Format	Sélectionnez l'affichage de l'heure au format 24 heures ou 12 heures.
Fuseau horaire		Cette section est disponible lorsque la case *Fuseau horaire* est sélectionnée pour le paramètre *Afficher*.
	Gras	Cochez la case pour afficher le fuseau horaire en caractères gras.
	Couleur	Sélectionnez la couleur du fuseau horaire dans le sélecteur de couleurs. D correspond à la couleur par défaut (qui dépend du thème de l'interface). Pour rétablir la valeur par défaut, cliquez sur le bouton *Utiliser la valeur par défaut* dans le sélecteur de couleurs.
	Format	Sélectionnez l'affichage du fuseau horaire au format abrégé (par exemple, `New York`) ou au format complet (par exemple, `(UTC-04:00) America/New York`).

Statut de découverte

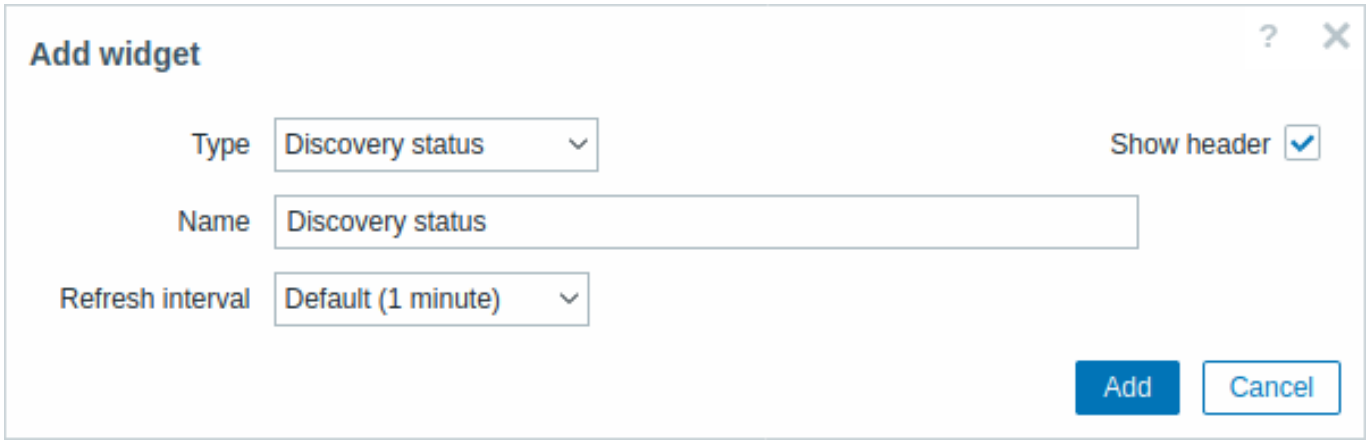
Vue d'ensemble

Le widget *Statut de la découverte* affiche l'état des appareils découverts par les règles de découverte réseau activées. Il aide à surveiller si la découverte réseau fonctionne correctement ainsi que le nombre d'appareils opérationnels ou indisponibles.



Configuration

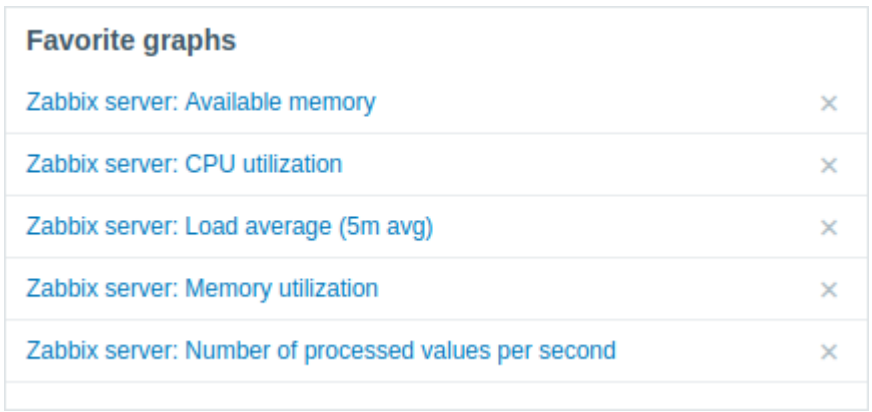
Tous les paramètres de configuration de ce widget sont communs à tous les widgets.



Graphiques favoris

Aperçu

Le widget *Graphiques favoris* affiche une liste alphabétique de raccourcis vers les graphiques marqués comme favoris par l'utilisateur actuel.



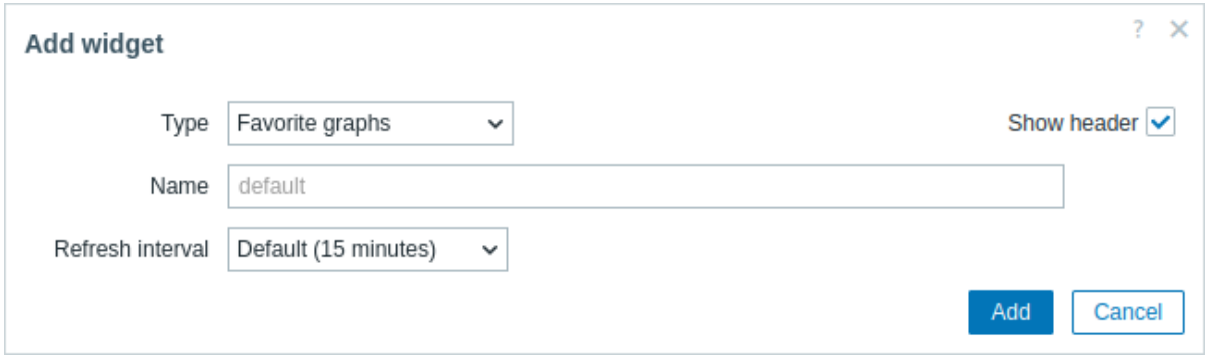
Un clic sur x supprime le graphique de la liste et des favoris.

Les graphiques sont marqués comme favoris en cliquant sur le bouton  *Ajouter aux favoris* lors de l'affichage des graphiques d'éléments dans *Surveillance* > *Dernières données*.



Configuration

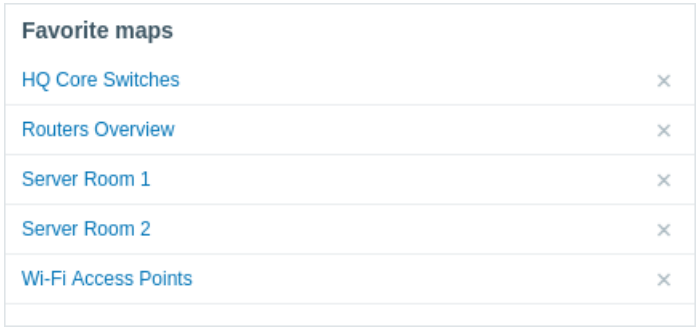
Tous les paramètres de configuration de ce widget sont communs à tous les widgets.



Cartes favorites

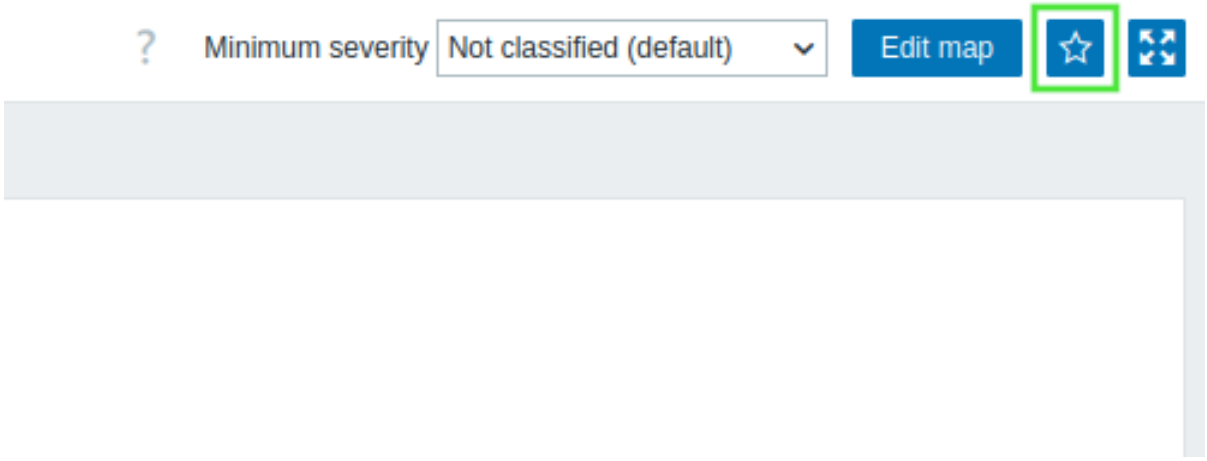
Aperçu

Le widget *Cartes favorites* affiche une liste de raccourcis vers les cartes marquées comme favorites par l'utilisateur actuel, triées par ordre alphabétique.



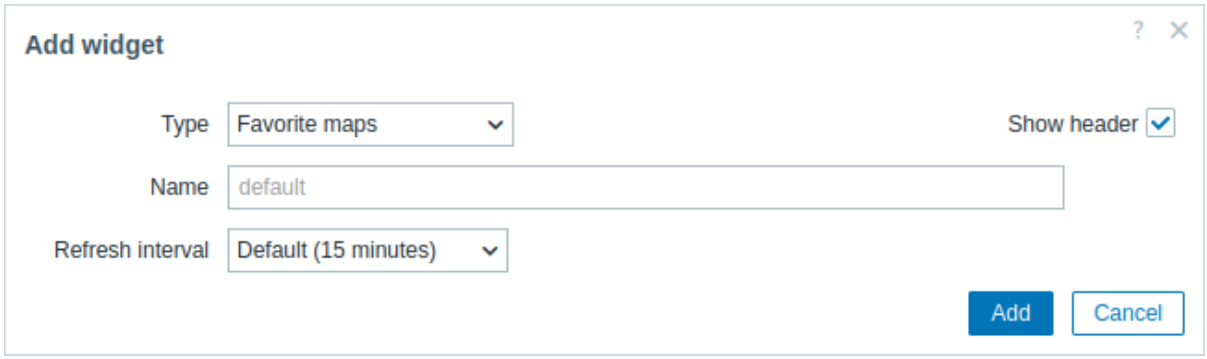
Un clic sur x supprime la carte des favoris.

Les cartes sont marquées comme favorites en cliquant sur le bouton  *Ajouter aux favoris* lors de l'affichage des cartes dans *Monitoring* > *Maps*.



Configuration

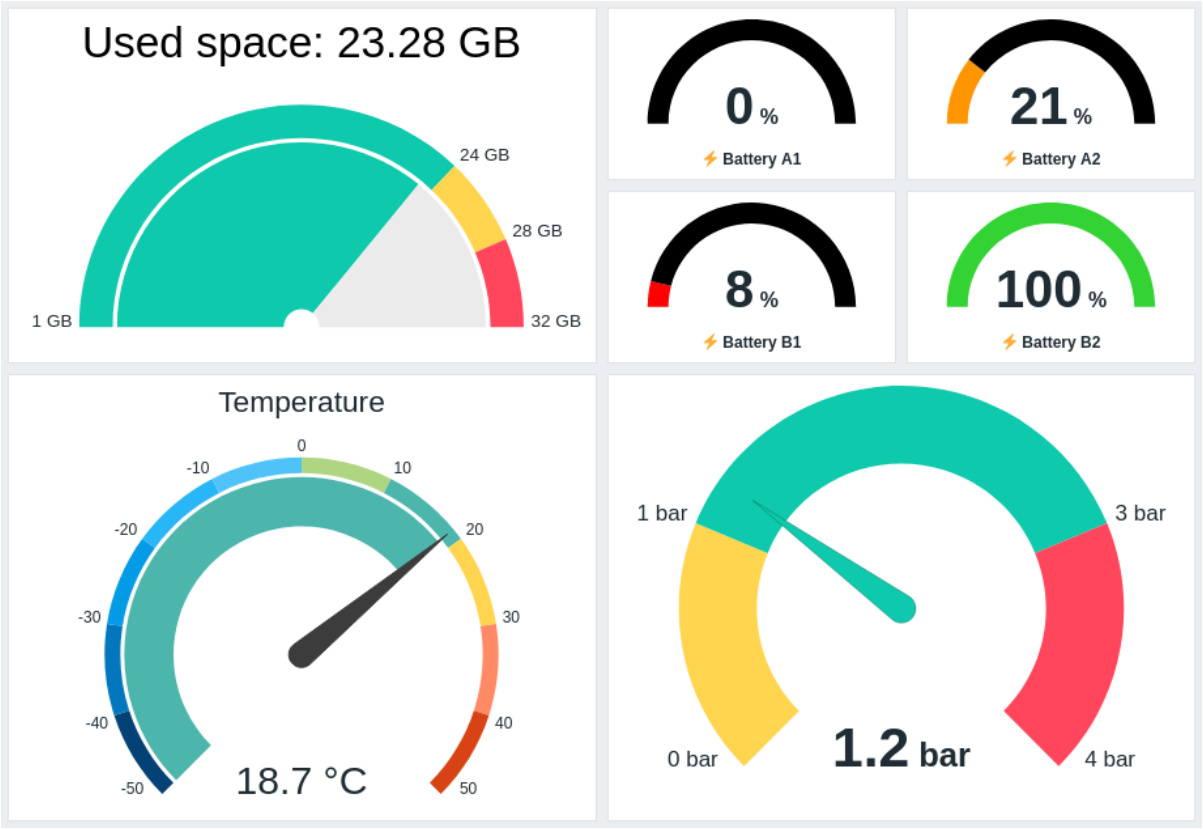
Tous les paramètres de configuration de ce widget sont communs à tous les widgets.



Jauge

Présentation

Le widget *Jauge* affiche la valeur d'un seul élément numérique sous forme de jauge. Il est utile pour surveiller les métriques importantes, visualiser les seuils et détecter les changements soudains.



Vous pouvez configurer le widget pour afficher les éléments suivants :

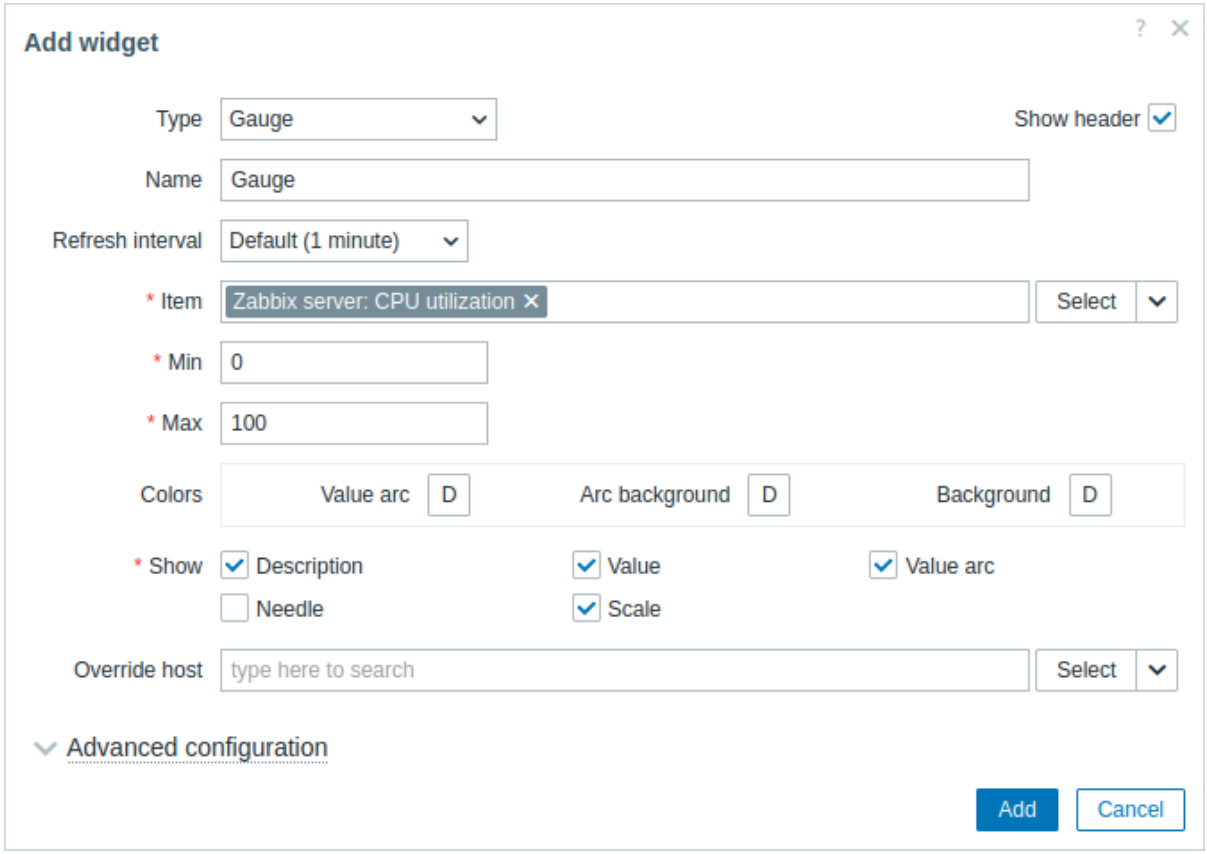
- Description de l'élément (Espace utilisé : {ITEM.LASTVALUE}, Température)
- Valeur et unités de l'élément (21 %, 18.7 °C, 1.2 bar)
- Échelle de la jauge (1GB/24GB/28GB/32GB, -50/-40/-30/etc.)
- Arc de la jauge (arc de la valeur de la jauge et arc des seuils de la jauge)
- Aiguille de la jauge

Cliquer sur le widget ouvre un graphique simple pour l'élément.

Les informations affichées dans le widget *Jauge* peuvent être téléchargées sous forme d'image PNG en sélectionnant l'option *Télécharger l'image* dans le menu *Actions* du widget ⋯, qui apparaît lorsque vous survolez le widget. Selon les paramètres de votre navigateur, l'image est soit enregistrée automatiquement dans le dossier `Downloads`, soit une boîte de dialogue *Enregistrer* s'ouvre, vous permettant de choisir l'emplacement.

Configuration

Pour configurer, sélectionnez *Gauge* comme type :



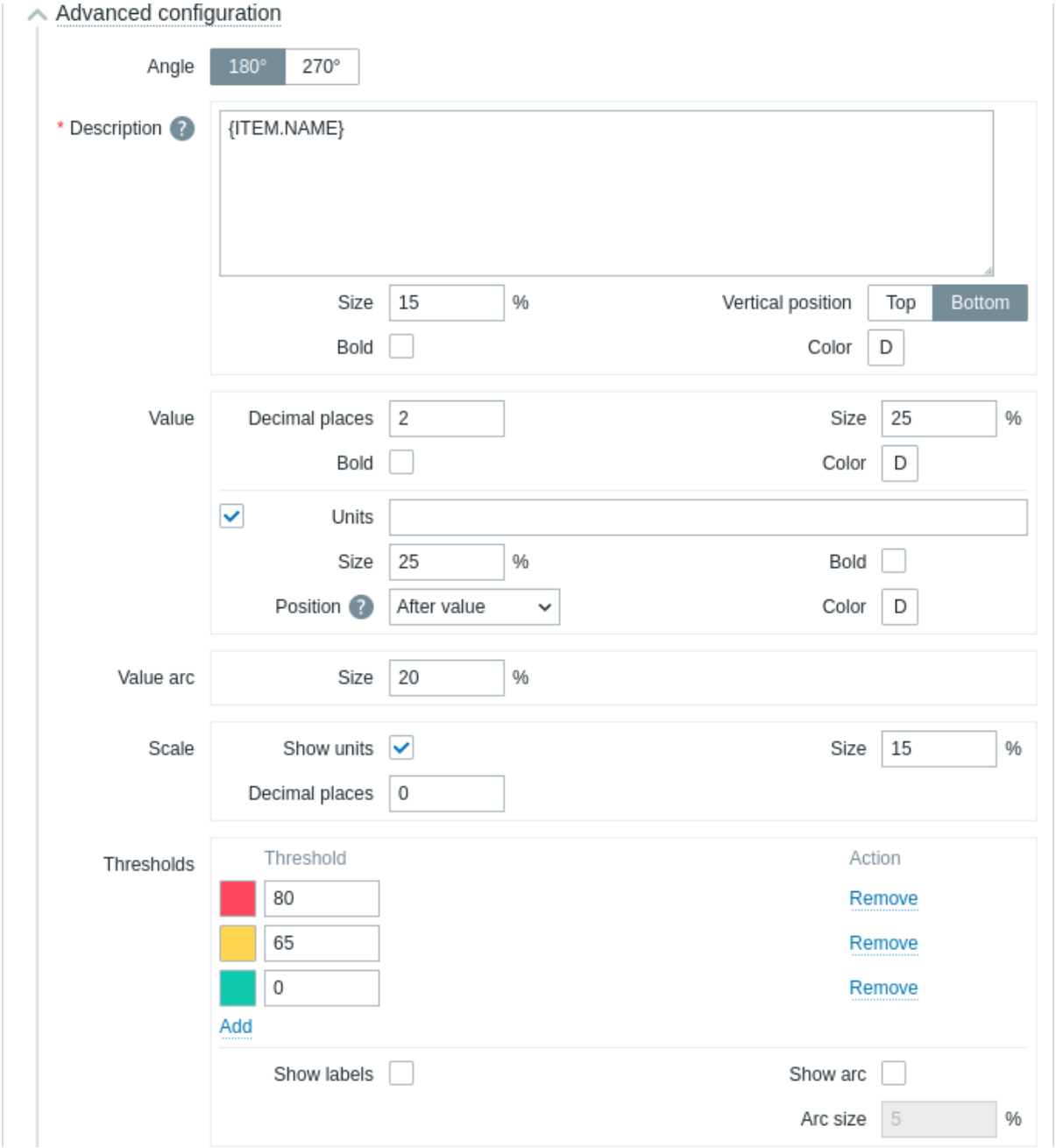
En plus des paramètres communs à tous les widgets, vous pouvez définir les options spécifiques suivantes :

Item	Sélectionnez l'élément. Vous pouvez également sélectionner un widget compatible comme source de données pour les éléments. Ce champ prend en charge l'autocomplétion ; ainsi, lorsque vous commencez à saisir le nom d'un élément, une liste déroulante des éléments correspondants est proposée. Notez que vous ne pouvez sélectionner que des éléments renvoyant des données numériques (à l'exception des données binaires).
Min	Saisissez la valeur minimale de la jauge. Les suffixes (par exemple, "1d", "2w", "4K", "8G") sont pris en charge. Les correspondances de valeurs sont prises en charge.
Max	Saisissez la valeur maximale de la jauge. Les suffixes (par exemple, "1d", "2w", "4K", "8G") sont pris en charge. Les correspondances de valeurs sont prises en charge.
Colors	Sélectionnez la couleur dans le sélecteur de couleurs : **Value arc** - sélectionnez la couleur de l'arc de valeur de la jauge ; **Arc background** - sélectionnez la couleur d'arrière-plan de l'arc de valeur de la jauge et de l'arc des seuils de la jauge ; **Background** - sélectionnez la couleur d'arrière-plan du widget. "D" désigne la couleur par défaut, qui dépend du thème de l'interface. Si des *Thresholds* sont définis, la couleur par défaut de **Value arc** dépend de la couleur du seuil. Pour revenir à la couleur par défaut, cliquez sur le bouton *Use default* dans le sélecteur de couleurs.
Show	Cochez la case pour afficher l'élément de jauge correspondant : description, valeur, arc de valeur, aiguille, échelle (la valeur minimale et la valeur maximale de la jauge au début et à la fin de l'arc de la jauge). Décochez-la pour masquer l'élément. Au moins un élément doit être sélectionné. Notez que l'aiguille et l'échelle de la jauge peuvent être affichées si l'arc de valeur de la jauge ou l'arc des seuils de la jauge (voir les options de configuration avancée) est affiché. Notez également que si l'aiguille de la jauge est affichée, la valeur est placée sous l'aiguille ; si l'aiguille est masquée, la valeur est alignée sur le bas de l'arc de la jauge.

Override host	Sélectionnez un widget compatible ou le sélecteur d'hôte du tableau de bord comme source de données pour les hôtes. Ce paramètre n'est pas disponible lors de la configuration du widget sur un tableau de bord de modèle.
Advanced configuration	Cliquez sur l'en-tête *Advanced configuration* pour afficher les options de configuration avancée. C'est également à cet endroit que vous pourrez ajuster les éléments de jauge sélectionnés dans le champ *Show*.

Configuration avancée

Les options de configuration avancée sont disponibles dans la section repliable *Configuration avancée* :



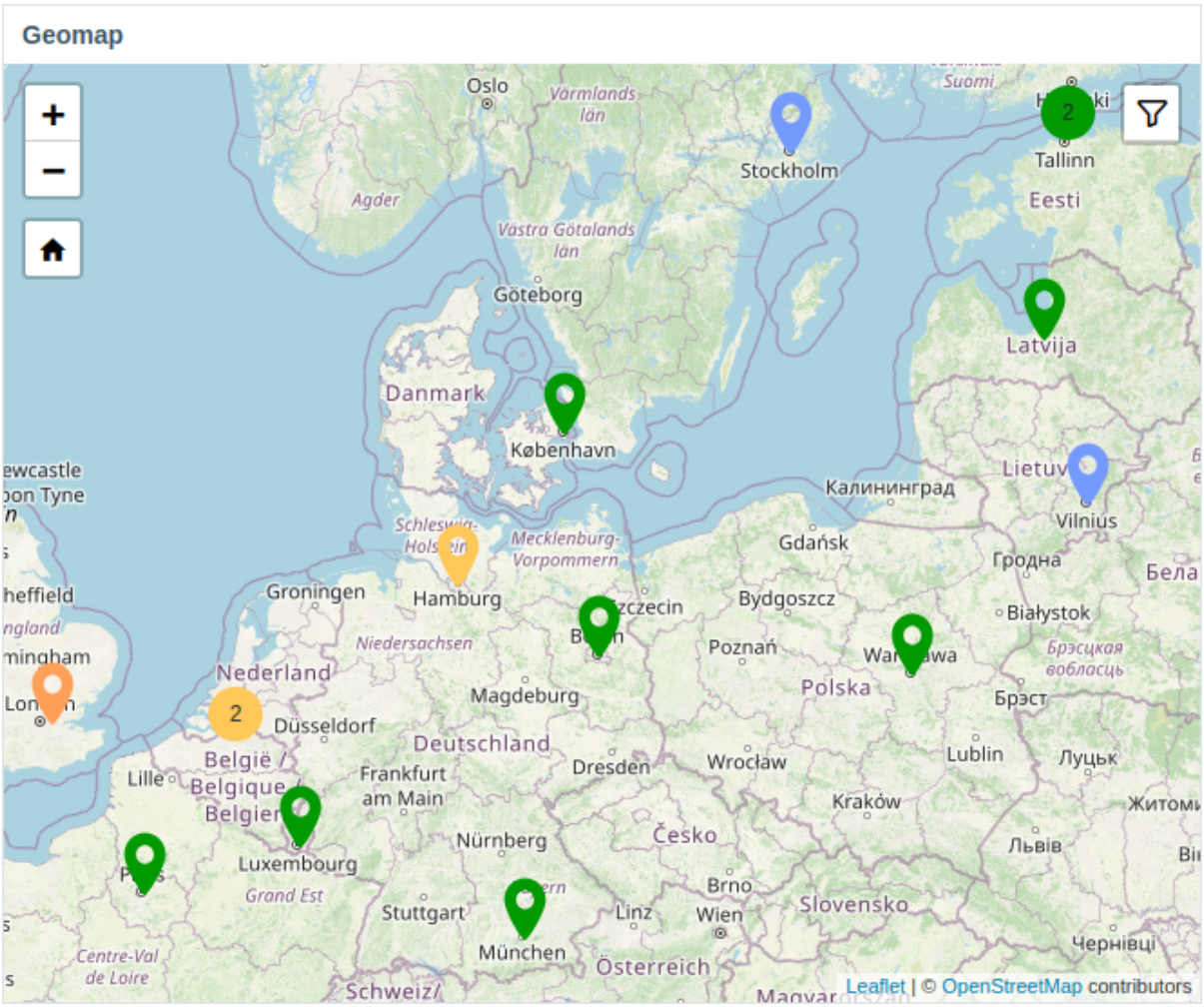
Angle	Sélectionnez l'angle de la jauge (180° ou 270°).
Description	
Description	Saisissez la description de l'élément. Cette description peut remplacer le nom d'élément par défaut. Les descriptions sur plusieurs lignes sont prises en charge. Une combinaison de texte et de macros prises en charge est possible. {HOST.*}, {ITEM.*}, {INVENTORY.*} et les macros utilisateur sont prises en charge.

Size	Saisissez la hauteur de la taille de police pour la description de l'élément (en pourcentage, par rapport à la hauteur totale du widget).
Vertical position	Sélectionnez la position verticale de la description de l'élément (en haut ou en bas, par rapport à l'arc de la jauge).
Bold	Cochez la case pour afficher la description de l'élément en gras.
Color	Sélectionnez la couleur de la description de l'élément dans le sélecteur de couleurs. "D" correspond à la couleur par défaut, qui dépend du thème de l'interface web. Pour revenir à la couleur par défaut, cliquez sur le bouton *Use default* dans le sélecteur de couleurs.
Value	
Decimal places	Saisissez le nombre de décimales à afficher avec la valeur. Cette option n'affecte que les éléments qui renvoient des données numériques (float).
Size	Saisissez la hauteur de la taille de police pour la valeur (en pourcentage, par rapport à la hauteur de l'arc de la jauge).
Bold	Cochez la case pour afficher la valeur en gras.
Color	Sélectionnez la couleur de la valeur dans le sélecteur de couleurs. "D" correspond à la couleur par défaut, qui dépend du thème de l'interface web. Pour revenir à la couleur par défaut, cliquez sur le bouton *Use default* dans le sélecteur de couleurs.
Units	
Units	Cochez la case pour afficher les unités avec la valeur de l'élément. Si vous saisissez un nom d'unité, il remplacera les unités définies dans la configuration de l'élément.
Size	Saisissez la hauteur de la taille de police pour les unités de l'élément (en pourcentage, par rapport à la hauteur de l'arc de la jauge).
Bold	Cochez la case pour afficher les unités de l'élément en gras.
Position	Sélectionnez la position des unités de l'élément (au-dessus, en dessous, avant ou après, par rapport à la valeur de l'élément). Cette option est ignorée pour les unités liées au temps suivantes : unixtime, uptime, s.
Color	Sélectionnez la couleur des unités de l'élément dans le sélecteur de couleurs. "D" correspond à la couleur par défaut, qui dépend du thème de l'interface web. Pour revenir à la couleur par défaut, cliquez sur le bouton *Use default* dans le sélecteur de couleurs.
Value arc	
Arc size	Saisissez la hauteur de taille de l'arc de valeur de la jauge (en pourcentage, par rapport au rayon de l'arc de la jauge).
Needle	
Color	Sélectionnez la couleur de l'aiguille de la jauge dans le sélecteur de couleurs. "D" correspond à la couleur par défaut, qui dépend du thème de l'interface web. Si des *Thresholds* sont définis, la couleur par défaut de l'aiguille dépend de la couleur du seuil. Pour revenir à la couleur par défaut, cliquez sur le bouton *Use default* dans le sélecteur de couleurs.
Scale	
Show units	Cochez la case pour afficher les unités avec les valeurs minimale et maximale de la jauge.
Size	Saisissez la hauteur de la taille de police pour les valeurs minimale et maximale de la jauge (en pourcentage, par rapport à la hauteur de l'arc de la jauge).
Decimal places	Saisissez le nombre de décimales à afficher avec les valeurs minimale et maximale de la jauge. Cette option n'affecte que les éléments qui renvoient des données numériques (float).
Thresholds	
Thresholds	Cliquez sur *Add* pour ajouter un seuil, sélectionnez une couleur de seuil dans le sélecteur de couleurs et indiquez une valeur numérique. La liste des seuils sera triée par ordre croissant lors de l'enregistrement. Notez que les couleurs configurées comme seuils ne s'afficheront correctement que pour les éléments numériques. Les suffixes (par exemple, "1d", "2w", "4K", "8G") sont pris en charge. Les mappages de valeurs sont pris en charge.
Show labels	Cochez la case pour afficher les valeurs de seuil comme étiquettes sur l'échelle de la jauge.
Show arc	Cochez la case pour afficher l'arc des seuils de la jauge.
Arc size	Saisissez la hauteur de taille de l'arc des seuils de la jauge (en pourcentage, par rapport au rayon de l'arc de la jauge).

Geomap

Vue d'ensemble

Le widget *Geomap* affiche les hôtes sous forme de marqueurs sur une carte interactive basée sur Leaflet, les couleurs des marqueurs indiquant le problème le plus grave de chaque hôte.

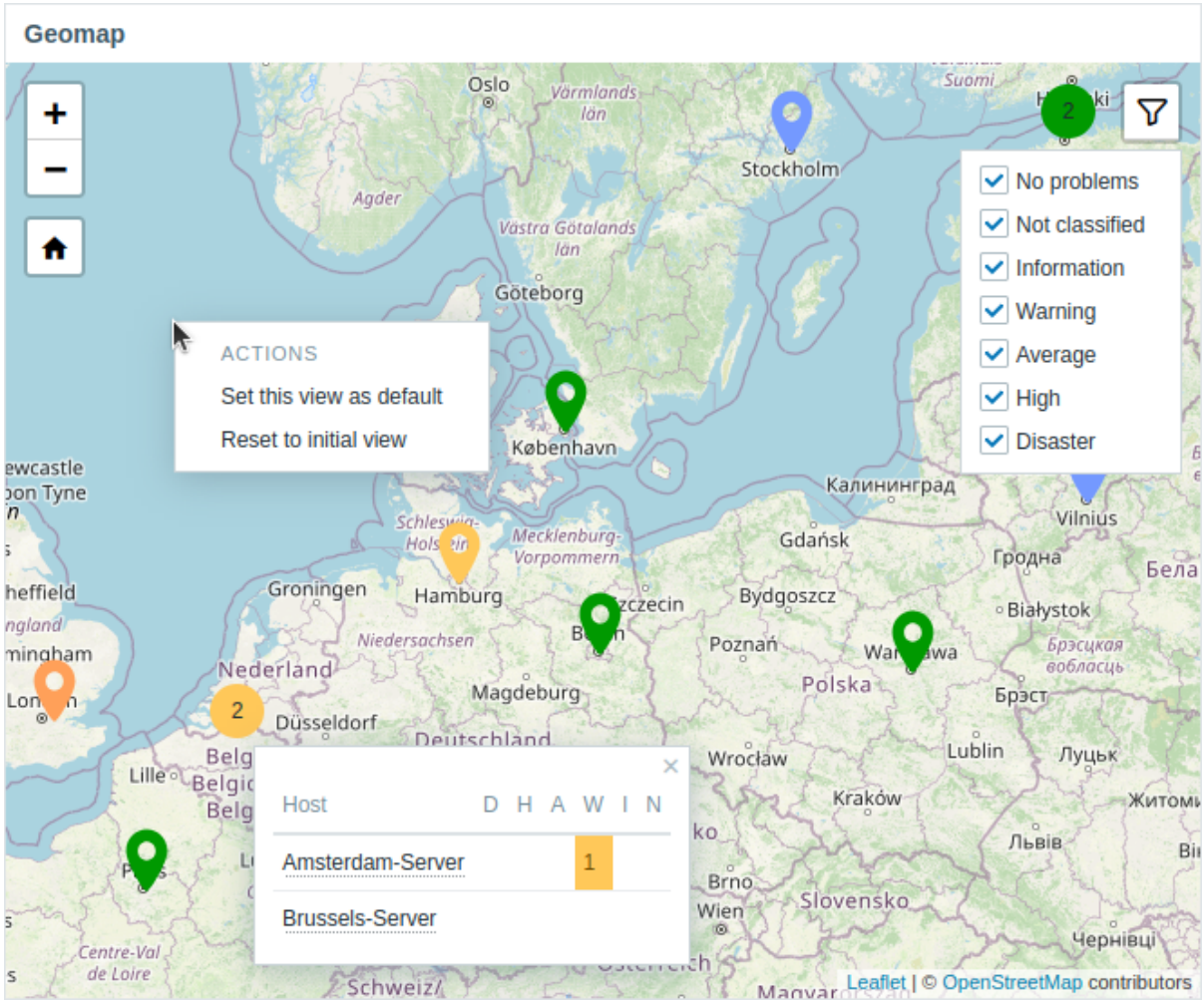


Note:
Zabbix propose plusieurs fournisseurs de tuiles cartographiques prédéfinis et vous permet d'en ajouter des personnalisés dans *Administration* > *General* > *Geographical maps*.

Par défaut, le widget affiche tous les hôtes activés qui possèdent des coordonnées valides (latitude de -90 à 90, longitude de -180 à 180) dans leur inventaire.

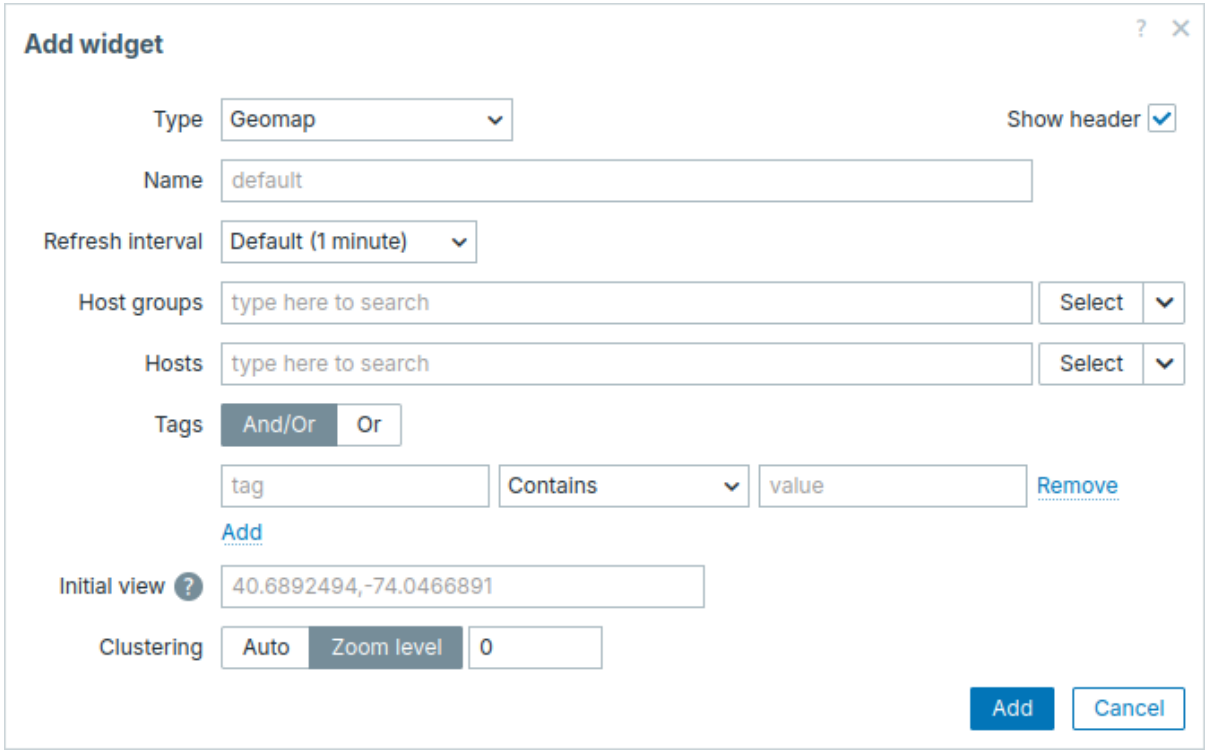
Utilisez les options suivantes pour interagir avec le widget :

- Cliquer sur un marqueur affiche le nom de l'hôte et les problèmes non résolus ; cliquer sur le nom de l'hôte ouvre le menu de l'hôte.
- Filtrez les hôtes par gravité du problème à l'aide de l'icône de filtre dans le coin supérieur droit.
- Cliquez avec le bouton droit sur la carte pour définir ou réinitialiser la vue par défaut ; utilisez l'icône d'accueil pour revenir à la vue par défaut.



Configuration

Pour ajouter le widget, sélectionnez *Geomap* comme type.



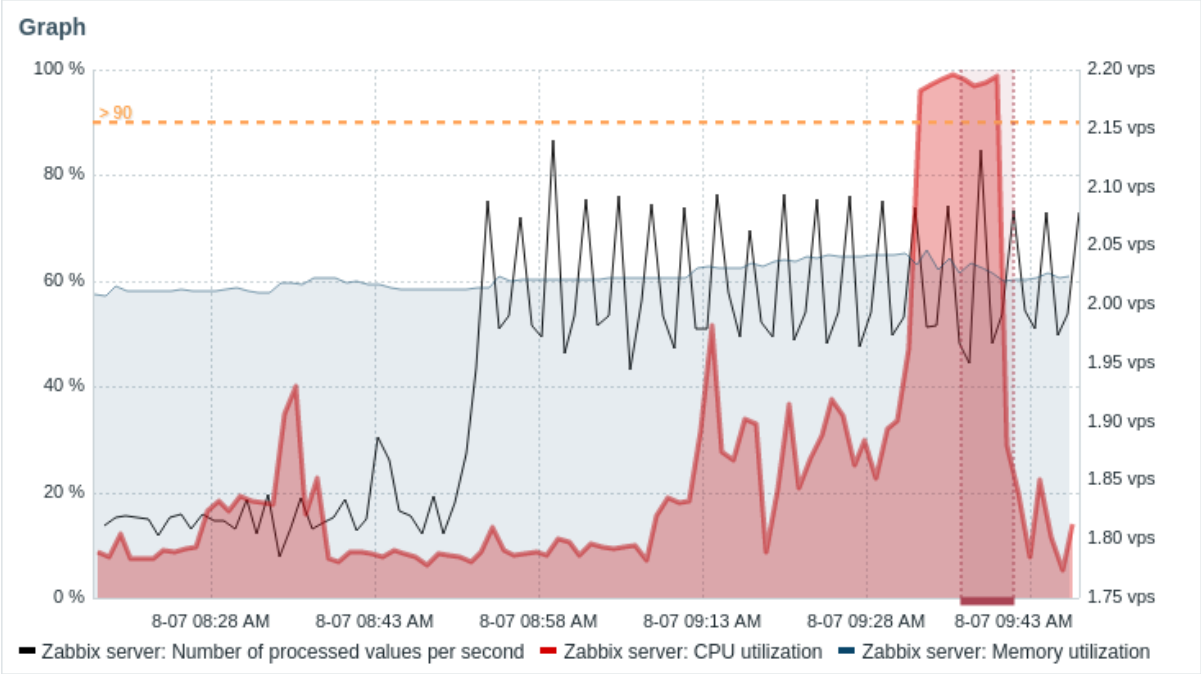
En plus des paramètres communs à tous les widgets, vous pouvez définir les options spécifiques suivantes :

Groupes d'hôtes	Sélectionnez les groupes d'hôtes à afficher sur la carte. Vous pouvez également sélectionner un widget compatible comme source de données pour les groupes d'hôtes. Ce champ prend en charge l'autocomplétion ; ainsi, lorsque vous commencez à saisir le nom d'un groupe, une liste déroulante des groupes correspondants est proposée. Si rien n'est sélectionné dans les champs *Groupes d'hôtes* et *Hôtes*, tous les hôtes ayant des coordonnées valides seront affichés. Ce paramètre n'est pas disponible lors de la configuration du widget sur un tableau de bord de modèle.
Hôtes	Sélectionnez les hôtes à afficher sur la carte. Vous pouvez également sélectionner un widget compatible ou le tableau de bord comme source de données pour les hôtes. Ce champ prend en charge l'autocomplétion ; ainsi, lorsque vous commencez à saisir le nom d'un hôte, une liste déroulante des hôtes correspondants est proposée. Si rien n'est sélectionné dans les champs *Groupes d'hôtes* et *Hôtes*, tous les hôtes ayant des coordonnées valides seront affichés. Ce paramètre n'est pas disponible lors de la configuration du widget sur un tableau de bord de modèle.
Tags	Spécifiez des tags pour limiter le nombre d'hôtes affichés dans le widget. Il est possible d'inclure comme d'exclure des tags et des valeurs de tag spécifiques. Plusieurs conditions peuvent être définies. La correspondance des noms de tag est toujours sensible à la casse. Plusieurs opérateurs sont disponibles pour chaque condition : **Existe** - inclure les noms de tag spécifiés ; **Égal à** - inclure les noms de tag et les valeurs spécifiés (sensible à la casse) ; **Contient** - inclure les noms de tag spécifiés lorsque les valeurs de tag contiennent la chaîne saisie (correspondance de sous-chaîne, insensible à la casse) ; **N'existe pas** - exclure les noms de tag spécifiés ; **N'est pas égal à** - exclure les noms de tag et les valeurs spécifiés (sensible à la casse) ; **Ne contient pas** - exclure les noms de tag spécifiés lorsque les valeurs de tag contiennent la chaîne saisie (correspondance de sous-chaîne, insensible à la casse). Deux types de calcul sont disponibles pour les conditions : **Et/Ou** - toutes les conditions doivent être remplies ; les conditions ayant le même nom de tag seront regroupées par la condition *Ou* ; **Ou** - il suffit qu'une seule condition soit remplie. Ce paramètre n'est pas disponible lors de la configuration du widget sur un tableau de bord de modèle.
Vue initiale	Coordonnées du centre séparées par des virgules et niveau de zoom facultatif à afficher lors du chargement initial du widget, au format `<latitude>,<longitude>,<zoom>` Si le zoom initial est spécifié, le widget Geomap est chargé avec le niveau de zoom indiqué. Sinon, le zoom initial est calculé comme la moitié du zoom max pour le fournisseur de tuiles concerné. La vue initiale est ignorée si la vue par défaut est définie (voir ci-dessous). Exemples : 40.6892494,-74.0466891 40.6892494,-122.0466891
Regroupement	Indiquez comment les marqueurs d'hôtes proches sont combinés en un seul marqueur avec un compteur : **Auto** - les marqueurs sont combinés automatiquement ; **Niveau de zoom** - les marqueurs sont combinés en fonction du niveau de zoom spécifié (0 - zoom arrière maximal, 30 - zoom avant maximal). Par exemple, si le paramètre est défini sur 0, les marqueurs ne sont combinés à aucun niveau de zoom ; s'il est défini sur 15, les marqueurs sont combinés à partir du zoom arrière maximal jusqu'au niveau de zoom 15 ; s'il est défini sur 30, les marqueurs sont combinés à tous les niveaux de zoom. Notez que le niveau de zoom avant maximal est déterminé par le fournisseur de tuiles cartographiques.

Graphe

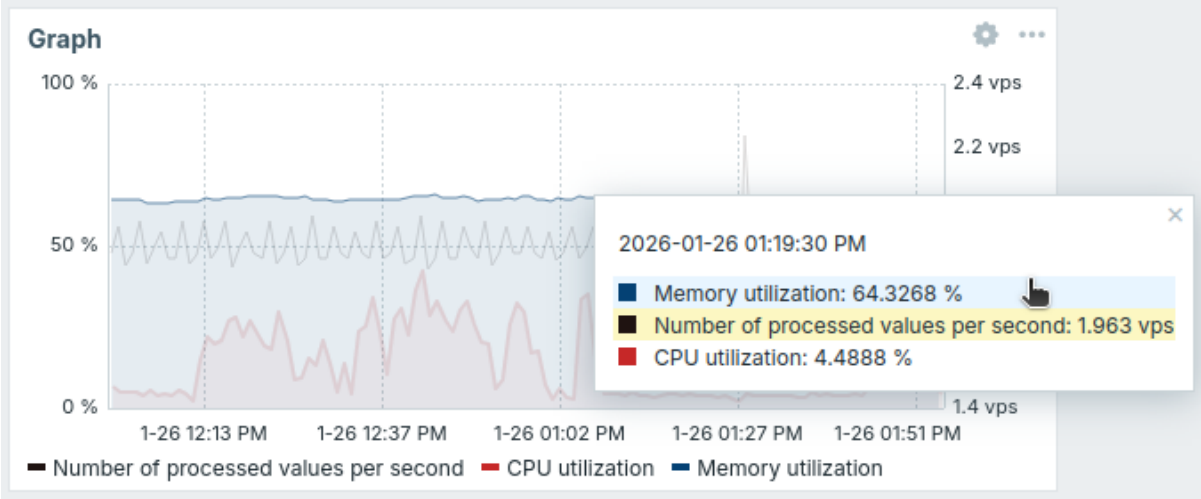
Présentation

Le widget *Graphique* affiche les données numériques des éléments sous forme de graphique vectoriel. Il peut vous aider à suivre les tendances, à repérer les problèmes et à comparer les valeurs au fil du temps et entre les hôtes.

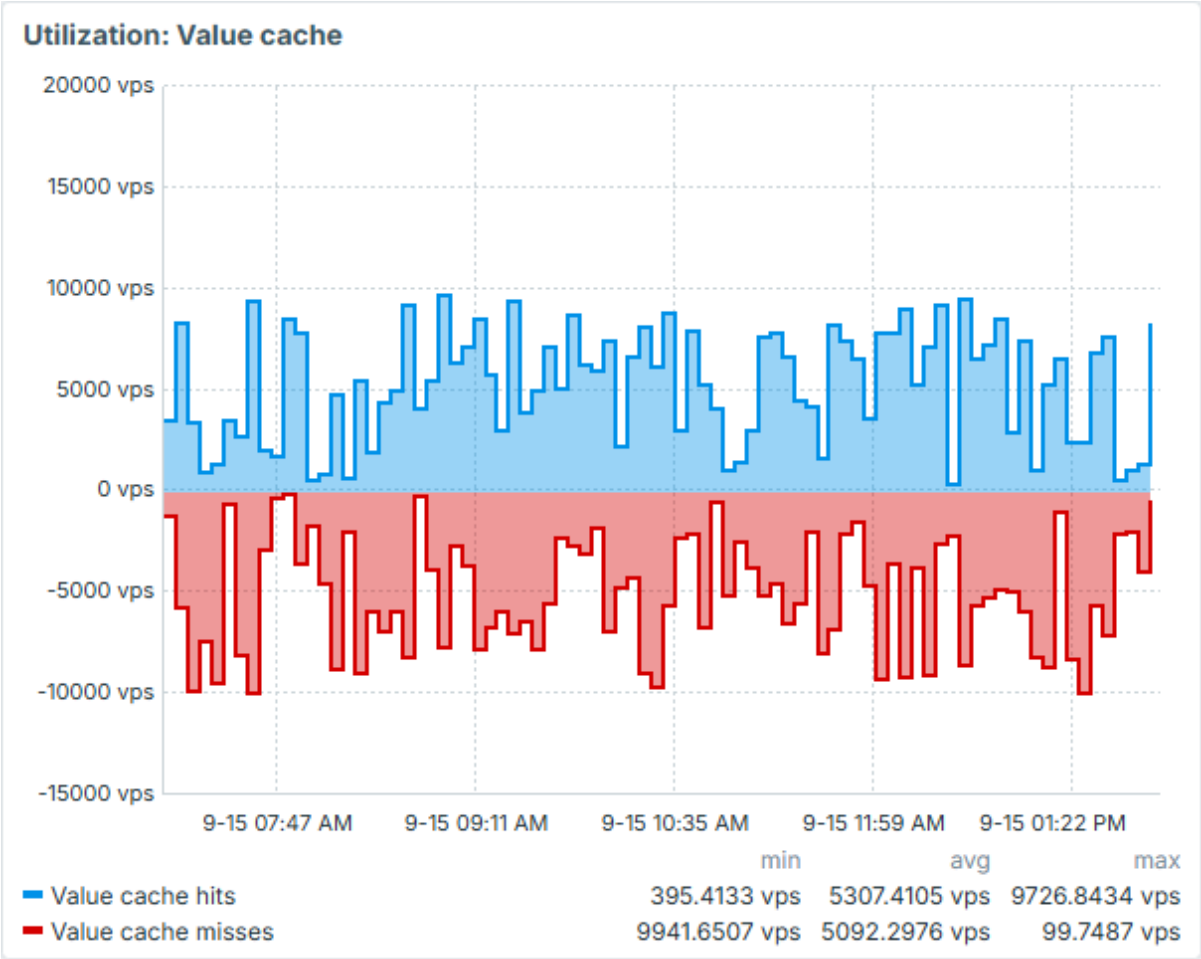


Lorsque vous survolez le graphique, une infobulle affiche la liste des éléments triés par valeur décroissante.

Lorsque vous survolez un élément dans l'infobulle, son graphique est mis en évidence (les autres sont estompés) ; la sélection de l'élément diffuse ses données vers les autres widgets.



Le graphique peut être affiché avec des valeurs inversées sur l'axe Y ; cela affecte uniquement la représentation visuelle et ne modifie pas les valeurs d'origine.

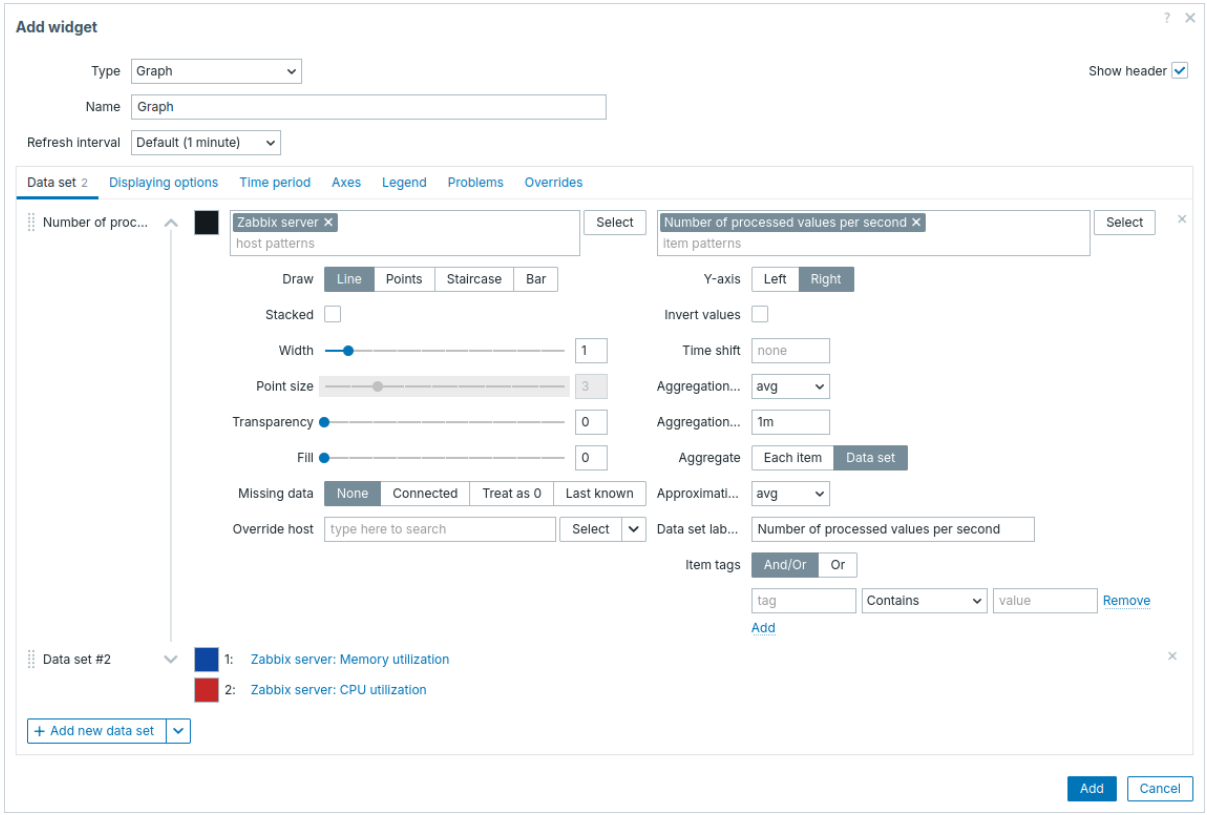


L'ancien widget Graphique (classique) est également disponible.

Les informations affichées dans le widget *Graphique* peuvent être téléchargées sous forme d'image PNG en sélectionnant l'option *Télécharger l'image* dans le menu *Actions* du widget ⋯, qui apparaît lorsque vous survolez le widget. Selon les paramètres de votre navigateur, l'image est soit enregistrée automatiquement dans le dossier `Downloads`, soit une boîte de dialogue *Enregistrer* s'ouvre et vous permet de choisir l'emplacement.

Configuration

Pour configurer, sélectionnez *Graph* comme type :



Ensemble de données

L'onglet **Ensemble de données** permet de sélectionner les données du graphique en ajoutant des ensembles de données. Deux types d'ensembles de données peuvent être ajoutés :

- *Motifs d'éléments* - les données des éléments correspondants sont affichées. Vous pouvez choisir une couleur de base unique ou sélectionner une ligne de palette pour attribuer des couleurs distinctes à chaque élément correspondant.
- *Liste d'éléments* - les données des éléments sélectionnés sont affichées. Vous pouvez choisir individuellement la couleur de chaque élément à l'aide du sélecteur.

Par défaut, un ensemble de données *Motifs d'éléments* est ajouté.

Ensemble de données	Pour l'ensemble de données **Motifs d'éléments** : Sélectionnez ou saisissez des motifs d'hôtes et d'éléments ; les données des éléments correspondant à ces motifs seront affichées sur le graphique ; jusqu'à 50 éléments peuvent être affichés. Des motifs génériques peuvent être utilisés pour la sélection (par exemple, ∗ renverra les résultats correspondant à zéro caractère ou plus). Pour spécifier un motif générique, saisissez manuellement la chaîne et appuyez sur *Entrée*. Le symbole générique est toujours interprété. Il n'est donc pas possible d'ajouter individuellement, par exemple, un élément nommé *item** s'il existe d'autres éléments correspondants (par exemple, *item2*, *item3*). La spécification de motifs d'hôtes et d'éléments est obligatoire pour les ensembles de données « Motifs d'éléments ». Voir également : Détails de la configuration de l'ensemble de données. Pour l'ensemble de données **Liste d'éléments** : Sélectionnez les éléments à afficher sur le graphique en cliquant sur le bouton *Ajouter un élément*. Vous pouvez également sélectionner des widgets compatibles comme source de données pour les éléments en cliquant sur le bouton *Ajouter un widget*. La spécification d'éléments ou de widgets est obligatoire pour les ensembles de données « Liste d'éléments ». Voir également : Détails de la configuration de l'ensemble de données. Notez que seuls les types d'éléments numériques sont autorisés. Lors de la configuration du widget sur un tableau de bord de modèle, le paramètre permettant de spécifier des motifs d'hôtes n'est pas disponible et le paramètre permettant de spécifier une liste d'éléments autorise uniquement la sélection des éléments configurés sur le modèle.
Tracé	Choisissez le type de tracé de la métrique. Types de tracé possibles : *Ligne* (défini par défaut), *Points*, *Escalier* et *Barres*. Notez que s'il n'y a qu'un seul point de données dans le graphique en ligne ou en escalier, il est tracé comme un point, quel que soit le type de tracé. La taille du point est calculée à partir de la largeur de la ligne, mais elle ne peut pas être inférieure à 3 pixels, même si la largeur de la ligne est moindre.
Empilé	Cochez la case pour afficher les données sous forme empilée (les zones remplies sont affichées). Cette option est désactivée lorsque le type de tracé *Points* est sélectionné.
Largeur	Définissez la largeur de la ligne. Cette option est disponible lorsque le type de tracé *Ligne* ou *Escalier* est sélectionné.
Taille des points	Définissez la taille des points. Cette option est disponible lorsque le type de tracé *Points* est sélectionné.
Transparence	Définissez le niveau de transparence.
Remplissage	Définissez le niveau de remplissage. Cette option est disponible lorsque le type de tracé *Ligne* ou *Escalier* est sélectionné.
Données manquantes	Sélectionnez l'option d'affichage des données manquantes : **Aucune** - l'espace est laissé vide ; **Connectées** - les deux valeurs limites sont reliées ; **Traiter comme 0** - les données manquantes sont affichées comme des valeurs égales à 0 ; **Dernière valeur connue** - les données manquantes sont affichées avec la même valeur que la dernière valeur connue ; ne s'applique pas aux types de tracé *Points* et *Barres*.
Remplacer l'hôte	Sélectionnez un widget compatible ou le sélecteur d'hôtes du tableau de bord comme source de données pour les hôtes. Ce paramètre n'est pas disponible lors de la configuration du widget sur un tableau de bord de modèle.
Axe Y	Sélectionnez le côté du graphique où l'axe Y sera affiché.

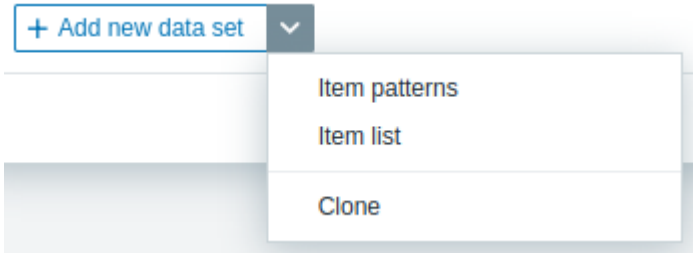
Inverser les valeurs	Cochez la case pour multiplier toutes les valeurs de l'axe Y par -1 avant de les afficher sur le graphique. Cela affecte uniquement la représentation visuelle ; les données d'origine restent inchangées. Cette option n'affecte pas l'infobulle, la légende ni la ligne de percentile. Si l'option *Déclencheurs simples* est sélectionnée pour l'axe, les seuils des déclencheurs seront affichés inversés.
Décalage temporel	Spécifiez un décalage temporel si nécessaire. Vous pouvez utiliser des suffixes temporels dans ce champ. Les valeurs négatives sont autorisées.
Fonction d'agrégation	Spécifiez la fonction d'agrégation à utiliser pour chaque élément ou pour un ensemble de données entier dans l'*intervalle d'agrégation* sélectionné : **min** - afficher la plus petite valeur ; **max** - afficher la plus grande valeur ; **avg** - afficher la valeur moyenne ; **sum** - afficher la somme des valeurs ; **count** - afficher le nombre de valeurs ; **first** - afficher la première valeur ; **last** - afficher la dernière valeur. Si **non utilisée** est sélectionné, toutes les valeurs sont affichées (sans agrégation). Seules les données numériques peuvent être affichées pour *min*, *max*, *avg* et *sum*. Pour *count*, les données non numériques sont converties en données numériques. Voir également : Agrégation dans les graphiques.
Intervalle d'agrégation	Spécifiez l'intervalle d'agrégation des valeurs. Vous pouvez utiliser des suffixes temporels dans ce champ. Une valeur numérique sans suffixe est considérée comme exprimée en secondes. Notez que si le widget est configuré pour afficher des données historiques basées sur les tendances (*Sélection des données historiques* est défini sur *Tendances* ou *Automatique*), il est recommandé d'utiliser un intervalle d'agrégation multiple d'une heure (par exemple, 3600, 60m, 1h, 3h, etc.). Les tendances stockent des valeurs agrégées par heure. L'utilisation d'un intervalle d'agrégation qui n'est pas un multiple d'une heure (par exemple, 100s, 7min, 15min, 90min, etc.) peut donc produire des résultats difficiles à interpréter.
Agrégation	Spécifiez s'il faut agréger : **Chaque élément** - chaque élément de l'ensemble de données sera agrégé et affiché séparément ; **Ensemble de données** - tous les éléments de l'ensemble de données seront agrégés et affichés sous la forme d'une seule valeur.
Approximation	Spécifiez la valeur à afficher lorsque plusieurs valeurs existent par pixel vertical du graphique : **all** - afficher les valeurs minimale, maximale et moyenne ; **min** - afficher la plus petite valeur ; **max** - afficher la plus grande valeur ; **avg** - afficher la valeur moyenne. Ce paramètre est utile lors de l'affichage d'un graphique couvrant une longue période avec un intervalle de mise à jour fréquent (par exemple, une année de valeurs collectées toutes les 10 minutes).
Libellé de l'ensemble de données	Spécifiez le libellé de l'ensemble de données affiché dans la configuration de l'*Ensemble de données* du graphique et dans la *Légende* du graphique (pour les ensembles de données agrégés). Tous les ensembles de données sont numérotés, y compris ceux auxquels un *Libellé de l'ensemble de données* a été attribué. Si aucun libellé n'est spécifié, l'ensemble de données reçoit automatiquement un libellé correspondant à son numéro (par exemple, « Ensemble de données n° 2 », « Ensemble de données n° 3 », etc.). La numérotation des ensembles de données est recalculée après leur réorganisation ou leur déplacement par glisser-déposer. Les libellés trop longs sont raccourcis pour s'adapter à leur emplacement d'affichage (par exemple, « Nombre de proc... »).

Étiquettes des éléments	Spécifiez les étiquettes permettant de filtrer les éléments affichés dans le widget. Il est possible d'inclure ou d'exclure des étiquettes et des valeurs d'étiquettes spécifiques. Plusieurs conditions peuvent être définies. La correspondance des noms d'étiquettes est toujours sensible à la casse. Plusieurs opérateurs sont disponibles pour chaque condition : **Existe** - inclure les noms d'étiquettes spécifiés ; **Est égal à** - inclure les noms et valeurs d'étiquettes spécifiés (sensible à la casse) ; **Contient** - inclure les noms d'étiquettes spécifiés lorsque les valeurs d'étiquettes contiennent la chaîne saisie (correspondance de sous-chaîne, insensible à la casse) ; **N'existe pas** - exclure les noms d'étiquettes spécifiés ; **N'est pas égal à** - exclure les noms et valeurs d'étiquettes spécifiés (sensible à la casse) ; **Ne contient pas** - exclure les noms d'étiquettes spécifiés lorsque les valeurs d'étiquettes contiennent la chaîne saisie (correspondance de sous-chaîne, insensible à la casse). Deux types de calcul sont disponibles pour les conditions : **Et/Ou** - toutes les conditions doivent être remplies ; les conditions ayant le même nom d'étiquette sont regroupées avec la condition Ou ; **Ou** - une seule condition remplie suffit.

Détails de configuration des ensembles de données

Les ensembles de données existants sont affichés dans une liste. Vous pouvez :

- Cliquer sur l'icône de déplacement et faire glisser un ensemble de données vers un nouvel emplacement dans la liste.
- Cliquer sur l'icône de développement pour développer les détails de l'ensemble de données. Une fois développé, cet icône devient une icône de réduction .
- Cliquer sur l'icône de couleur pour ouvrir le sélecteur. Vous pouvez saisir un code hexadécimal, choisir un échantillon de couleur uni ou passer à l'onglet *Palette* et sélectionner une ligne de couleurs prédéfinies. La couleur choisie s'applique directement aux ensembles de données *Item list* ou sert de base aux nuances générées dans *Item patterns*. Utilisez la touche Tab pour passer d'un contrôle à l'autre dans la boîte de dialogue, les touches fléchées pour parcourir les échantillons ou les lignes de la palette, la touche Entrée pour sélectionner et la touche Échap pour annuler.
- Cliquer sur le bouton *Add new data set* pour ajouter un ensemble de données vide permettant de sélectionner des hôtes et des modèles d'éléments. Si vous cliquez sur l'icône pointant vers le bas à côté du bouton *Add new data set*, un menu déroulant apparaît. Il vous permet d'ajouter un nouvel ensemble de données *Item patterns* ou *Item list*, ou de *Clone* l'ensemble de données actuellement ouvert. Si tous les ensembles de données sont réduits, l'option *Clone* n'est pas disponible.

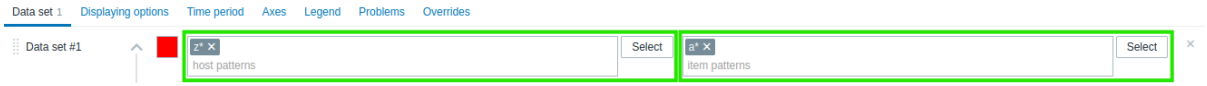


L'ensemble de données **Modèles d'éléments** contient les champs *Modèles d'hôtes* et *Modèles d'éléments*, qui reconnaissent tous deux les noms complets ou les modèles contenant un symbole générique (*). Cette fonctionnalité vous permet de sélectionner tous les noms d'hôtes et noms d'éléments contenant le modèle sélectionné. Lors de la saisie du nom de l'élément ou du modèle d'élément dans le champ *Modèles d'éléments*, seuls les éléments appartenant au(x) nom(s) d'hôte sélectionné(s) sont affichés dans la liste déroulante.

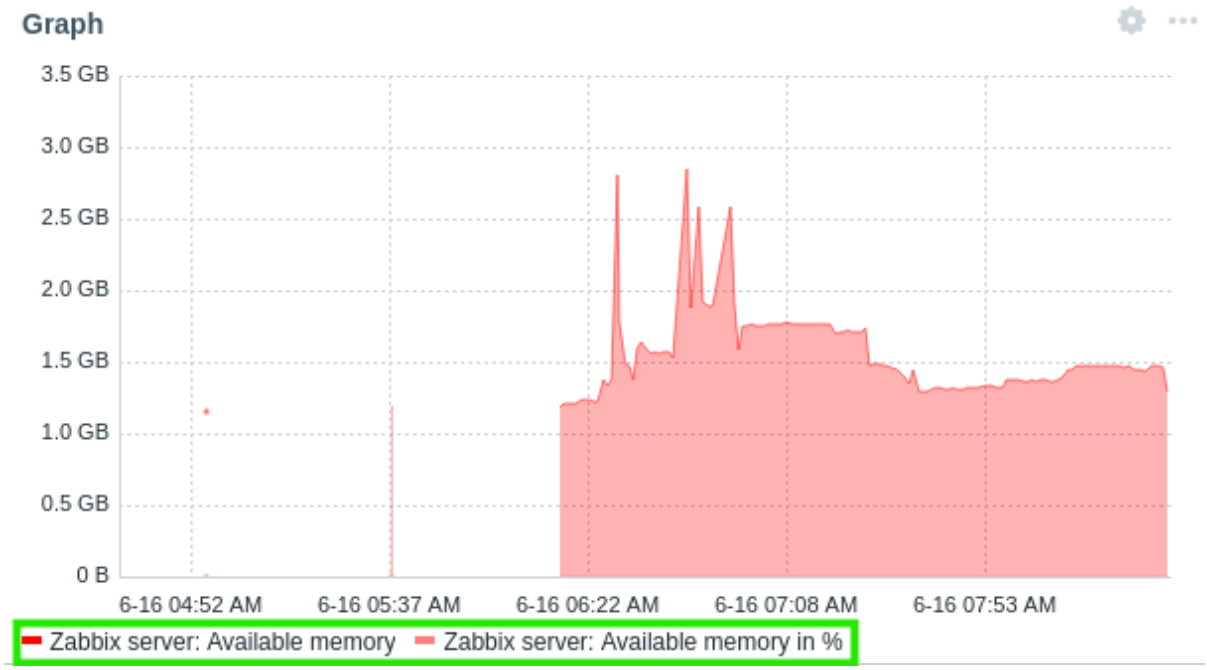
Par exemple, après avoir saisi le modèle **z*** dans le champ *Modèles d'hôtes*, la liste déroulante affiche tous les noms d'hôtes contenant ce modèle : **z*, Zabbix server, Zabbix proxy**. Après avoir appuyé sur *Entrée*, ce modèle est accepté et affiché comme **z***. De même, après avoir saisi le modèle **a*** dans le champ *Modèles d'éléments*, la liste déroulante affiche tous les noms d'éléments contenant ce modèle : **a*, Available memory, Available memory in %**.



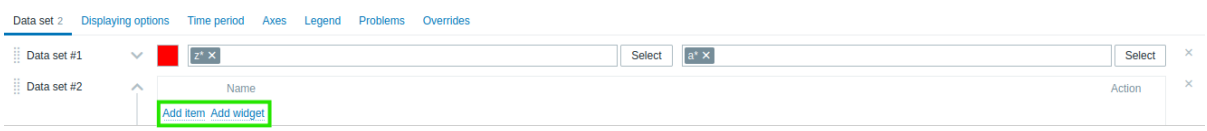
Après avoir appuyé sur *Entrée*, le modèle est accepté et affiché comme **a***.



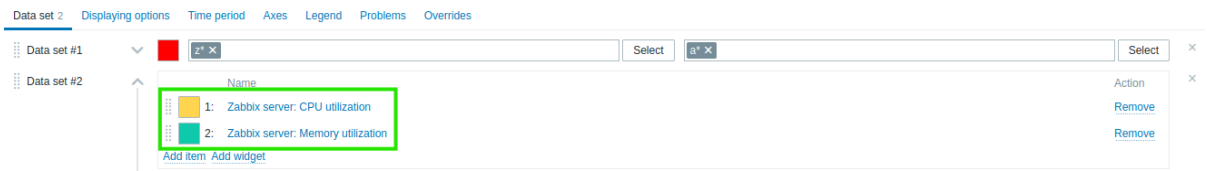
Le graphique affiche alors tous les éléments appartenant au(x) nom(s) d'hôte sélectionné(s).



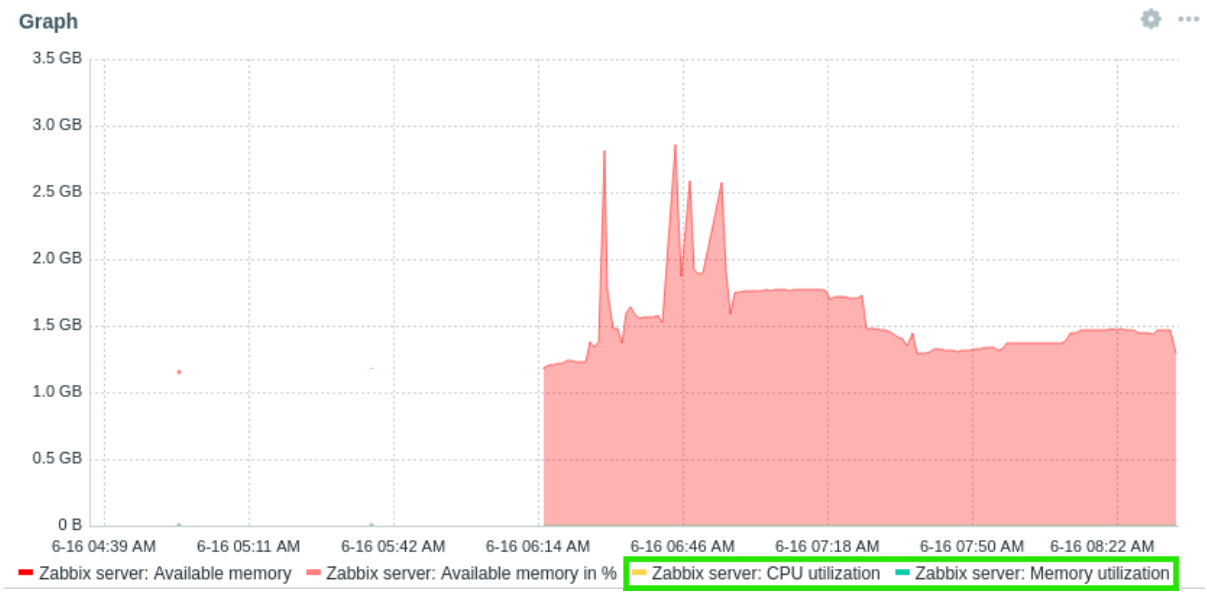
L'ensemble de données **Liste des éléments** contient le bouton *Ajouter un élément*, qui vous permet d'ajouter des éléments à afficher sur le graphique. Vous pouvez également ajouter des widgets compatibles comme source de données pour les éléments en cliquant sur le bouton *Ajouter un widget*.



Par exemple, lorsque vous cliquez sur le bouton *Ajouter un élément*, une fenêtre contextuelle s'ouvre et contient un paramètre *Hôte*. Après avoir sélectionné un hôte, tous ses éléments disponibles à la sélection sont affichés dans une liste.



Après avoir sélectionné un ou plusieurs éléments, ils seront affichés dans la liste des éléments de l'ensemble de données ainsi que dans le graphique.

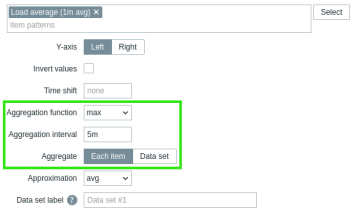


Agrégation dans les graphiques

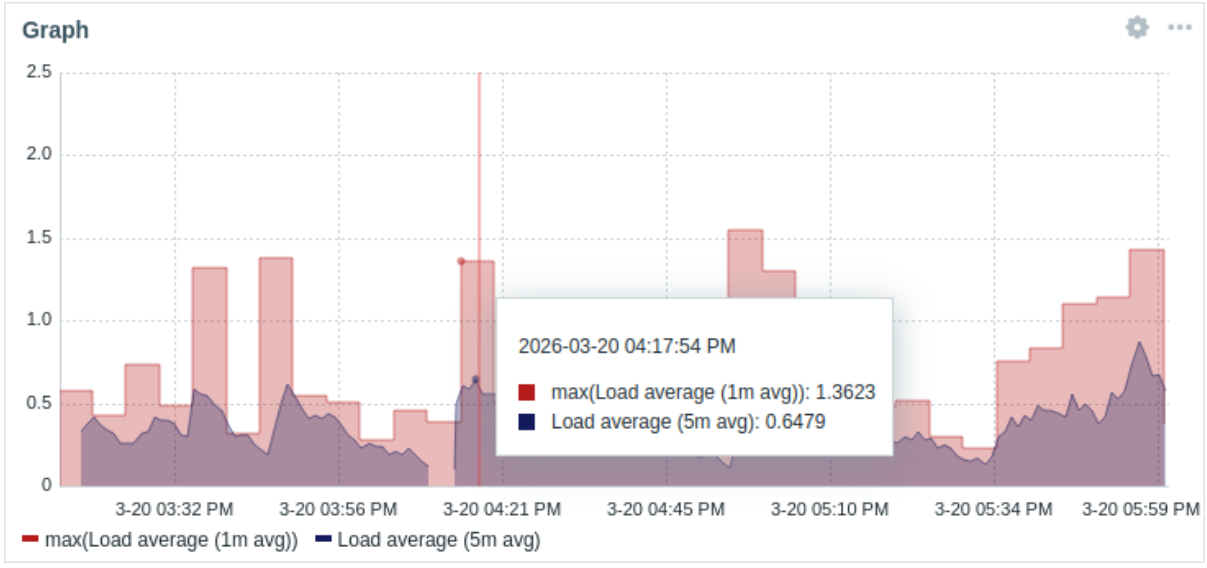
L'agrégation permet d'afficher une valeur agrégée (minimum, maximum, moyenne, etc.) pour l'intervalle spécifié (5 minutes, une heure, un jour, etc.) au lieu de toutes les valeurs individuelles. Dans un ensemble de données, l'agrégation peut être appliquée soit à chaque élément séparément, soit à l'ensemble des données.

Voici quelques cas d'utilisation de la fonction d'agrégation : - Le nombre moyen de requêtes adressées à un serveur NGINX par jour. - L'espace disque disponible le plus faible parmi les clusters par semaine.

Dans l'exemple suivant, la fonction d'agrégation « max » est appliquée à l'élément avec un intervalle d'agrégation de cinq minutes :



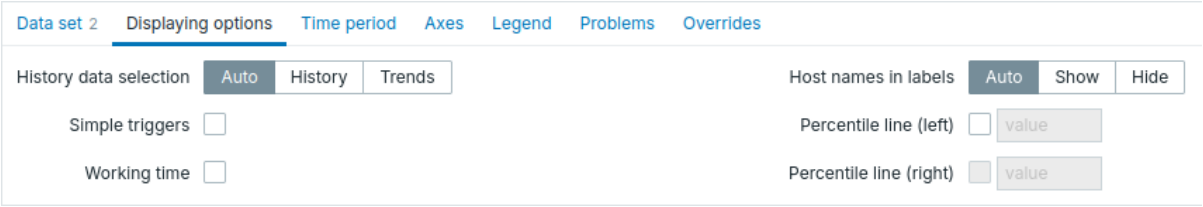
L'exemple montre la charge CPU maximale sur cinq minutes (agrégée à partir de valeurs moyennes calculées sur une minute), comparée à la charge CPU moyenne sur cinq minutes.



Lorsque *Afficher la fonction d'agrégation* est sélectionné pour la légende, les éléments sont affichés entre parenthèses, précédés de la fonction d'agrégation dans la légende et dans l'infobulle. Si *Libellé de l'ensemble de données* est spécifié pour un graphique, le libellé est affiché entre parenthèses, précédé de la fonction d'agrégation.

Options d'affichage

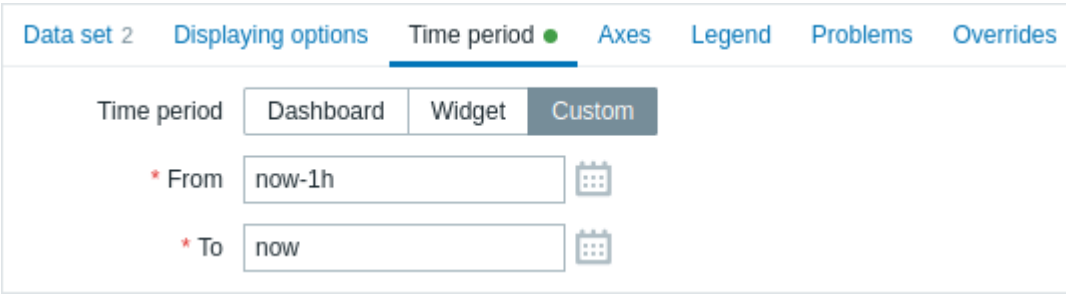
L'onglet **Options d'affichage** vous permet de définir la sélection des données d'historique :



Sélection des données d'historique	Définissez la source des données du graphique : **Auto** - les données sont sélectionnées conformément à l'algorithme classique des graphiques (par défaut) ; **Historique** - données provenant de l'historique ; **Tendances** - données provenant des tendances.
Déclencheurs simples	Cochez la case pour afficher les seuils des déclencheurs simples. Les seuils sont représentés par des lignes en pointillés utilisant la couleur de gravité du déclencheur. Un déclencheur simple est un déclencheur comportant une seule fonction (uniquement `last`, `max`, `min`, `avg`) pour un élément dans l'expression. Un maximum de trois déclencheurs peuvent être représentés. Notez que le déclencheur doit se trouver dans la plage affichée pour être visible.
Temps de fonctionnement	Cochez la case pour afficher le temps de fonctionnement sur le graphique. Le temps de fonctionnement (jours ouvrés) est affiché sur les graphiques avec un arrière-plan blanc, tandis que le temps hors fonctionnement est affiché en gris (avec le thème d'interface *Original blue* par défaut).
Noms des hôtes dans les libellés	Choisissez d'afficher ou non les noms des hôtes dans la légende du graphique et dans l'infobulle qui apparaît lorsque vous survolez les données du graphique : **Auto** - les noms des hôtes sont affichés uniquement si plusieurs hôtes sont présents dans l'ensemble des jeux de données (par défaut). **Afficher** - les noms des hôtes sont affichés. **Masquer** - les noms des hôtes sont masqués. Ce paramètre n'est pas disponible lors de la configuration du widget sur un tableau de bord de modèle.
Ligne de percentile (gauche)	Cochez la case et saisissez la valeur du percentile pour afficher le percentile spécifié sous forme de ligne sur l'axe Y gauche du graphique. Par exemple, si un percentile de 95 % est défini, la ligne du percentile se trouvera au niveau sous lequel se situent 95 % des valeurs.
Ligne de percentile (droite)	Cochez la case et saisissez la valeur du percentile pour afficher le percentile spécifié sous forme de ligne sur l'axe Y droit du graphique. Par exemple, si un percentile de 95 % est défini, la ligne du percentile se trouvera au niveau sous lequel se situent 95 % des valeurs.

Période

L'onglet **Période** vous permet de définir une période pour laquelle afficher les données dans le graphique :

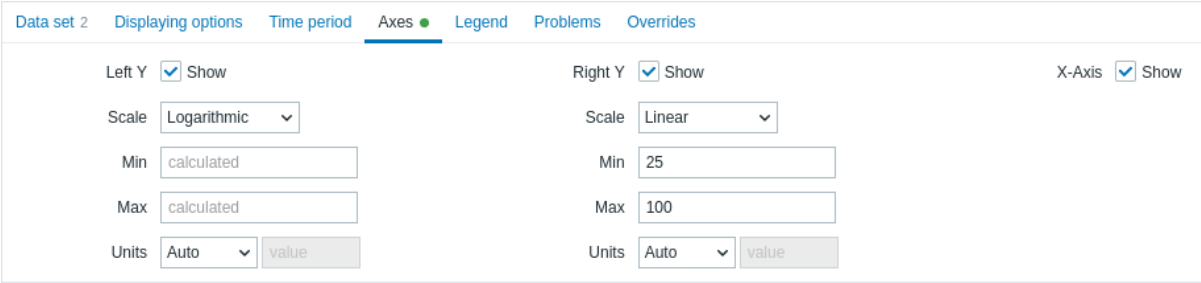


Période	Sélectionnez la source de données pour la période : **Tableau de bord** - utilisez le sélecteur de période du tableau de bord ; **Widget** - utilisez un widget compatible (défini dans le paramètre *Widget*) ; **Personnalisée** - utilisez une période personnalisée définie dans les paramètres *Du* et *Au* ; si elle est définie, une icône d'horloge s'affiche dans le coin supérieur droit du widget et indique l'heure définie au survol de la souris. Notez que, quelle que soit la configuration *Période* du widget, les widgets compatibles peuvent toujours l'utiliser comme source de données pour la période.

Widget	Saisissez ou sélectionnez un widget compatible comme source de données pour la période. Ce paramètre est disponible si *Période* est défini sur « Widget ».
Du	Saisissez ou sélectionnez le début de la période. La syntaxe des périodes relatives (`now`, `now/d`, `now/w-1w`, etc.) est prise en charge. Ce paramètre est disponible si *Période* est défini sur « Personnalisée ».
Au	Saisissez ou sélectionnez la fin de la période. La syntaxe des périodes relatives (`now`, `now/d`, `now/w-1w`, etc.) est prise en charge. Ce paramètre est disponible si *Période* est défini sur « Personnalisée ».

Axes

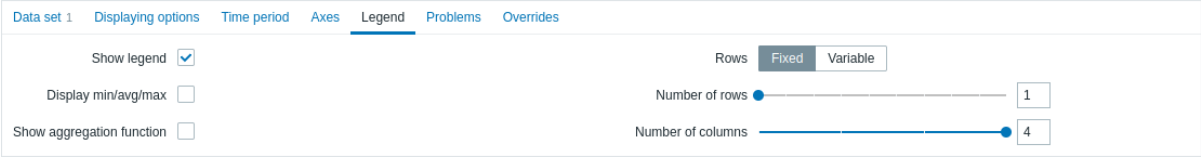
L'onglet **Axes** vous permet de personnaliser l'affichage des axes :



Y gauche	Cochez cette case pour rendre l'axe Y gauche visible. La case peut être désactivée si elle n'est pas sélectionnée dans l'onglet *Ensemble de données* ou *Remplacements*.
Y droite	Cochez cette case pour rendre l'axe Y droit visible. La case peut être désactivée si elle n'est pas sélectionnée dans l'onglet *Ensemble de données* ou *Remplacements*.
Axe X	Décochez cette case pour masquer l'axe X (cochée par défaut).
Échelle	Choisissez l'échelle des valeurs de l'axe du graphique dans la liste déroulante : **Linéaire** - les valeurs de l'axe augmentent d'une quantité fixe (par exemple, 10, 20, 30), ce qui convient aux données qui évoluent régulièrement ou couvrent une plage petite à modérée ; **Logarithmique** - les valeurs de l'axe augmentent de manière exponentielle (par exemple, 10, 100, 1000), ce qui convient aux données qui évoluent rapidement ou couvrent une grande plage.
Min	Définissez la valeur minimale de l'axe correspondant. La valeur minimale de la plage visible de l'axe Y est spécifiée.
Max	Définissez la valeur maximale de l'axe correspondant. La valeur maximale de la plage visible de l'axe Y est spécifiée.
Unités	Choisissez l'unité des valeurs de l'axe du graphique dans la liste déroulante : **Automatique** - les valeurs de l'axe sont affichées avec l'unité du premier élément de l'ensemble de données ; **Statique** - les valeurs de l'axe sont affichées avec l'unité spécifiée dans le champ de saisie *value* ; si le champ est laissé vide, seules les valeurs numériques sont affichées.

Légende

L'onglet **Légende** vous permet de personnaliser la légende du graphique :

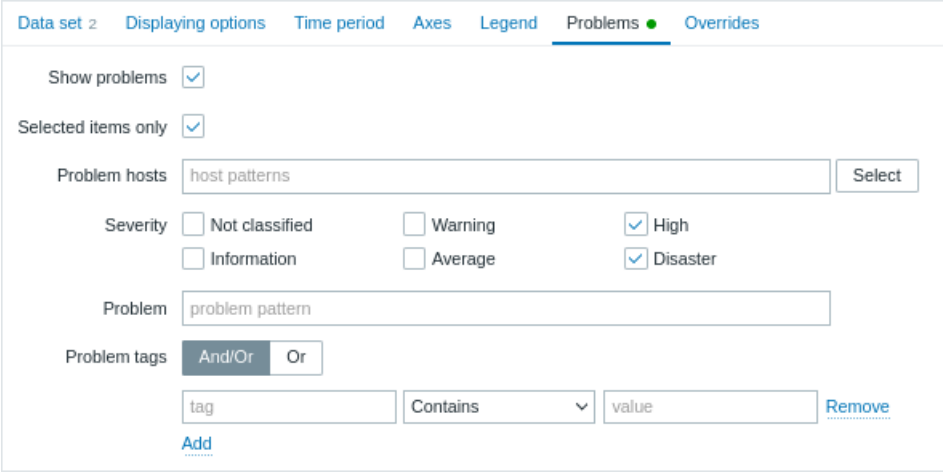


Afficher la légende	Décochez cette case pour masquer la légende sur le graphique (cochée par défaut).
Afficher le min./moy./max.	Cochez cette case pour afficher les valeurs minimale, moyenne et maximale de l'élément dans la légende.
Afficher la fonction d'agrégation	Cochez cette case pour afficher la fonction d'agrégation dans la légende.

Lignes	Sélectionnez le mode d'affichage des lignes de la légende : **Fixe** - le nombre de lignes affichées est déterminé par la valeur du paramètre *Nombre de lignes* ; **Variable** - le nombre de lignes affichées est déterminé par le nombre d'éléments configurés, sans dépasser la valeur du paramètre *Nombre maximal de lignes*.
Nombre de lignes/ Nombre maximal de lignes	Si *Lignes* est défini sur « Fixe », définissez le nombre de lignes de légende à afficher (1-10). Si *Lignes* est défini sur « Variable », définissez le nombre maximal de lignes de légende à afficher (1-10).
Nombre de colonnes	Définissez le nombre de colonnes de légende à afficher (1-4). Ce paramètre est disponible si *Afficher le min./moy./max.* n'est pas coché.

Problèmes

L'onglet **Problèmes** vous permet de personnaliser l'affichage des problèmes :

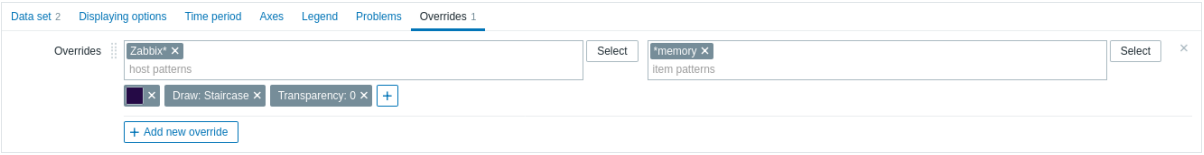


Afficher les problèmes	Cochez cette case pour activer l'affichage des problèmes sur le graphique (décochée, c'est-à-dire désactivée par défaut).
Éléments sélectionnés uniquement	Cochez cette case pour afficher sur le graphique uniquement les problèmes associés aux éléments sélectionnés.
Hôtes concernés par les problèmes	Sélectionnez les hôtes concernés par les problèmes à afficher sur le graphique. Des modèles génériques peuvent être utilisés (par exemple, * renverra les résultats correspondant à zéro caractère ou plus). Pour spécifier un modèle générique, saisissez simplement la chaîne manuellement et appuyez sur *Entrée*. Pendant la saisie, observez que tous les hôtes correspondants s'affichent dans la liste déroulante. Ce paramètre n'est pas disponible lors de la configuration du widget sur un tableau de bord de modèle.
Gravité	Cochez les niveaux de gravité des problèmes afin de filtrer les problèmes à afficher sur le graphique. Si aucun niveau de gravité n'est coché, tous les problèmes seront affichés.
Problème	Indiquez le nom du problème à afficher sur le graphique.

Étiquettes des problèmes	Indiquez les étiquettes des problèmes afin de limiter le nombre de problèmes affichés dans le widget. Il est possible d'inclure ou d'exclure des étiquettes et des valeurs d'étiquette spécifiques. Plusieurs conditions peuvent être définies. La correspondance des noms d'étiquette est toujours sensible à la casse. Plusieurs opérateurs sont disponibles pour chaque condition : **Existe** - inclut les noms d'étiquette spécifiés ; **Est égal à** - inclut les noms et les valeurs d'étiquette spécifiés (sensible à la casse) ; **Contient** - inclut les noms d'étiquette spécifiés lorsque les valeurs d'étiquette contiennent la chaîne saisie (correspondance de sous-chaîne, insensible à la casse) ; **N'existe pas** - exclut les noms d'étiquette spécifiés ; **N'est pas égal à** - exclut les noms et les valeurs d'étiquette spécifiés (sensible à la casse) ; **Ne contient pas** - exclut les noms d'étiquette spécifiés lorsque les valeurs d'étiquette contiennent la chaîne saisie (correspondance de sous-chaîne, insensible à la casse). Il existe deux types de calcul pour les conditions : **Et/Ou** - toutes les conditions doivent être remplies ; les conditions ayant le même nom d'étiquette sont regroupées avec la condition *Ou* ; **Ou** - une seule condition remplie suffit.

Remplacements

L'onglet **Remplacements** vous permet d'ajouter des remplacements personnalisés pour les ensembles de données :



Les remplacements sont utiles lorsque plusieurs éléments sont sélectionnés pour un ensemble de données à l'aide du caractère générique * et que vous souhaitez modifier la manière dont les éléments sont affichés par défaut (par exemple, la couleur de base par défaut ou toute autre propriété).

Les remplacements existants, le cas échéant, sont affichés dans une liste. Pour ajouter un nouveau remplacement :

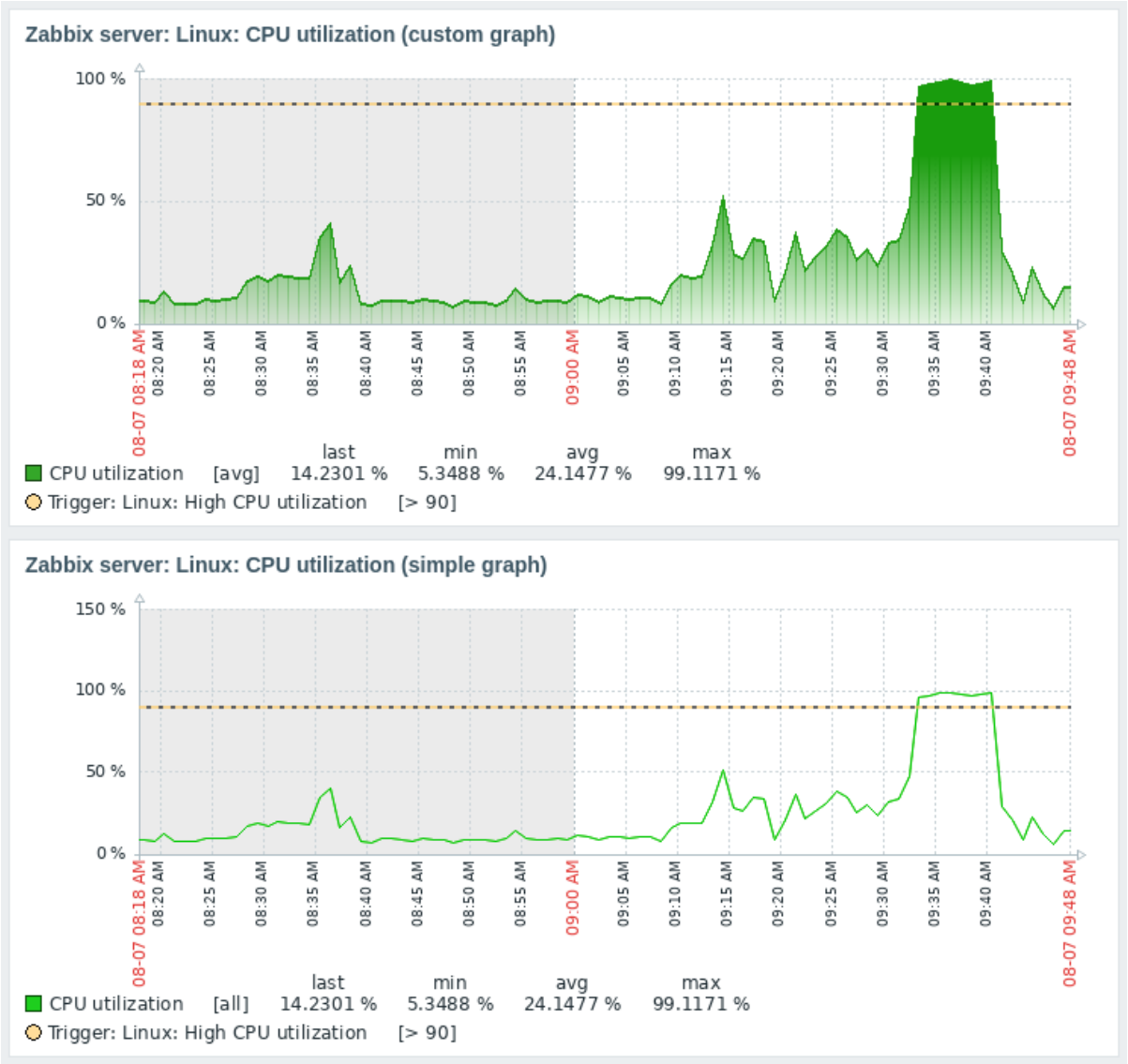
- Cliquez sur le bouton 

- Sélectionnez les hôtes et les éléments concernés par le remplacement. Vous pouvez également saisir des modèles d'hôtes et d'éléments. Les modèles avec caractères génériques peuvent être utilisés (par exemple, * renverra les résultats correspondant à zéro caractère ou plus). Pour spécifier un modèle avec caractère générique, saisissez simplement la chaîne manuellement et appuyez sur *Entrée*. Pendant la saisie, notez que tous les hôtes correspondants sont affichés dans la liste déroulante. Le symbole générique est toujours interprété. Il n'est donc pas possible d'ajouter individuellement, par exemple, un élément nommé "item*" s'il existe d'autres éléments correspondants (par exemple, item2, item3). Les paramètres de modèle d'hôte et de modèle d'élément sont obligatoires. Le paramètre permettant de spécifier les modèles d'hôtes n'est pas disponible lors de la configuration du widget sur un tableau de bord de modèle. Le paramètre permettant de spécifier une liste d'éléments autorise uniquement la sélection des éléments configurés sur le modèle lors de la configuration du widget sur un tableau de bord de modèle.
- Cliquez sur  pour sélectionner les paramètres de remplacement. Au moins un paramètre de remplacement doit être sélectionné. Pour obtenir la description des paramètres, consultez l'onglet *Ensemble de données* ci-dessus.

Graphique (classique)

Présentation

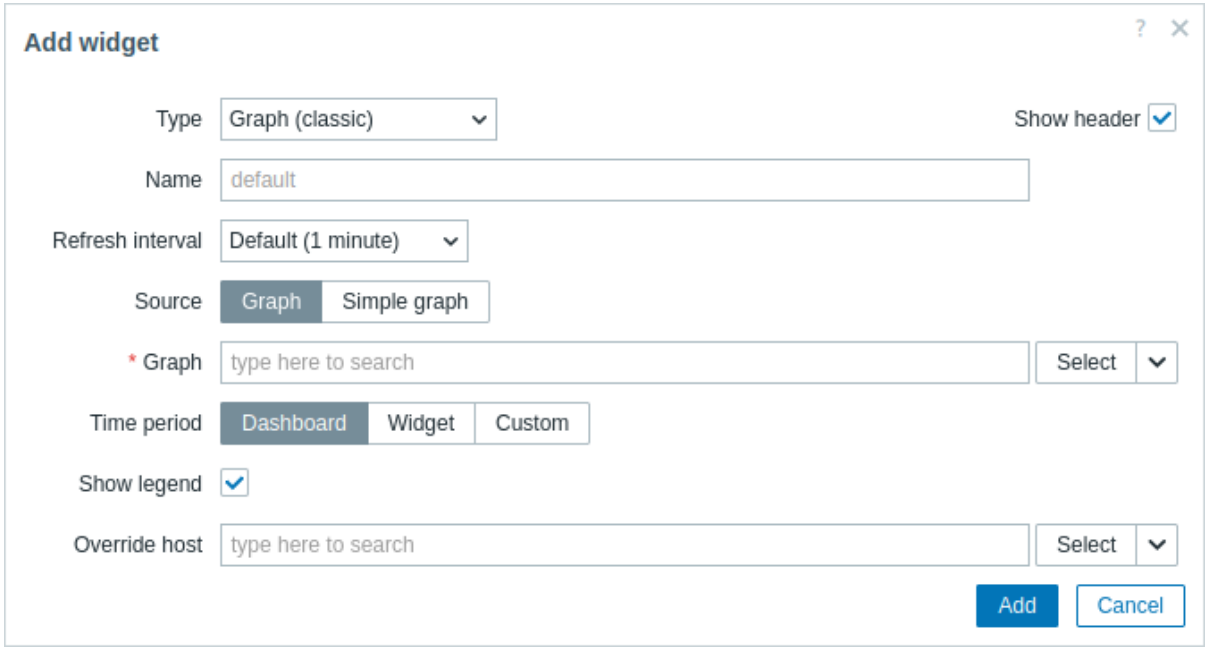
Le widget *Graphique (classique)* affiche les données numériques d'un élément sous forme d'image, à l'aide d'un graphique personnalisé ou d'un graphique simple. Il prend en charge la sélection d'une période, les lignes de déclencheur et l'affichage facultatif d'une légende.



Les informations affichées dans le widget *Graphique (classique)* peuvent être téléchargées sous forme d'image PNG en sélectionnant l'option *Télécharger l'image* dans le menu *Actions* du widget ⋯, qui apparaît lorsque vous survolez le widget. Selon les paramètres de votre navigateur, l'image est soit enregistrée automatiquement dans le dossier `Downloads`, soit une boîte de dialogue *Enregistrer* s'ouvre pour vous permettre de choisir l'emplacement.

Configuration

Pour configurer, sélectionnez *Graph (classic)* comme type :



En plus des paramètres communs à tous les widgets, vous pouvez définir les options spécifiques suivantes :

Source	Sélectionnez le type de graphique : **Graph** - graphique personnalisé ; **Simple graph** - graphique simple.
Graph	Sélectionnez le graphique personnalisé à afficher. Vous pouvez également sélectionner un widget compatible comme source de données pour les graphiques. Ce paramètre est disponible si *Source* est défini sur "Graph".
Item	Sélectionnez l'élément à afficher dans un graphique simple. Vous pouvez également sélectionner un widget compatible comme source de données pour les éléments. Ce paramètre est disponible si *Source* est défini sur "Simple graph".
Time period	Définissez une période pour laquelle afficher les données dans le graphique. Sélectionnez la source de données pour la période : **Dashboard** - utiliser le sélecteur de période du tableau de bord ; **Widget** - utiliser un widget compatible (défini dans le paramètre *Widget*) ; **Custom** - utiliser une période personnalisée définie dans les paramètres *From* et *To* ; si elle est définie, une icône d'horloge sera affichée dans le coin supérieur droit du widget, indiquant l'heure définie au survol de la souris. Notez que, quelle que soit la configuration de *Time period* du widget, les widgets compatibles peuvent toujours l'utiliser comme source de données pour la période.
Widget	Saisissez ou sélectionnez un widget compatible comme source de données pour la période. Ce paramètre est disponible si *Time period* est défini sur "Widget".
From	Saisissez ou sélectionnez le début de la période. La syntaxe de temps relatif (`now`, `now/d`, `now/w-1w`, etc.) est prise en charge. Ce paramètre est disponible si *Time period* est défini sur "Custom".
To	Saisissez ou sélectionnez la fin de la période. La syntaxe de temps relatif (`now`, `now/d`, `now/w-1w`, etc.) est prise en charge. Ce paramètre est disponible si *Time period* est défini sur "Custom".
Show legend	Décochez cette case pour masquer la légende sur le graphique (cochée par défaut).
Override host	Sélectionnez un widget compatible ou le sélecteur d'hôte du tableau de bord comme source de données pour les hôtes. Ce paramètre n'est pas disponible lors de la configuration du widget sur un tableau de bord de modèle.

La légende du graphique se compose de trois sections :

- Les éléments et leurs valeurs agrégées
- Les percentiles (si configurés)
- Les déclencheurs (s'ils sont associés aux éléments affichés)

Si la hauteur du graphique dans le widget est insuffisante, la légende peut ne pas être affichée ou n'être affichée que partiellement. Les déclencheurs et les percentiles sont masqués en premier, suivis de la légende des éléments. Pour afficher la légende complète, augmentez la taille verticale du widget.

Note:
Pas plus de 3 lignes de déclencheur peuvent être affichées. S'il y a davantage de déclencheurs, ceux ayant une gravité plus faible sont prioritaires pour l'affichage.

Prototype de graphique

Aperçu

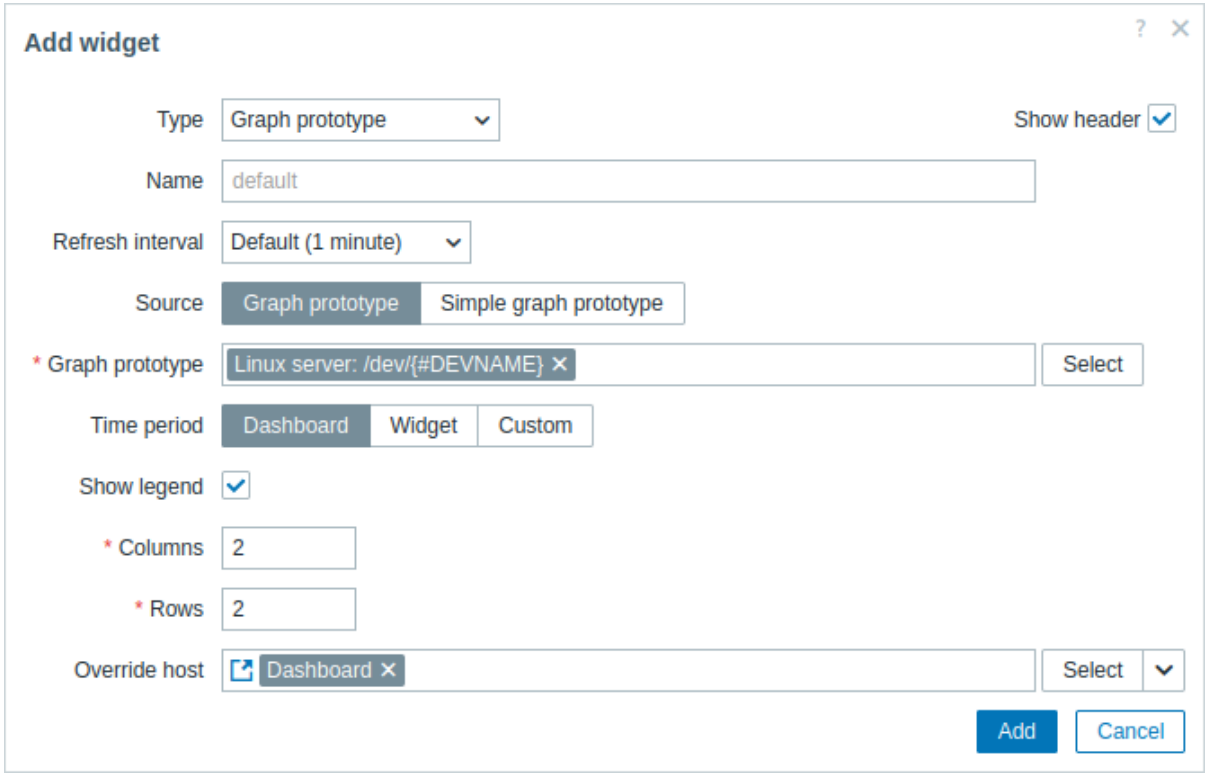
Le widget *Prototype de graphique* affiche une grille de graphiques personnalisés créés automatiquement, basée sur des prototypes de graphiques ou des prototypes d'éléments de règle de découverte de bas niveau.



Les graphiques sont disposés dans une grille configurable ; si le nombre de graphiques dépasse la disposition définie, la pagination est activée dans le widget.

Configuration

Pour configurer le widget, sélectionnez *Graph prototype* comme type de widget :



En plus des paramètres communs à tous les widgets, vous pouvez définir les options spécifiques suivantes :

Source	Sélectionnez la source des graphiques : **Graph prototype** ou **Simple graph prototype**.
Graph prototype	Sélectionnez un prototype de graphique pour afficher les graphiques découverts par le prototype de graphique. Ce paramètre est disponible si *Source* est défini sur « Graph prototype ».
Item prototype	Sélectionnez un prototype d'élément pour afficher les graphiques simples des éléments découverts par le prototype d'élément. Ce paramètre est disponible si *Source* est défini sur « Simple graph prototype ».
Time period	Définissez une période pour laquelle afficher les données dans les graphiques. Sélectionnez la source de données de la période : **Dashboard** - utilisez le sélecteur de période du tableau de bord ; **Widget** - utilisez un widget compatible (défini dans le paramètre *Widget*) ; **Custom** - utilisez une période personnalisée définie dans les paramètres *From* et *To* ; si elle est définie, une icône d'horloge s'affiche dans l'angle supérieur droit du widget et indique l'heure définie au survol de la souris. Notez que, quelle que soit la configuration *Time period* du widget, les widgets compatibles peuvent toujours l'utiliser comme source de données pour la période.
Widget	Saisissez ou sélectionnez un widget compatible comme source de données de la période. Ce paramètre est disponible si *Time period* est défini sur « Widget ».
From	Saisissez ou sélectionnez le début de la période. La syntaxe des périodes relatives (`now`, `now/d`, `now/w-1w`, etc.) est prise en charge. Ce paramètre est disponible si *Time period* est défini sur « Custom ».
To	Saisissez ou sélectionnez la fin de la période. La syntaxe des périodes relatives (`now`, `now/d`, `now/w-1w`, etc.) est prise en charge. Ce paramètre est disponible si *Time period* est défini sur « Custom ».
Show legend	Désélectionnez cette case pour masquer la légende des graphiques (sélectionnée par défaut).
Override host	Sélectionnez un widget compatible ou le sélecteur d'hôte du tableau de bord comme source de données des hôtes. Ce paramètre n'est pas disponible lors de la configuration du widget sur un tableau de bord de modèle.
Columns	Saisissez le nombre de colonnes de graphiques à afficher dans un widget de prototype de graphique.
Rows	Saisissez le nombre de lignes de graphiques à afficher dans un widget de prototype de graphique.

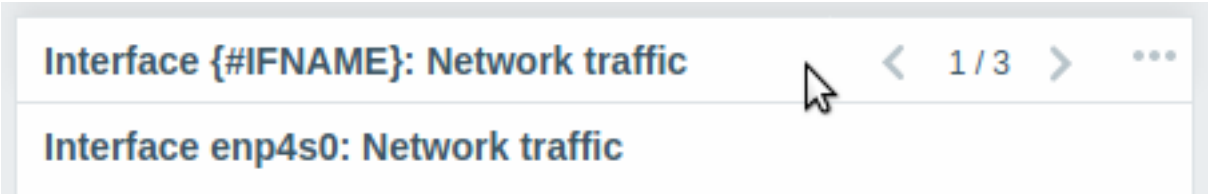
La légende du graphique se compose de trois sections :

- Éléments et leurs valeurs agrégées
- Percentiles (s'ils sont configurés)
- Déclencheurs (s'ils sont associés aux éléments affichés)

Si la hauteur du graphique dans le widget est insuffisante, la légende peut ne pas s'afficher ou ne s'afficher que partiellement. Les déclencheurs et les percentiles sont masqués en premier, suivis de la légende des éléments. Pour afficher la légende complète, augmentez la taille verticale du widget.

Note:
Pas plus de 3 lignes de déclencheurs peuvent être affichées. S'il y a davantage de déclencheurs, ceux dont la sévérité est la plus faible sont prioritaires pour l'affichage.

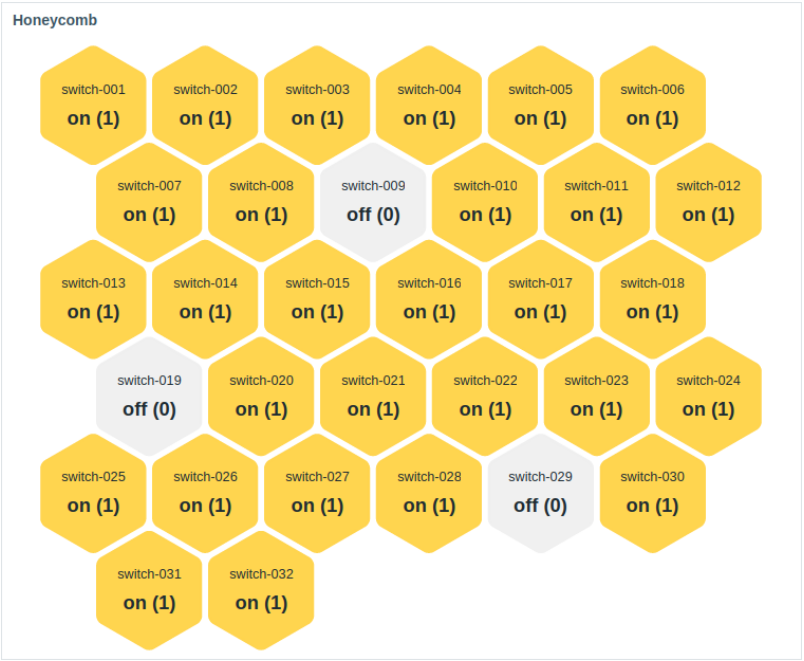
Bien que les paramètres *Columns* et *Rows* permettent d'afficher plusieurs graphiques dans le widget, il peut toujours y avoir plus de graphiques découverts que de colonnes ou de lignes dans le widget. Dans ce cas, la pagination devient disponible dans le widget et un en-tête coulissant permet de passer d'une page à l'autre à l'aide des flèches gauche et droite :

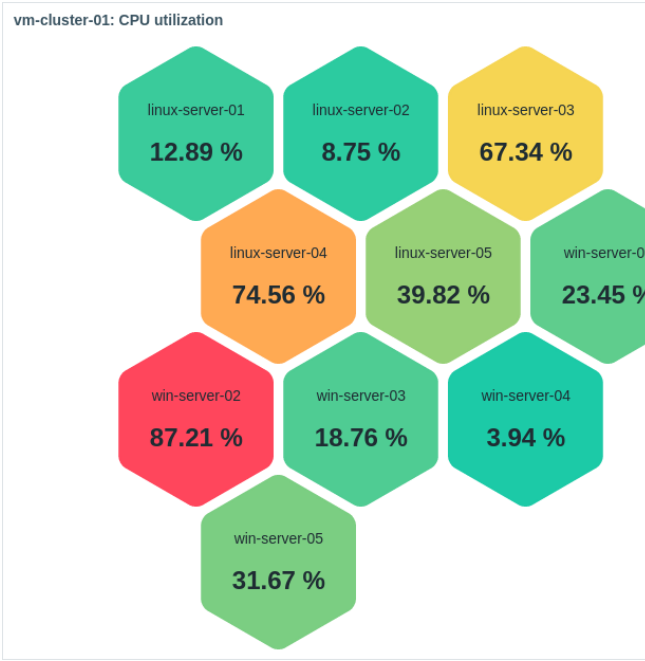


Honeycomb

Présentation

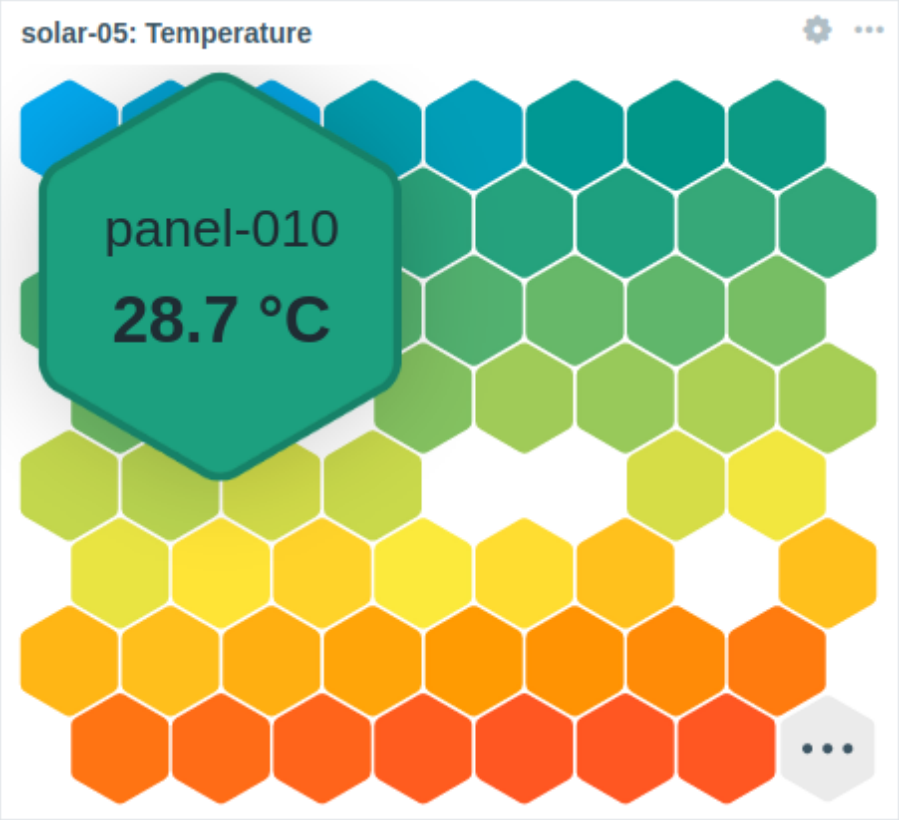
Le widget *Honeycomb* affiche les hôtes ou les éléments sous forme de cellules hexagonales dans une grille. Cette disposition permet de détecter facilement les hôtes/éléments présentant des problèmes, d'identifier les regroupements de problèmes, de comparer des groupes en un coup d'œil, etc.





Le nombre de cellules affichées est limité par la taille du widget et la taille minimale des cellules (32px). Si toutes les cellules ne tiennent pas, la dernière cellule affiche des points de suspension. Le widget peut être redimensionné, le nombre, la taille et le positionnement des cellules s'ajustant dynamiquement.

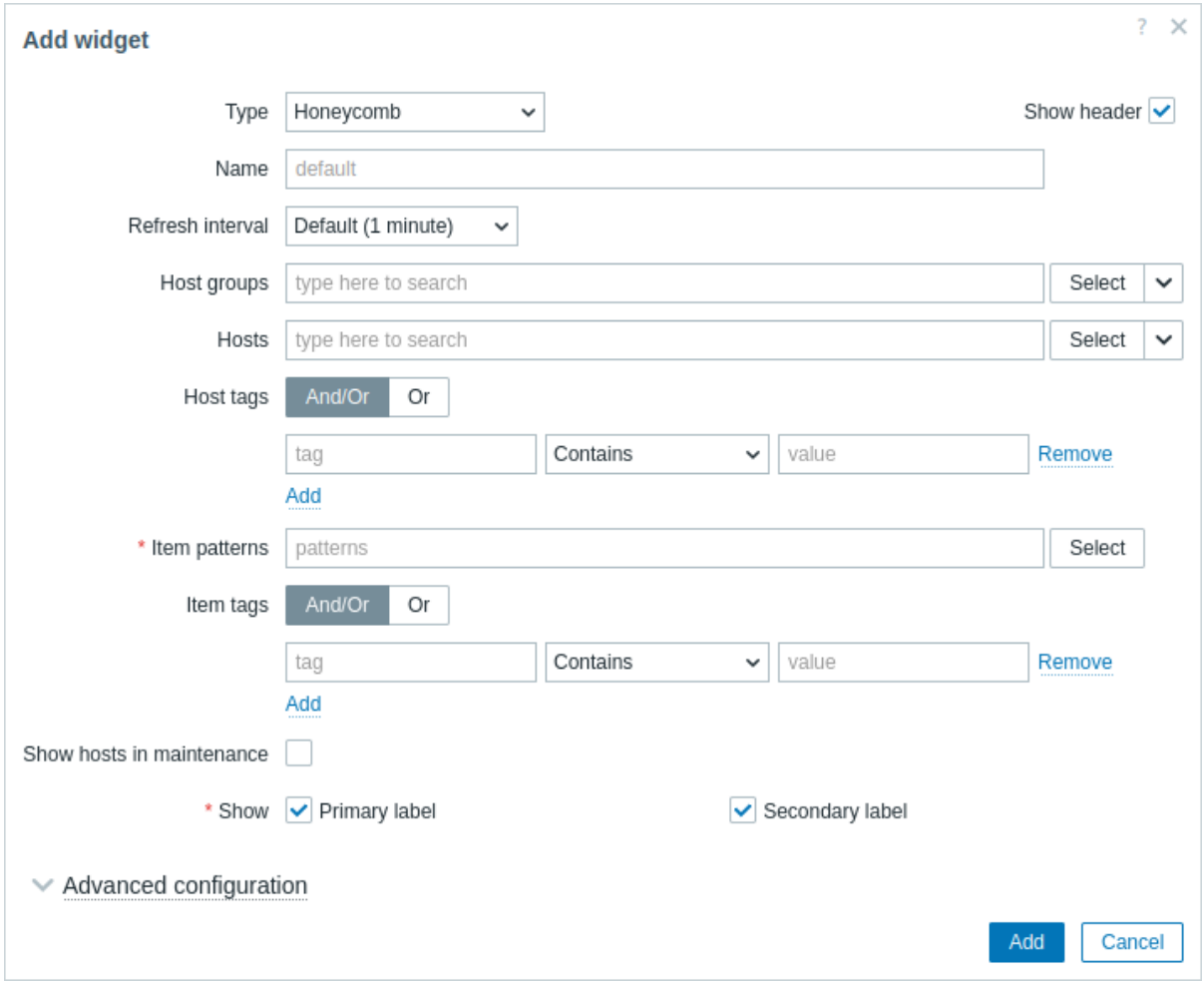
Au survol de la souris, la cellule ciblée s'agrandit pour une meilleure visibilité. Cliquer sur une cellule met sa bordure en évidence jusqu'à ce qu'une autre cellule soit sélectionnée.



Les informations affichées dans le widget *Honeycomb* peuvent être téléchargées sous forme d'image PNG en sélectionnant l'option *Télécharger l'image* dans le menu *Actions* du widget …, qui apparaît lorsque vous survolez le widget. Selon les paramètres de votre navigateur, l'image est soit enregistrée automatiquement dans le dossier `Downloads`, soit une boîte de dialogue *Enregistrer* s'ouvre, vous permettant de choisir l'emplacement.

Configuration

Pour configurer, sélectionnez *Honeycomb* comme type :



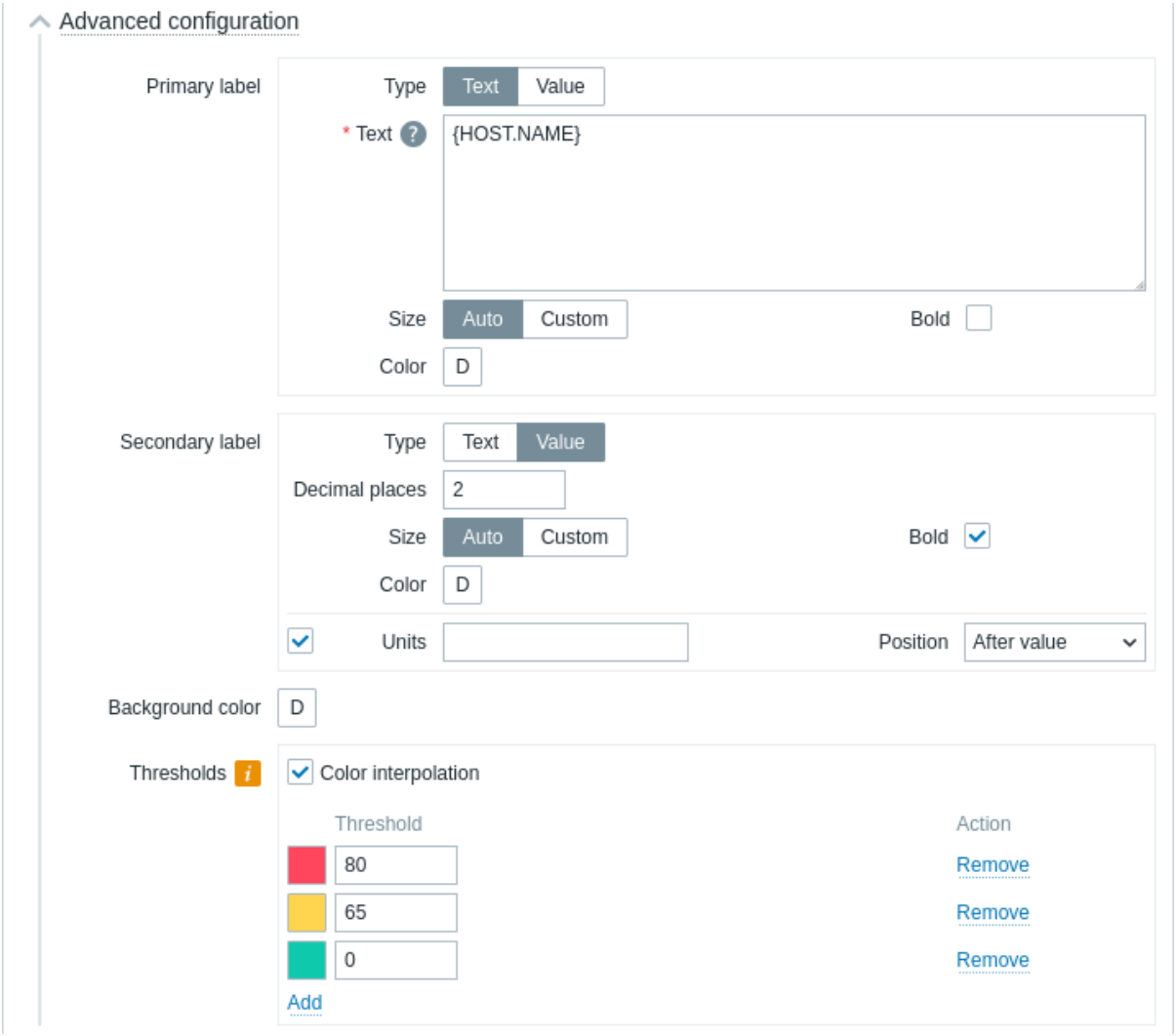
En plus des paramètres communs à tous les widgets, vous pouvez définir les options spécifiques suivantes :

Host groups	Sélectionnez des groupes d'hôtes. Vous pouvez également sélectionner un widget compatible comme source de données pour les groupes d'hôtes. Ce champ prend en charge l'autocomplétion ; ainsi, lorsque vous commencez à saisir le nom d'un groupe, une liste déroulante de groupes correspondants s'affiche. La sélection d'un groupe d'hôtes parent sélectionne implicitement tous les groupes d'hôtes imbriqués ; si aucun groupe d'hôtes n'est sélectionné, le widget affichera tous les groupes d'hôtes contenant des hôtes avec des éléments correspondant au *Modèle d'élément* sélectionné (voir ci-dessous). Ce paramètre n'est pas disponible lors de la configuration du widget sur un tableau de bord de modèle.
Hosts	Sélectionnez des hôtes. Vous pouvez également sélectionner un widget compatible ou le tableau de bord comme source de données pour les hôtes. Ce champ prend en charge l'autocomplétion ; ainsi, lorsque vous commencez à saisir le nom d'un hôte, une liste déroulante d'hôtes correspondants s'affiche. Si aucun hôte n'est sélectionné, le widget affichera tous les hôtes avec des éléments correspondant au *Modèle d'élément* sélectionné (voir ci-dessous). Ce paramètre n'est pas disponible lors de la configuration du widget sur un tableau de bord de modèle.

Host tags	Spécifiez des tags pour limiter le nombre d'hôtes affichés dans le widget. Il est possible d'inclure ou d'exclure des tags et des valeurs de tag spécifiques. Plusieurs conditions peuvent être définies. La correspondance des noms de tag est toujours sensible à la casse. Plusieurs opérateurs sont disponibles pour chaque condition : **Exists** - inclure les noms de tag spécifiés ; **Equals** - inclure les noms de tag et les valeurs spécifiés (sensible à la casse) ; **Contains** - inclure les noms de tag spécifiés lorsque les valeurs de tag contiennent la chaîne saisie (correspondance de sous-chaîne, insensible à la casse) ; **Does not exist** - exclure les noms de tag spécifiés ; **Does not equal** - exclure les noms de tag et les valeurs spécifiés (sensible à la casse) ; **Does not contain** - exclure les noms de tag spécifiés lorsque les valeurs de tag contiennent la chaîne saisie (correspondance de sous-chaîne, insensible à la casse). Deux types de calcul sont disponibles pour les conditions : **And/Or** - toutes les conditions doivent être remplies, les conditions ayant le même nom de tag seront regroupées par la condition Or ; **Or** - il suffit qu'une seule condition soit remplie. Ce paramètre n'est pas disponible lors de la configuration du widget sur un tableau de bord de modèle.
Item patterns	Saisissez des modèles d'élément ou sélectionnez des éléments existants comme modèles d'élément. Les données des éléments correspondant aux modèles saisis ou sélectionnés seront affichées dans le honeycomb. Le paramètre *Item patterns* est obligatoire. Des modèles génériques peuvent être utilisés pour la sélection (par exemple, `*` renverra les éléments correspondant à zéro ou plusieurs caractères ; `Zabbix*` renverra les éléments commençant par "Zabbix"). Pour spécifier un modèle générique, saisissez la chaîne manuellement et appuyez sur *Entrée*. Lorsque vous commencez à saisir, une liste déroulante affiche les éléments correspondants, limitée à ceux appartenant aux *Hosts* sélectionnés ou aux hôtes des *Host groups* sélectionnés, le cas échéant. Le symbole générique est toujours interprété ; il n'est donc pas possible d'ajouter individuellement, par exemple, un élément nommé *item** s'il existe d'autres éléments correspondants (par exemple, item2, item3). Lors de la configuration du widget sur un tableau de bord de modèle, ce paramètre permet de sélectionner uniquement les éléments configurés sur le modèle.
Item tags	Spécifiez des tags pour limiter le nombre d'éléments affichés dans le widget. Pour plus d'informations, voir *Host tags* ci-dessus.
Show hosts in maintenance	Cochez cette case pour afficher les hôtes en maintenance (dans ce cas, l'icône de maintenance sera affichée à côté du nom de l'hôte). Ce paramètre est intitulé *Show data in maintenance* lors de la configuration du widget sur un tableau de bord de modèle.
Show	Cochez cette case pour afficher l'élément de cellule honeycomb correspondant : étiquette principale, étiquette secondaire. Au moins un élément doit être sélectionné.
Advanced configuration	Cliquez sur l'en-tête *Advanced configuration* pour afficher les options de configuration avancée.

Configuration avancée

Les options de configuration avancée sont disponibles dans la section repliable *Configuration avancée* et uniquement pour les éléments sélectionnés dans le champ *Afficher* (voir ci-dessus), ainsi que pour la couleur d'arrière-plan ou les seuils des cellules en nid d'abeille.



Étiquette principale/secondaire

Type	Sélectionnez le type d'étiquette : **Texte** - l'étiquette affichera le texte spécifié pour le paramètre *Texte* ; **Valeur** - l'étiquette affichera la valeur de l'élément avec le nombre de décimales spécifié pour le paramètre *Décimales*.
Texte	Saisissez le texte de l'étiquette. Ce texte peut remplacer le nom d'élément par défaut. Le texte multiligne est pris en charge. Une combinaison de texte et de macros prises en charge est possible. `{HOST.\*}`, `{ITEM.\*}`, `{INVENTORY.\*}` et les macros utilisateur sont prises en charge. Les cellules en nid d'abeille sont triées par ordre alphabétique selon le nom de l'hôte et, pour chaque hôte, selon le nom de l'élément. Ce paramètre est disponible si *Type* est défini sur *Texte*.
Décimales	Saisissez le nombre de décimales à afficher avec la valeur. Ce paramètre est disponible si *Type* est défini sur *Valeur* et n'affecte que les éléments qui renvoient des données numériques (flottant).
Taille	Sélectionnez la taille de l'étiquette : **Auto** - utiliser une taille d'étiquette ajustée automatiquement ; **Personnalisée** - saisir une taille d'étiquette personnalisée (en pourcentage, par rapport à la taille de la cellule en nid d'abeille). Notez que les étiquettes qui ne tiennent pas dans la taille de la cellule en nid d'abeille sont tronquées.
Gras	Cochez la case pour afficher les étiquettes en gras.
Couleur	Sélectionnez la couleur de l'étiquette dans le sélecteur de couleurs. "D" représente la couleur par défaut, qui dépend du thème de l'interface. Pour revenir à la couleur par défaut, cliquez sur le bouton *Utiliser la valeur par défaut* dans le sélecteur de couleurs.

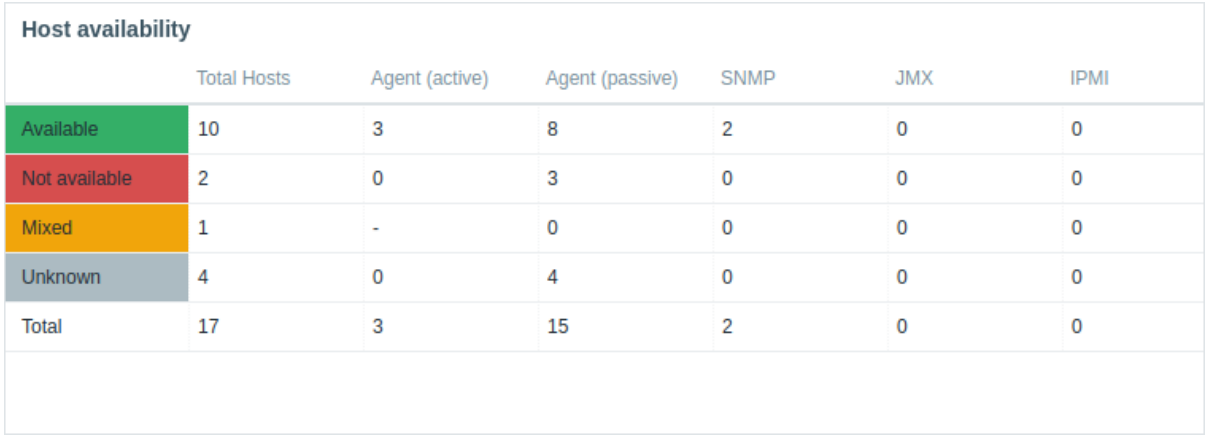
Unités

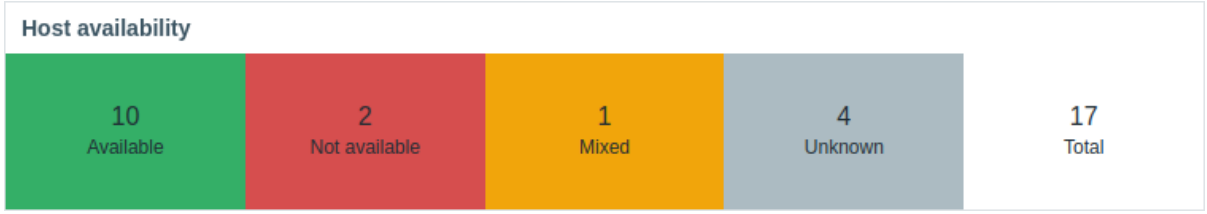
Unités	Cochez la case pour afficher les unités avec la valeur de l'élément. Si vous saisissez un nom d'unité, il remplacera les unités définies dans la configuration de l'élément. Ce paramètre est disponible si *Type* est défini sur *Valeur*.
Position	Sélectionnez la position des unités de l'élément (avant ou après la valeur de l'élément). Ce paramètre est ignoré pour les unités liées au temps suivantes : unixtime, uptime, s. Ce paramètre est disponible si *Type* est défini sur *Valeur* et que l'option *Unités* est sélectionnée.
Couleur d'arrière-plan	
Couleur d'arrière-plan	Sélectionnez la couleur d'arrière-plan des cellules en nid d'abeille dans le sélecteur de couleurs. "D" représente la couleur par défaut, qui dépend du thème de l'interface. Pour revenir à la couleur par défaut, cliquez sur le bouton *Utiliser la valeur par défaut* dans le sélecteur de couleurs.
Seuils	
Interpolation des couleurs	Cochez la case pour activer une transition fluide entre les couleurs de seuil des cellules en nid d'abeille. Ce paramètre est disponible si deux seuils ou plus sont définis.
Seuil	Cliquez sur *Ajouter* pour ajouter un seuil, sélectionnez une couleur de seuil dans le sélecteur de couleurs et spécifiez une valeur numérique. La liste des seuils sera triée par ordre croissant lors de l'enregistrement. Notez que les couleurs configurées comme seuils ne s'afficheront correctement que pour les éléments numériques. Les suffixes (par exemple, "1d", "2w", "4K", "8G") sont pris en charge. Les correspondances de valeurs sont prises en charge.

Disponibilité de l'hôte

Vue d'ensemble

Le widget *Disponibilité des hôtes* affiche la disponibilité des hôtes dans les groupes d'hôtes sélectionnés, ce qui vous permet de surveiller le nombre d'hôtes opérationnels ou non opérationnels.



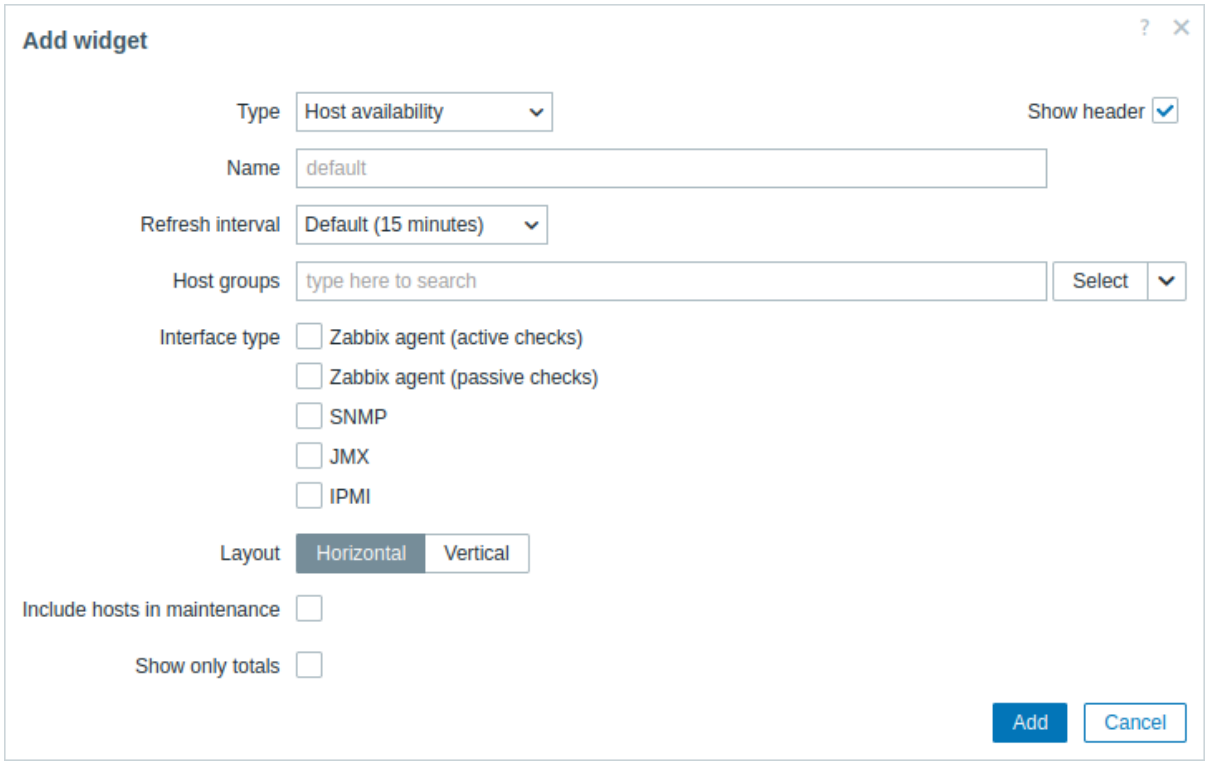


La disponibilité des hôtes est comptabilisée comme suit :

- *Disponible* - hôtes dont toutes les interfaces sont disponibles
- *Non disponible* - hôtes dont toutes les interfaces sont non disponibles
- *Mixte* - hôtes ayant au moins une interface non disponible et au moins une interface disponible ou inconnue
- *Inconnu* - hôtes ayant au moins une interface inconnue, mais aucune non disponible
- *Total* - total de tous les hôtes

Configuration

Pour configurer, sélectionnez *Disponibilité des hôtes* comme type :



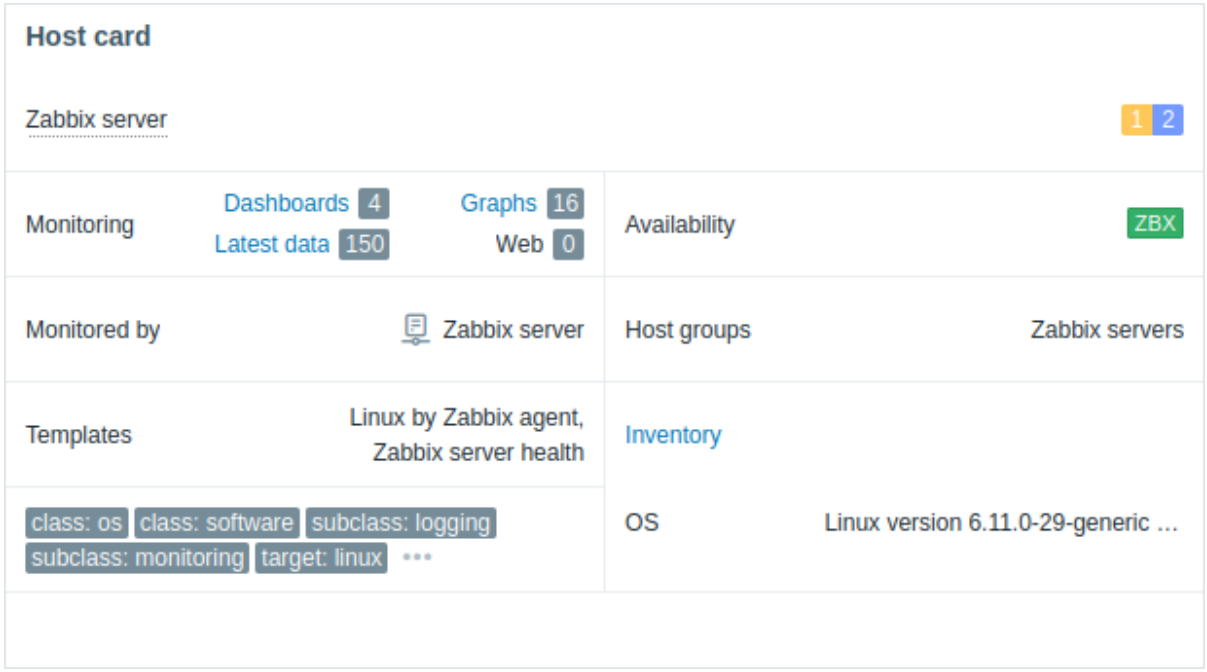
En plus des paramètres communs à tous les widgets, vous pouvez définir les options spécifiques suivantes :

Groupes d'hôtes	Sélectionnez des groupes d'hôtes. Vous pouvez également sélectionner un widget compatible comme source de données pour les groupes d'hôtes. Ce champ prend en charge l'autocomplétion ; ainsi, lorsque vous commencez à saisir le nom d'un groupe, une liste déroulante des groupes correspondants s'affiche. Ce paramètre n'est pas disponible lors de la configuration du widget sur un tableau de bord de modèle.
Type d'interface	Sélectionnez les interfaces d'hôte pour lesquelles vous souhaitez afficher les données de disponibilité — *Agent Zabbix (contrôles actifs)*, *Agent Zabbix (contrôles passifs)*, *SNMP*, *JMX*, *IPMI*. Pour *Agent Zabbix (contrôles actifs)*, la cellule *Mixte* sera toujours vide, car ce type d'éléments ne peut pas avoir plusieurs interfaces. La disponibilité de toutes les interfaces est affichée par défaut si rien n'est sélectionné.
Disposition	Sélectionnez un affichage horizontal (colonnes) ou vertical (lignes).
Inclure les hôtes en maintenance	Incluez dans les statistiques les hôtes en maintenance. Ce paramètre est intitulé *Afficher les données en maintenance* lors de la configuration du widget sur un tableau de bord de modèle.
Afficher uniquement les totaux	Si cette option est cochée, seul le total des hôtes, sans ventilation par interfaces, est affiché. Cette option est désactivée si une seule interface est sélectionnée.

Carte d'hôte

Aperçu

Le widget *Carte d'hôte* affiche les détails d'un seul hôte, ce qui vous permet d'évaluer rapidement l'état et la configuration d'un hôte (disponibilité, problèmes, inventaire, etc.).

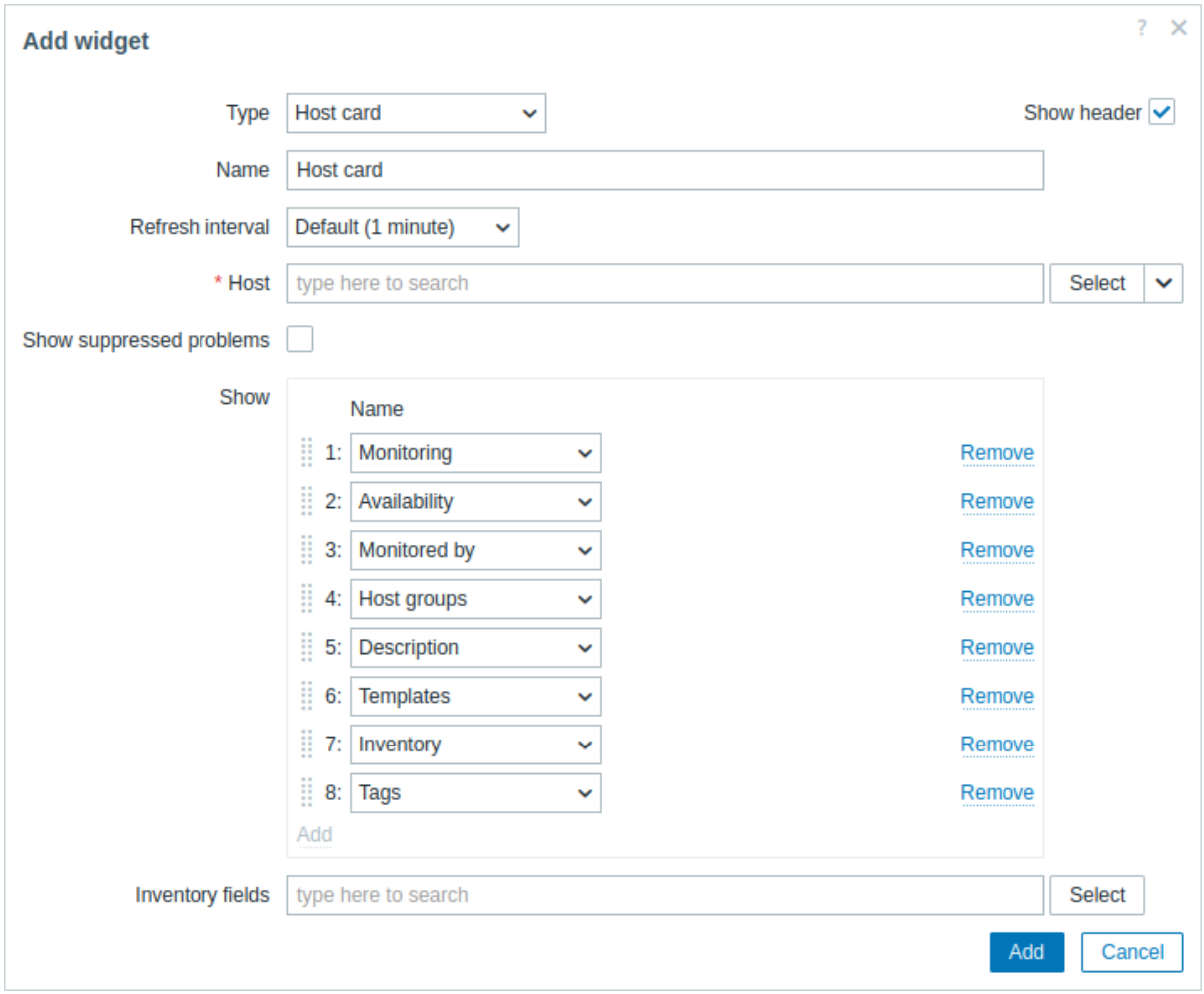


L'hôte peut être défini directement dans la configuration du widget ou sélectionné à partir d'un widget compatible ou du sélecteur d'hôte sur le tableau de bord.

Le widget peut afficher plusieurs sections, chacune présentant des informations différentes. La disposition des sections est ajustée automatiquement en fonction de la largeur du widget ; lorsqu'il est étendu horizontalement, les sections sont automatiquement réorganisées en plusieurs colonnes.

Configuration

Pour configurer, sélectionnez *Carte d'hôte* comme type :



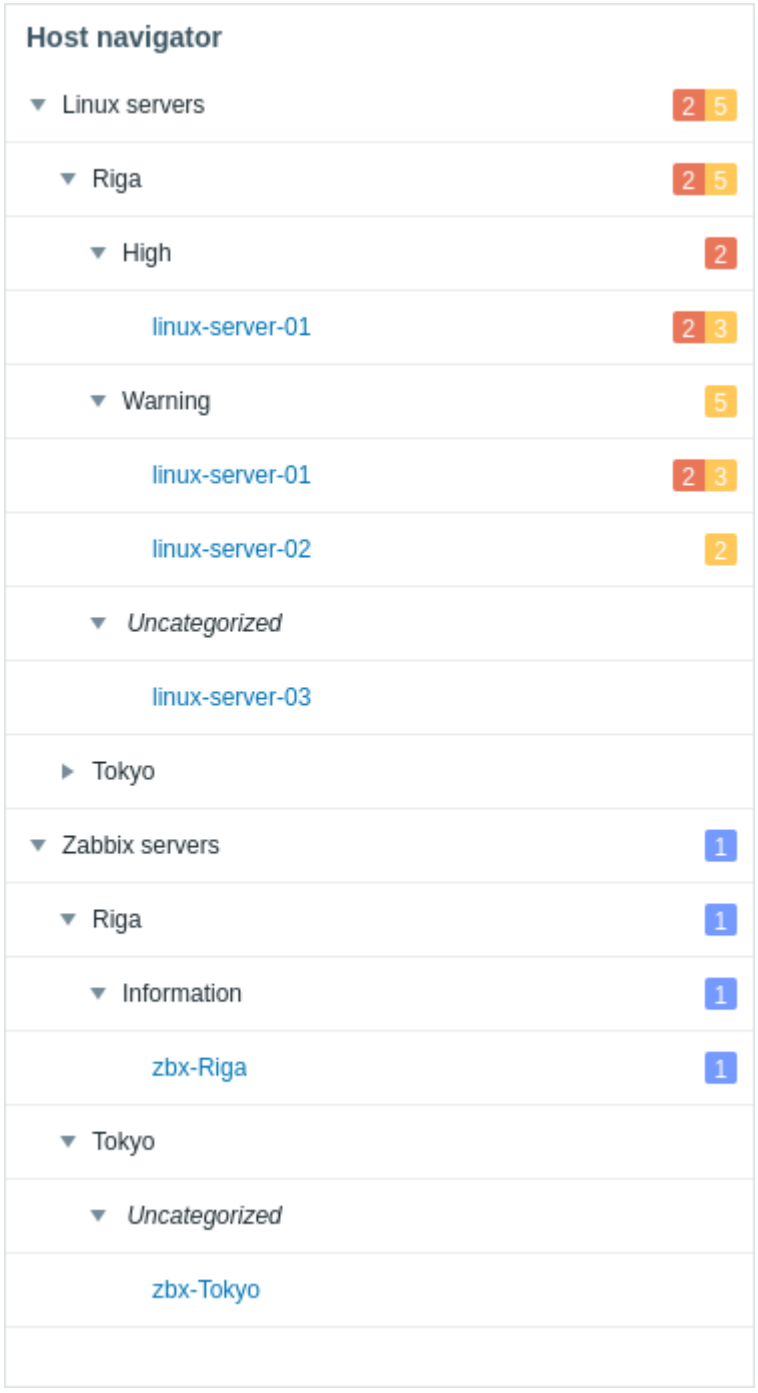
En plus des paramètres communs à tous les widgets, vous pouvez définir les paramètres spécifiques suivants :

Hôte	Sélectionnez l'hôte. Vous pouvez également sélectionner un widget compatible ou le tableau de bord comme source de données pour les hôtes. Des suggestions correspondantes apparaissent au fur et à mesure de la saisie. La section supérieure du widget affiche toujours le **nom de l'hôte** (un clic dessus ouvre le menu de l'hôte) : - Si l'hôte est désactivé, une étiquette rouge *Désactivé* s'affiche à côté du nom de l'hôte. - Si l'hôte est en maintenance, l'icône de maintenance s'affiche à côté du nom de l'hôte. La section supérieure du widget affiche également le nombre de **problèmes de l'hôte** regroupés par gravité (un clic dessus ouvre les problèmes de l'hôte). Ce paramètre n'est pas disponible lors de la configuration du widget sur un tableau de bord de modèle.
Afficher les problèmes supprimés	Cochez cette case pour inclure les problèmes supprimés dans le calcul des problèmes.
Afficher	Ajoutez des sections à afficher dans le widget. Les sections peuvent être réorganisées par glisser-déposer.
Groupes d'hôtes	Affiche les groupes d'hôtes auxquels l'hôte appartient.
Description	Affiche la description de l'hôte.
Surveillance	Affiche des liens rapides pour accéder aux Tableaux de bord, aux Dernières données, aux Graphiques et aux Scénarios web de l'hôte. Le nombre à côté de chaque lien indique le nombre d'entités correspondantes.
Disponibilité	Affiche la disponibilité de l'hôte par interface.
Surveillé par	Indique si l'hôte est surveillé par le serveur Zabbix ou par un proxy spécifique ou un groupe de proxys.
Modèles	Affiche les modèles liés à l'hôte.
Inventaire	Affiche les champs d'inventaire de l'hôte.
Tags	Affiche les tags de l'hôte.
Champs d'inventaire	Sélectionnez les champs d'inventaire à afficher. Des suggestions correspondantes apparaissent au fur et à mesure de la saisie. Si aucun champ d'inventaire n'est spécifié, tous les champs d'inventaire renseignés seront affichés. Ce paramètre est disponible si *Afficher* contient *Inventaire*.

Navigateur d'hôte

Vue d'ensemble

Le widget *Navigateur d'hôtes* affiche une liste d'hôtes selon diverses options de filtrage et de regroupement.



Les groupes selon lesquels les hôtes sont organisés peuvent être développés ou réduits.

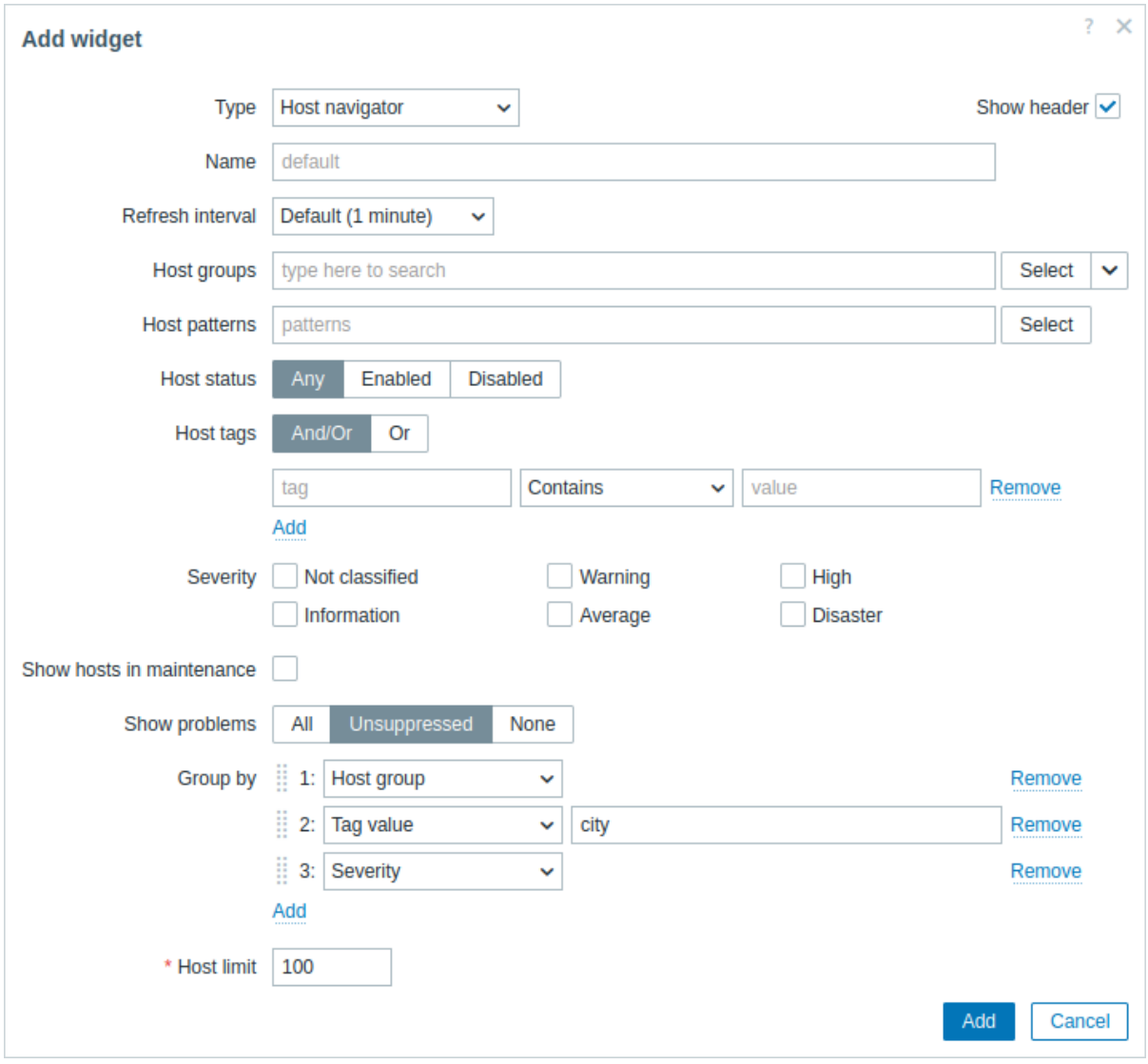
Pour les groupes, les problèmes et les hôtes en maintenance, des détails supplémentaires sont accessibles via des infobulles au survol de la souris.

Le widget est particulièrement utile pour contrôler ce que les autres widgets affichent en fonction de l'hôte sélectionné.



Configuration

Pour configurer le widget, sélectionnez *Host navigator* comme type :



En plus des paramètres communs à tous les widgets, vous pouvez définir les options spécifiques suivantes :

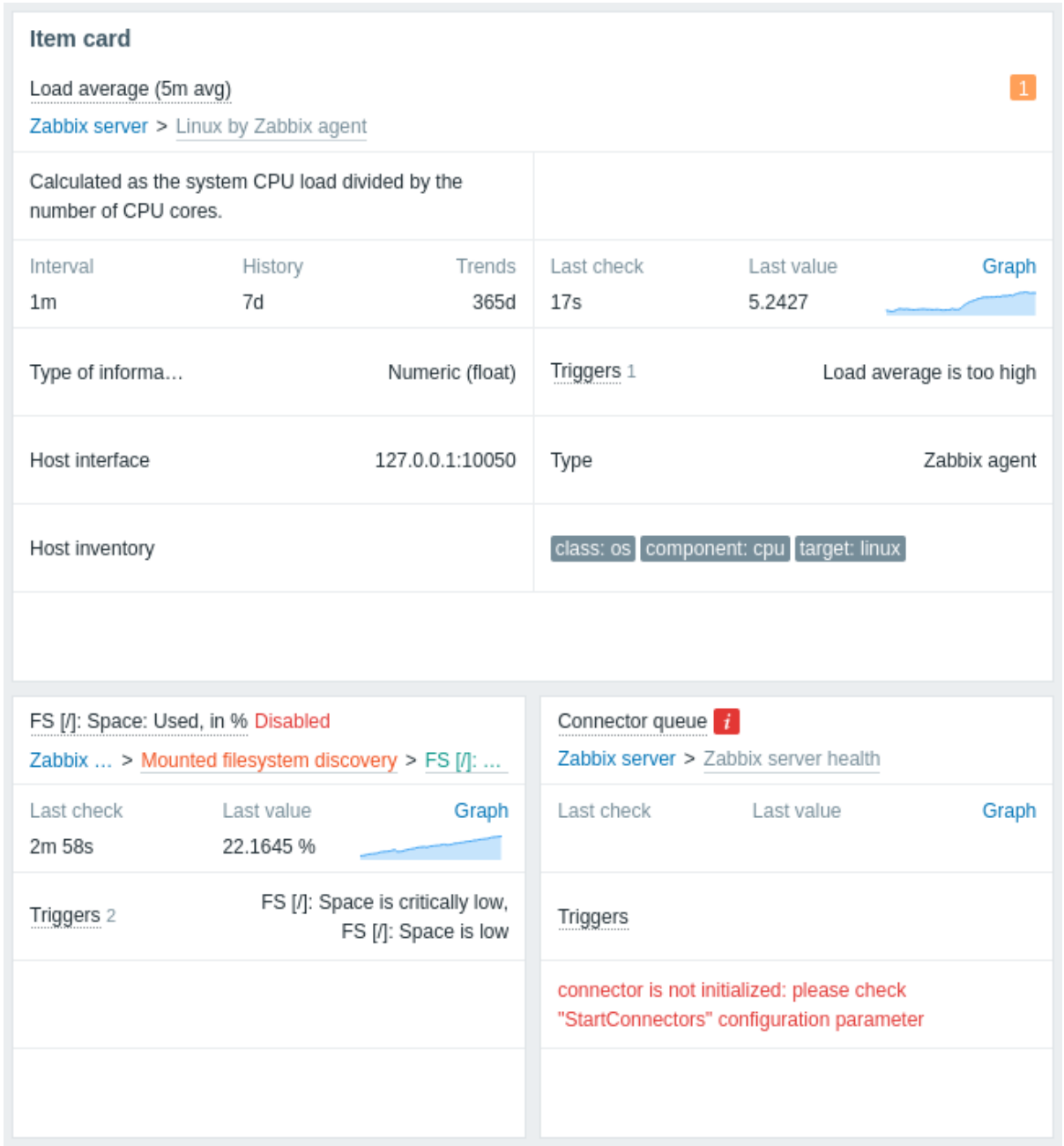
Host groups	Sélectionnez des groupes d'hôtes. Vous pouvez également sélectionner un widget compatible comme source de données pour les groupes d'hôtes. Ce champ propose la saisie semi-automatique : commencez à saisir le nom d'un groupe pour afficher une liste déroulante des groupes correspondants. La sélection d'un groupe d'hôtes parent sélectionne implicitement tous les groupes d'hôtes imbriqués ; si aucun groupe d'hôtes n'est sélectionné, le widget affiche tous les hôtes de tous les groupes d'hôtes. Ce paramètre n'est pas disponible lors de la configuration du widget sur un tableau de bord de modèle.
Host patterns	Saisissez des motifs d'hôtes ou sélectionnez des hôtes existants comme motifs d'hôtes. Les hôtes correspondant aux motifs spécifiés seront affichés dans le navigateur d'hôtes. Ce champ propose la saisie semi-automatique : commencez à saisir le nom d'un hôte pour afficher une liste déroulante des hôtes correspondants. Si aucun hôte n'est sélectionné, le widget affiche tous les hôtes. Des motifs génériques peuvent être utilisés pour la sélection (par exemple, ∗ renvoie les hôtes correspondant à zéro caractère ou plus ; `Zabbix∗` renvoie les hôtes dont le nom commence par « Zabbix »). Pour spécifier un motif générique, saisissez la chaîne manuellement et appuyez sur *Entrée*. Lorsque vous commencez à saisir du texte, une liste déroulante affiche les hôtes correspondants, limitée aux hôtes appartenant aux hôtes des *Host groups* sélectionnés, le cas échéant. Ce paramètre n'est pas disponible lors de la configuration du widget sur un tableau de bord de modèle.
Host status	Filtrez les hôtes à afficher en fonction de leur statut (tous, activés, désactivés). Ce paramètre n'est pas disponible lors de la configuration du widget sur un tableau de bord de modèle.
Host tags	Spécifiez des tags pour filtrer les hôtes affichés dans le widget. Il est possible d'inclure ou d'exclure des tags et des valeurs de tags spécifiques. Plusieurs conditions peuvent être définies. La correspondance des noms de tags est toujours sensible à la casse. Plusieurs opérateurs sont disponibles pour chaque condition : **Exists** - inclure les noms de tags spécifiés ; **Equals** - inclure les noms et les valeurs de tags spécifiés (sensible à la casse) ; **Contains** - inclure les noms de tags dont les valeurs contiennent la chaîne saisie (correspondance de sous-chaîne, insensible à la casse) ; **Does not exist** - exclure les noms de tags spécifiés ; **Does not equal** - exclure les noms et les valeurs de tags spécifiés (sensible à la casse) ; **Does not contain** - exclure les noms de tags dont les valeurs contiennent la chaîne saisie (correspondance de sous-chaîne, insensible à la casse). Il existe deux types de calcul pour les conditions : **And/Or** - toutes les conditions doivent être remplies ; les conditions ayant le même nom de tag sont regroupées avec la condition Or ; **Or** - une seule condition remplie suffit. Ce paramètre n'est pas disponible lors de la configuration du widget sur un tableau de bord de modèle.
Severity	Sélectionnez les niveaux de gravité des problèmes afin de filtrer les hôtes présentant des problèmes à afficher dans le widget. Si aucun niveau de gravité n'est sélectionné, tous les hôtes présentant des problèmes, quel que soit leur niveau de gravité, seront affichés.
Show hosts in maintenance	Cochez cette case pour afficher les hôtes en maintenance (dans ce cas, l'icône de maintenance est affichée à côté du nom de l'hôte). Ce paramètre est intitulé *Show data in maintenance* lors de la configuration du widget sur un tableau de bord de modèle.

Show problems	Filtrez les problèmes à afficher en fonction de leur statut (tous, non supprimés, aucun). Les problèmes sont indiqués par des blocs colorés à la fin de la ligne de l'hôte. Les couleurs sont basées sur le niveau de gravité du problème, qui peut être personnalisé dans *Administration > General > Trigger displaying options*. Les niveaux de gravité des problèmes peuvent être modifiés lors de la mise à jour des problèmes.
Group by	Ajoutez un attribut de regroupement selon lequel regrouper les hôtes sélectionnés : **Host group** - regrouper les hôtes par groupe d'hôtes ; **Tag value** - saisissez un nom de tag pour regrouper les hôtes selon les valeurs de ce tag (par exemple, saisissez « city » pour regrouper les hôtes selon les valeurs « Riga », « Tokyo », etc.) ; **Severity** - regrouper les hôtes selon le niveau de gravité de leurs problèmes. Si *Show problems* est configuré pour afficher les problèmes, ceux-ci sont affichés comme suit : - pour chaque groupe de niveaux de gravité, seul le nombre de problèmes correspondant est affiché ; - pour chaque hôte, le nombre de problèmes correspondant à tous ses niveaux de gravité est affiché. Notez que les hôtes seront regroupés uniquement selon les niveaux de gravité sélectionnés dans le paramètre *Severity* ; si aucun niveau de gravité n'est sélectionné, tous les hôtes seront regroupés selon tous les niveaux de gravité. Les attributs de regroupement peuvent être réordonnés en les faisant glisser vers le haut ou vers le bas à l'aide de la poignée située avant le nom du groupe. Notez que l'ordre des attributs de regroupement détermine l'ordre d'imbrication des groupes. Par exemple, la spécification de plusieurs noms de tags (1 : color, 2 : city) regroupera les hôtes par couleur (rouge, bleu, etc.), puis par ville (Riga, Tokyo, etc.). Un hôte peut être affiché dans plusieurs groupes selon les attributs de regroupement configurés (par exemple, lors d'un regroupement par groupe d'hôtes si l'hôte appartient à plusieurs groupes d'hôtes). Cliquer sur ces hôtes les sélectionne et les met en évidence dans tous les groupes. Les hôtes qui ne correspondent pas aux attributs de regroupement configurés sont affichés dans le groupe *Uncategorized*. Jusqu'à 10 attributs de regroupement peuvent être spécifiés et tous doivent être uniques. Si aucun attribut de regroupement n'est spécifié, les hôtes ne sont pas regroupés.
Host limit	Saisissez le nombre maximal d'hôtes à afficher. Les valeurs possibles vont de 1 à 9999. Lorsque le nombre d'hôtes disponibles à afficher est supérieur à la limite définie, un message correspondant est affiché sous les hôtes affichés (par exemple, « 100 of 100+ hosts are shown »). Notez que la limite d'hôtes configurée affecte également l'affichage des groupes configurés ; par exemple, si la limite d'hôtes est définie sur 100 et que les hôtes sont regroupés par valeurs de tags (plus de 200), seules les 100 premières valeurs de tags avec les hôtes correspondants seront affichées dans le widget. Ce paramètre n'est pas affecté par le paramètre *Limit for search and filter results* dans *Administration > General > GUI*. Ce paramètre n'est pas disponible lors de la configuration du widget sur un tableau de bord de modèle.

Carte d'élément

Vue d'ensemble

Le widget *Carte d'élément* affiche les détails d'un seul élément, ce qui vous permet d'évaluer rapidement l'état et la configuration d'un élément (dernières données, erreurs, description, etc.).

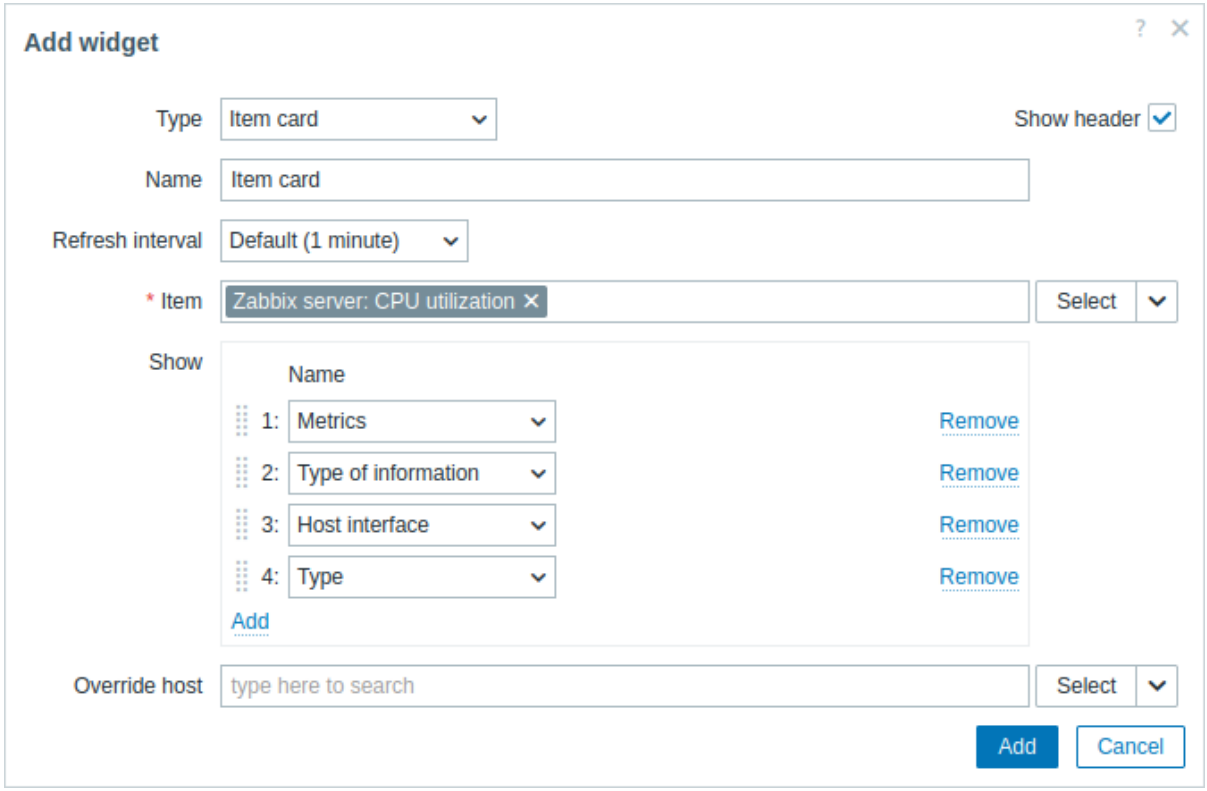


L'élément peut être défini directement dans la configuration du widget ou sélectionné à partir d'un widget compatible sur le tableau de bord.

Le widget peut afficher plusieurs sections, chacune présentant des informations différentes. La disposition des sections est ajustée automatiquement en fonction de la largeur du widget ; lorsqu'il est étendu horizontalement, les sections sont automatiquement réorganisées en plusieurs colonnes.

Configuration

Pour configurer, sélectionnez *Item card* comme type :



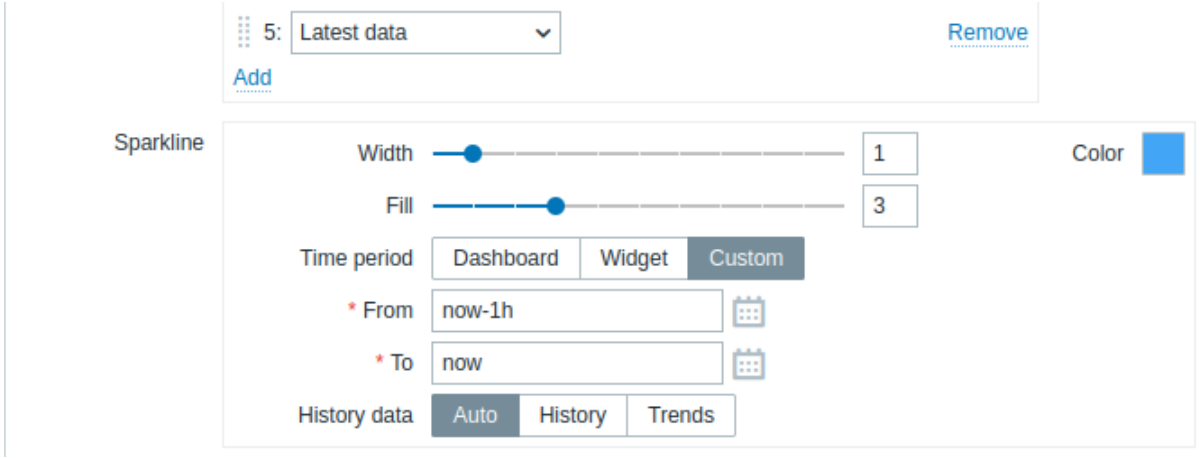
En plus des paramètres communs à tous les widgets, vous pouvez définir les options spécifiques suivantes :

Item	Sélectionnez l'élément. Vous pouvez également sélectionner un widget compatible comme source de données pour les éléments. Des suggestions correspondantes apparaissent au fur et à mesure de la saisie. La section supérieure du widget affiche toujours le **nom de l'élément** (un clic dessus ouvre le menu de l'élément) : - Si l'élément est désactivé, une étiquette rouge *Disabled* s'affiche à côté du nom de l'élément. - En cas d'erreurs, une icône carrée avec la lettre « i » s'affiche. En cas de problèmes, une icône carrée avec le nombre de problèmes s'affiche pour chaque sévérité de problème. Le survol d'une icône ouvre une fenêtre contextuelle avec les détails. - Sur un tableau de bord de modèle, le nom technique (au lieu du nom visible) de l'élément est affiché. La section supérieure du widget affiche également le **chemin de l'élément** — son hôte et, le cas échéant, le modèle, la règle de découverte de bas niveau (LLD) ou l'élément maître (un clic sur une entité ouvre sa configuration). Le chemin de l'élément n'est affiché qu'aux utilisateurs disposant des autorisations sur les entités respectives.
Show	Ajoutez des sections à afficher dans le widget. Les sections peuvent être réorganisées par glisser-déposer.
Description	Affiche la description de l'élément. Les macros sont résolues.
Error text	Affiche l'erreur de l'élément (le cas échéant).
Metrics	Affiche les informations suivantes : - Intervalle de mise à jour de l'élément (sauf pour les éléments trapper) ; - Configuration de l'historique de l'élément (le cas échéant) ; - Configuration des tendances de l'élément (le cas échéant). Les macros sont résolues. Si l'élément utilise des intervalles personnalisés, une icône d'aide s'affiche à côté de l'intervalle de mise à jour de l'élément. Le survol de l'icône ouvre une fenêtre contextuelle avec les détails de l'intervalle personnalisé.

Latest data	Affiche les informations suivantes : - Temps écoulé depuis la dernière vérification de l'élément ; - Dernière valeur de l'élément ; - Lien vers le graphique simple/historique des valeurs de l'élément (si l'historique ou les tendances sont stockés) ; - Graphique sparkline configurable (si l'élément est numérique). Pour les éléments de type binaire, une miniature ou l'option *Show* s'affiche à la place de la dernière valeur. Le survol de la miniature ouvre une fenêtre contextuelle avec l'image. Le survol de *Show* ouvre une fenêtre contextuelle avec la valeur de l'élément (chaîne Base64). Sur un tableau de bord de modèle, *No data* s'affiche à la place de la dernière valeur de l'élément.
Type of information	Affiche le type d'information de l'élément.
Triggers	Affiche les informations suivantes : - Étiquette *Triggers* avec le nombre de déclencheurs (le survol de *Triggers* ouvre une fenêtre contextuelle avec les détails des déclencheurs) ; - Liste des noms des déclencheurs (les macros sont résolues).
Host interface	Affiche l'interface hôte utilisée par l'élément, ou *No data* si l'élément n'a pas d'interface hôte (par exemple, les éléments dépendants, les vérifications simples). Sur un tableau de bord de modèle, *No data* s'affiche.
Type	Affiche le type de l'élément.
Host inventory	Affiche le champ d'inventaire de l'hôte que la valeur de l'élément renseigne.
Tags	Affiche les tags de l'élément.
Override host	Sélectionnez un widget compatible ou le sélecteur d'hôte du tableau de bord comme source de données pour les hôtes. Ce paramètre n'est pas disponible lors de la configuration du widget sur un tableau de bord de modèle.

Sparkline

Les options de configuration du graphique sparkline sont disponibles uniquement si la section *Dernières données* est ajoutée au widget.



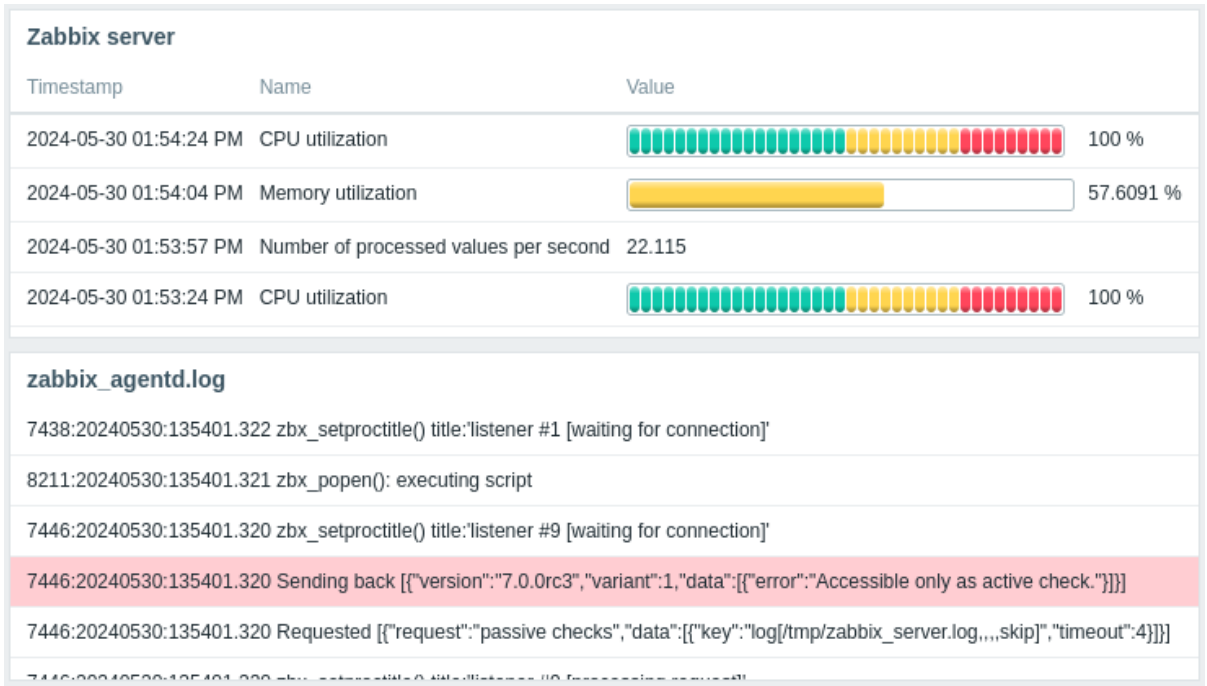
Largeur	Définissez l'épaisseur de la ligne du graphique à l'aide du curseur ou en saisissant manuellement une valeur comprise entre 0 et 10.
Couleur	Sélectionnez la couleur de la ligne et du remplissage.
Remplissage	Définissez le niveau de transparence de la couleur de remplissage à l'aide du curseur ou en saisissant manuellement une valeur comprise entre 0 et 10.

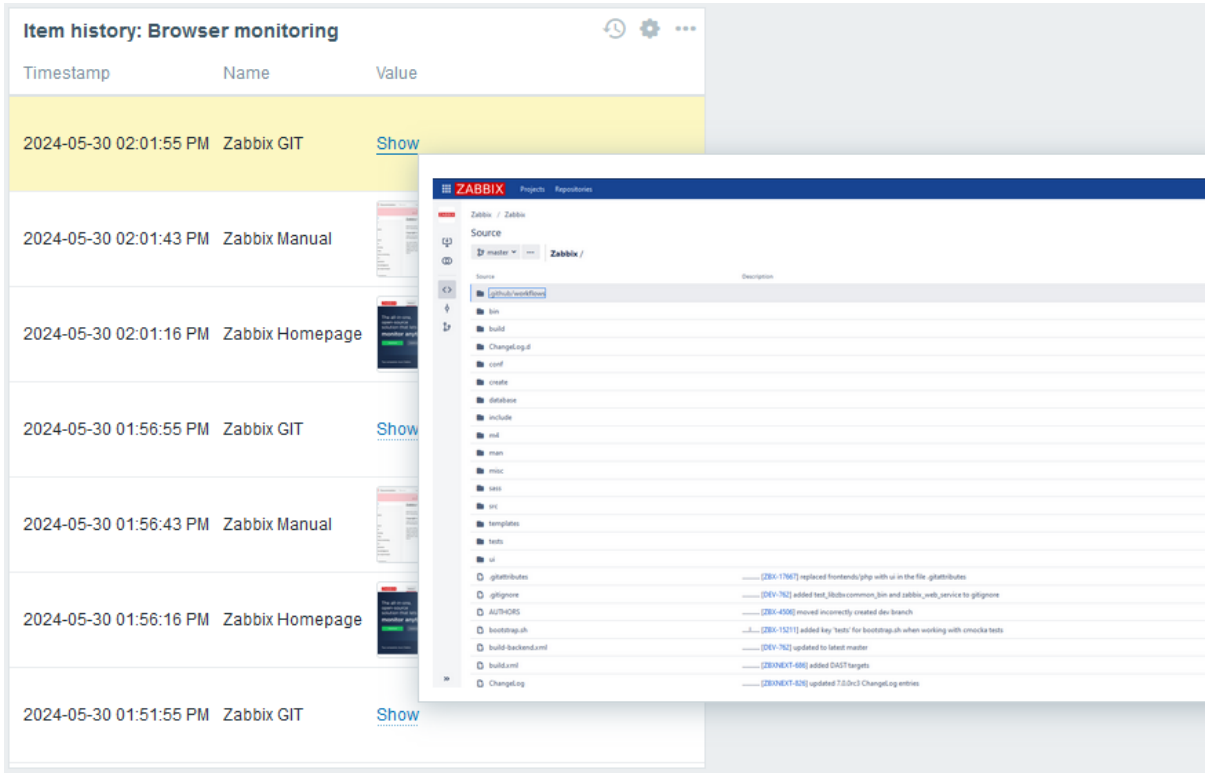
Période	Sélectionnez la source de données pour la période : **Tableau de bord** - utiliser le sélecteur de période du tableau de bord ; **Widget** - utiliser un widget compatible (défini dans le paramètre *Widget*) ; **Personnalisée** - utiliser une période personnalisée définie dans les paramètres *De* et *À* ; si elle est définie, une icône d'horloge s'affichera dans le coin supérieur droit du widget, indiquant l'heure définie au survol de la souris. Notez que, quelle que soit la configuration de *Période* du widget, les widgets compatibles peuvent toujours l'utiliser comme source de données pour la période.
Widget	Saisissez ou sélectionnez un widget compatible comme source de données pour la période. Ce paramètre est disponible si *Période* est défini sur "Widget".
De	Saisissez ou sélectionnez le début de la période. La syntaxe de temps relatif (`now`, `now/d`, `now/w-1w`, etc.) est prise en charge. Ce paramètre est disponible si *Période* est défini sur "Personnalisée".
À	Saisissez ou sélectionnez la fin de la période. La syntaxe de temps relatif (`now`, `now/d`, `now/w-1w`, etc.) est prise en charge. Ce paramètre est disponible si *Période* est défini sur "Personnalisée".
Données d'historique	Prendre les données de l'historique ou des tendances : **Auto** - sélection automatique ; **Historique** - prendre les données de l'historique ; **Tendances** - prendre les données de tendance. Ce paramètre s'applique uniquement aux données numériques. Les données non numériques seront toujours prises depuis l'historique.

Historique des éléments

Vue d'ensemble

Le widget *Historique des éléments* affiche les dernières données pour tous les types d'éléments (numérique, texte, etc.) sous forme de tableau. Il peut afficher des barres de progression et des images (utile pour les éléments de navigateur) et peut mettre en évidence des valeurs (utile pour la surveillance des fichiers journaux).

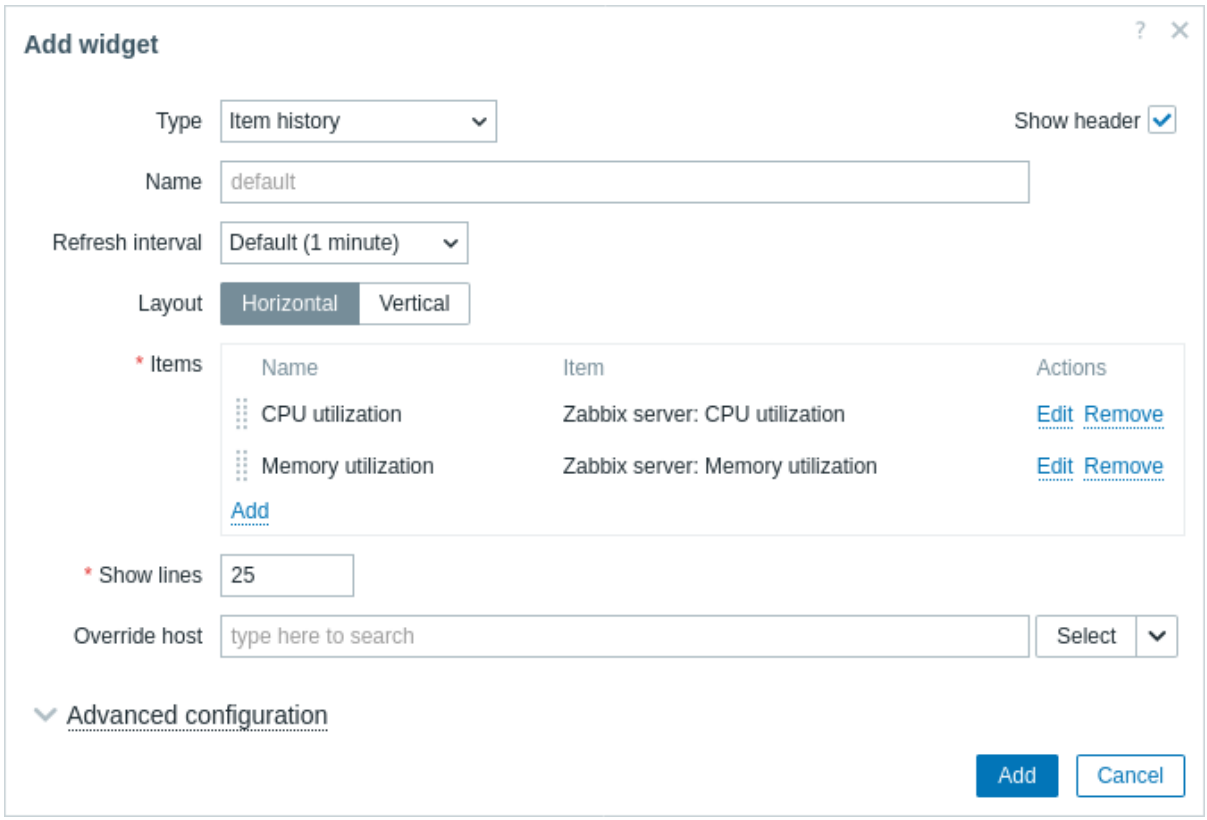




Jusqu'à 1000 enregistrements peuvent être affichés.

Configuration

Pour configurer, sélectionnez *Historique des éléments* comme type :



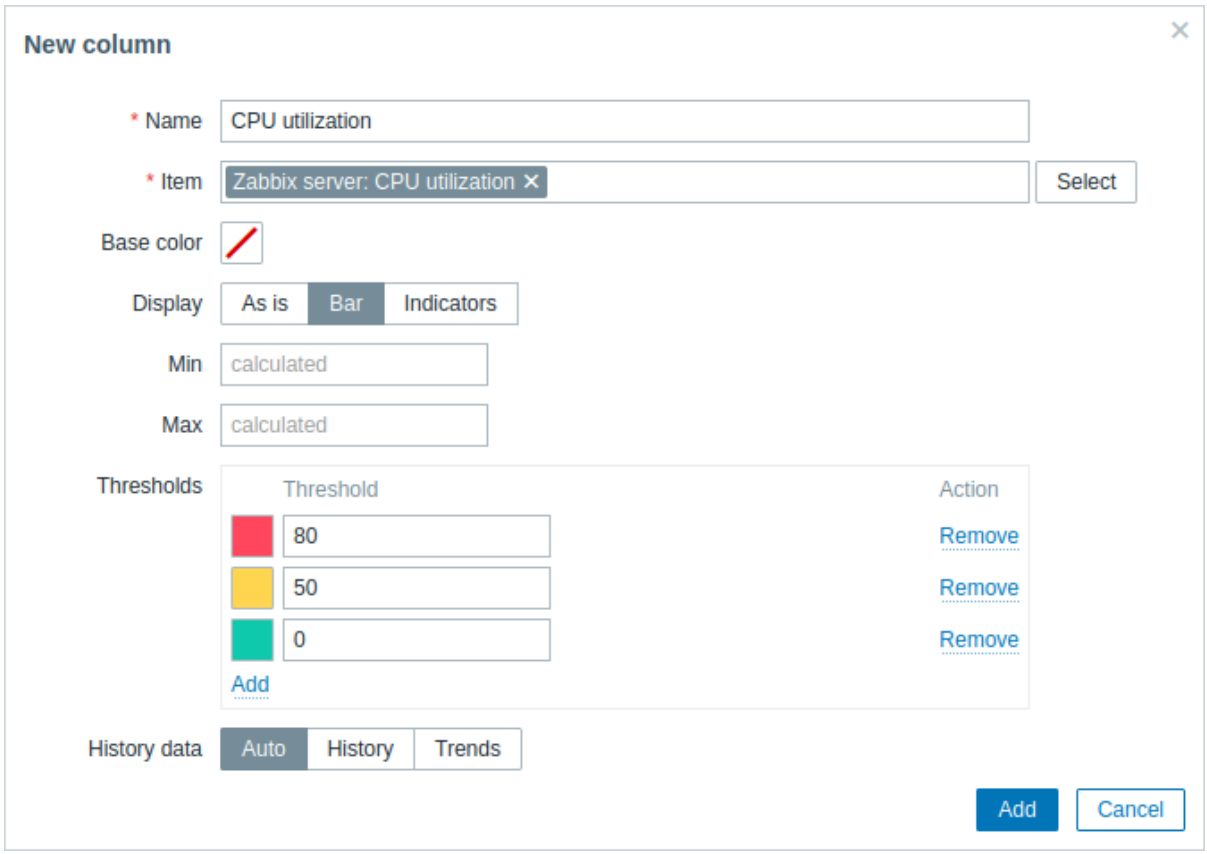
En plus des paramètres communs à tous les widgets, vous pouvez définir les options spécifiques suivantes :

Disposition	Sélectionnez l'option de disposition pour les colonnes d'éléments : **Horizontal** - les éléments seront affichés horizontalement, les valeurs verticalement ; **Vertical** - les éléments seront affichés verticalement, les valeurs horizontalement.

Éléments	Ajoutez des colonnes d'éléments à afficher. L'ordre des éléments détermine leur ordre d'affichage. Les éléments peuvent être réorganisés en les faisant glisser vers le haut ou vers le bas à l'aide de la poignée située avant le nom de l'élément.
Afficher les lignes	Indiquez le nombre de lignes de valeurs d'élément à afficher.
Remplacer l'hôte	Sélectionnez un widget compatible ou le sélecteur d'hôte du tableau de bord comme source de données pour les hôtes. Ce paramètre n'est pas disponible lors de la configuration du widget sur un tableau de bord de modèle.
Configuration avancée	Cliquez sur l'intitulé *Configuration avancée* pour afficher les options de configuration avancée.

Configuration des colonnes

Pour configurer les colonnes des éléments, cliquez sur *Ajouter* dans le paramètre *Items* :



Paramètres communs des colonnes :

Nom	Saisissez le nom de la colonne. Si ce champ est laissé vide, le nom de l'élément du paramètre *Item* est utilisé.
Item	Sélectionnez l'élément. Notez que les paramètres de configuration de la colonne varient en fonction du type d'information de l'élément sélectionné ; pour plus d'informations, consultez les paramètres individuels ci-dessous. Lors de la configuration du widget sur un tableau de bord de modèle, seuls les éléments configurés sur le modèle peuvent être sélectionnés.
Couleur de base	Sélectionnez la couleur d'arrière-plan de la colonne ou la couleur de remplissage si *Display* est défini sur « Bar » ou « Indicators ». Notez que la couleur de base peut être remplacée par les couleurs de seuil ou de mise en évidence.

Paramètres de colonne spécifiques aux éléments de type numérique :

Affichage	Sélectionnez comment la valeur de l'élément doit être affichée : **Tel quel** - comme du texte normal ; **Barre** - sous forme de barre pleine remplie de couleur ; **Indicateurs** - sous forme de barre segmentée remplie de couleur.
Min	Saisissez la valeur minimale pour la barre/les indicateurs. Si ce champ est laissé vide, le widget utilisera la valeur minimale de l'élément. Ce paramètre n'est disponible que lorsque *Affichage* est défini sur « Barre » ou « Indicateurs ».
Max	Saisissez la valeur maximale pour la barre/les indicateurs. Si ce champ est laissé vide, le widget utilisera la valeur maximale de l'élément. Ce paramètre n'est disponible que lorsque *Affichage* est défini sur « Barre » ou « Indicateurs ».
Seuils	Cliquez sur *Ajouter* pour ajouter un seuil, sélectionnez une couleur de seuil dans le sélecteur de couleurs et indiquez une valeur numérique. La liste des seuils sera triée par ordre croissant lors de l'enregistrement. Les suffixes (par exemple, « 1d », « 2w », « 4K », « 8G ») sont pris en charge. Les correspondances de valeurs sont prises en charge.
Données d'historique	Sélectionnez si les données doivent être prises depuis l'historique ou les tendances : **Auto** - sélection automatique ; **Historique** - utiliser les données d'historique ; **Tendances** - utiliser les données de tendance.

Paramètres de colonne spécifiques aux éléments de type caractère, texte, journal et JSON :

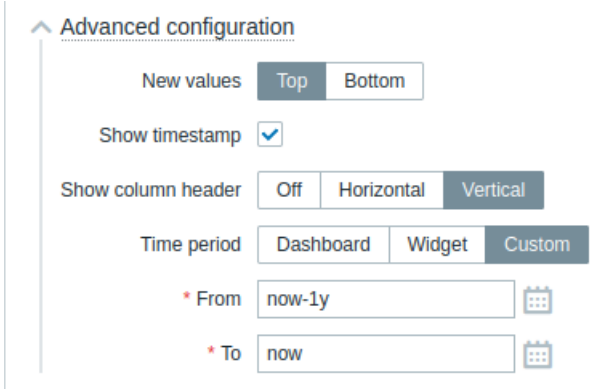
Surlignages	Cliquez sur *Ajouter* pour ajouter un surlignage, sélectionnez une couleur de surlignage dans le sélecteur de couleurs et spécifiez une expression régulière. La couleur sélectionnée sera utilisée comme couleur d'arrière-plan pour les valeurs d'élément lorsque l'expression régulière spécifiée correspond au texte.
Affichage	Sélectionnez la manière dont la valeur de l'élément doit être affichée : **Tel quel** - exactement comme reçue. Les retours à la ligne sont conservés si la valeur contient des sauts de ligne. Le retour à la ligne automatique est appliqué si *Disposition* est défini sur "Vertical" et si plus d'une colonne est configurée ; **HTML** - comme texte formaté en HTML ; **Ligne unique** - sur une seule ligne, tronquée à une longueur spécifiée (1-500 caractères). Le survol ou le clic sur la valeur tronquée ouvre une fenêtre contextuelle avec la valeur complète.
Utiliser une police à chasse fixe	Cochez cette case pour afficher la valeur de l'élément dans une police à chasse fixe (non cochée par défaut).
Afficher l'heure locale	Cochez cette case pour afficher l'heure locale au lieu de l'horodatage dans la colonne d'horodatage. Notez que la case *Afficher l'horodatage* dans la configuration avancée doit également être cochée. Ce paramètre est disponible uniquement pour les éléments de type journal.

Paramètres de colonne spécifiques aux éléments de type binaire :

Afficher la miniature	Cochez cette case pour afficher une miniature pour les binaires d'image ou une option « Afficher » pour les binaires non image. Décochez cette case pour afficher une option « Afficher » pour toutes les valeurs d'élément binaires. Le survol ou le clic sur l'option « Afficher » ouvre une fenêtre contextuelle contenant la valeur de l'élément (image ou chaîne Base64). Si la valeur de l'élément est une chaîne vide, l'option « Afficher » est affichée ; son survol ou son clic ouvre une fenêtre contextuelle contenant « Chaîne vide ».

Configuration avancée

Des options de configuration avancée sont disponibles dans la section repliable *Configuration avancée* :

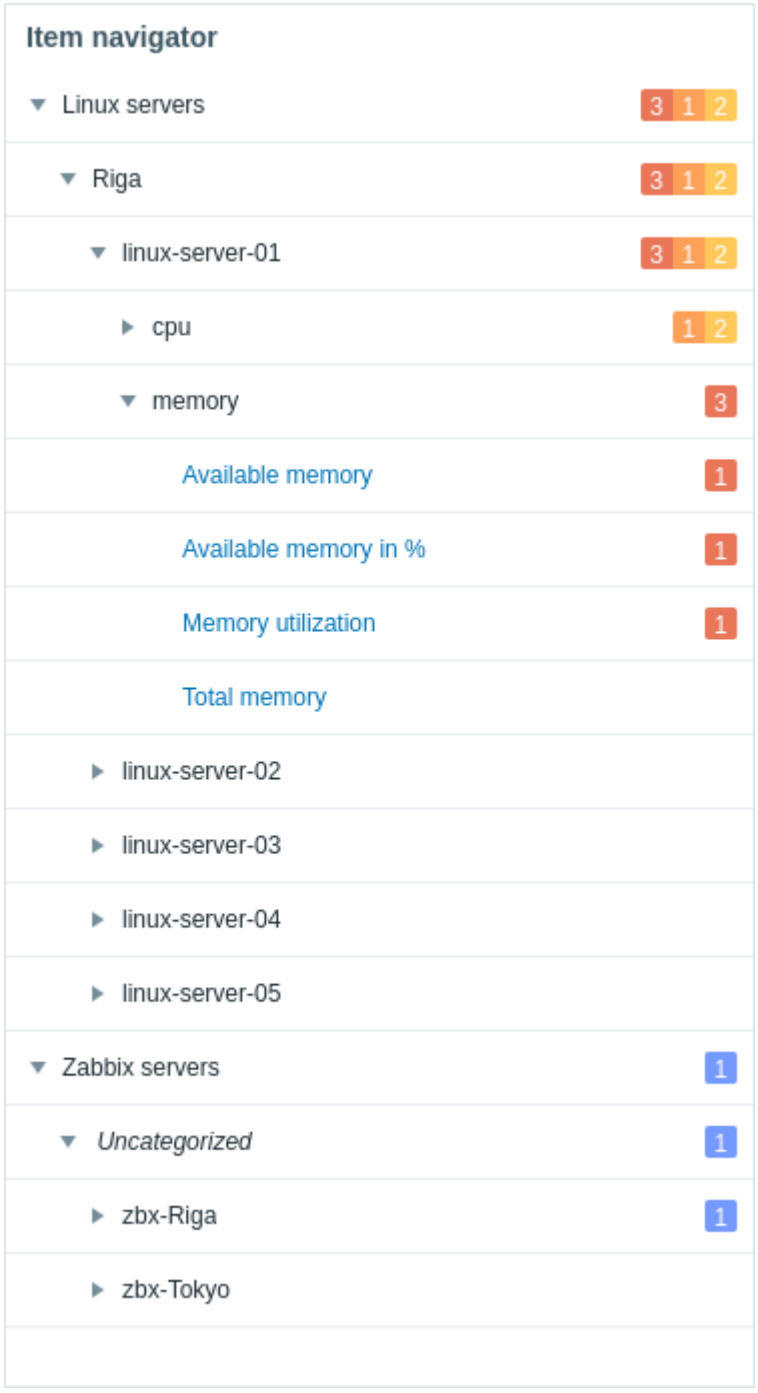


Nouvelles valeurs	Sélectionnez où les nouvelles valeurs d'élément doivent être ajoutées : **En haut** - en haut des colonnes ; **En bas** - en bas des colonnes.
Afficher l'horodatage	Cochez cette case pour afficher la colonne d'horodatage (décochée par défaut).
Afficher l'en-tête de colonne	Sélectionnez l'orientation de l'en-tête de colonne : **Désactivé** - masquer l'en-tête ; **Horizontal** - afficher l'en-tête horizontalement ; **Vertical** - afficher l'en-tête verticalement.
Période	Sélectionnez la source de données pour la période : **Tableau de bord** - utiliser le sélecteur de période du tableau de bord ; **Widget** - utiliser un widget compatible (défini dans le paramètre *Widget*) ; **Personnalisée** - utiliser une période personnalisée définie dans les paramètres *De* et *À* ; si elle est définie, une icône d'horloge sera affichée dans le coin supérieur droit du widget, indiquant l'heure définie au survol de la souris. Notez que, quelle que soit la configuration de *Période* du widget, les widgets compatibles peuvent toujours l'utiliser comme source de données pour la période.
Widget	Saisissez ou sélectionnez un widget compatible (*Graphique*, *Graphique (classique)*, *Prototype de graphique*) comme source de données pour la période. Ce paramètre est disponible si *Période* est défini sur "Widget".
De	Saisissez ou sélectionnez le début de la période. La syntaxe de temps relatif (`now`, `now/d`, `now/w-1w`, etc.) est prise en charge. Ce paramètre est disponible si *Période* est défini sur "Personnalisée".
À	Saisissez ou sélectionnez la fin de la période. La syntaxe de temps relatif (`now`, `now/d`, `now/w-1w`, etc.) est prise en charge. Ce paramètre est disponible si *Période* est défini sur "Personnalisée".

Navigateur d'éléments

Aperçu

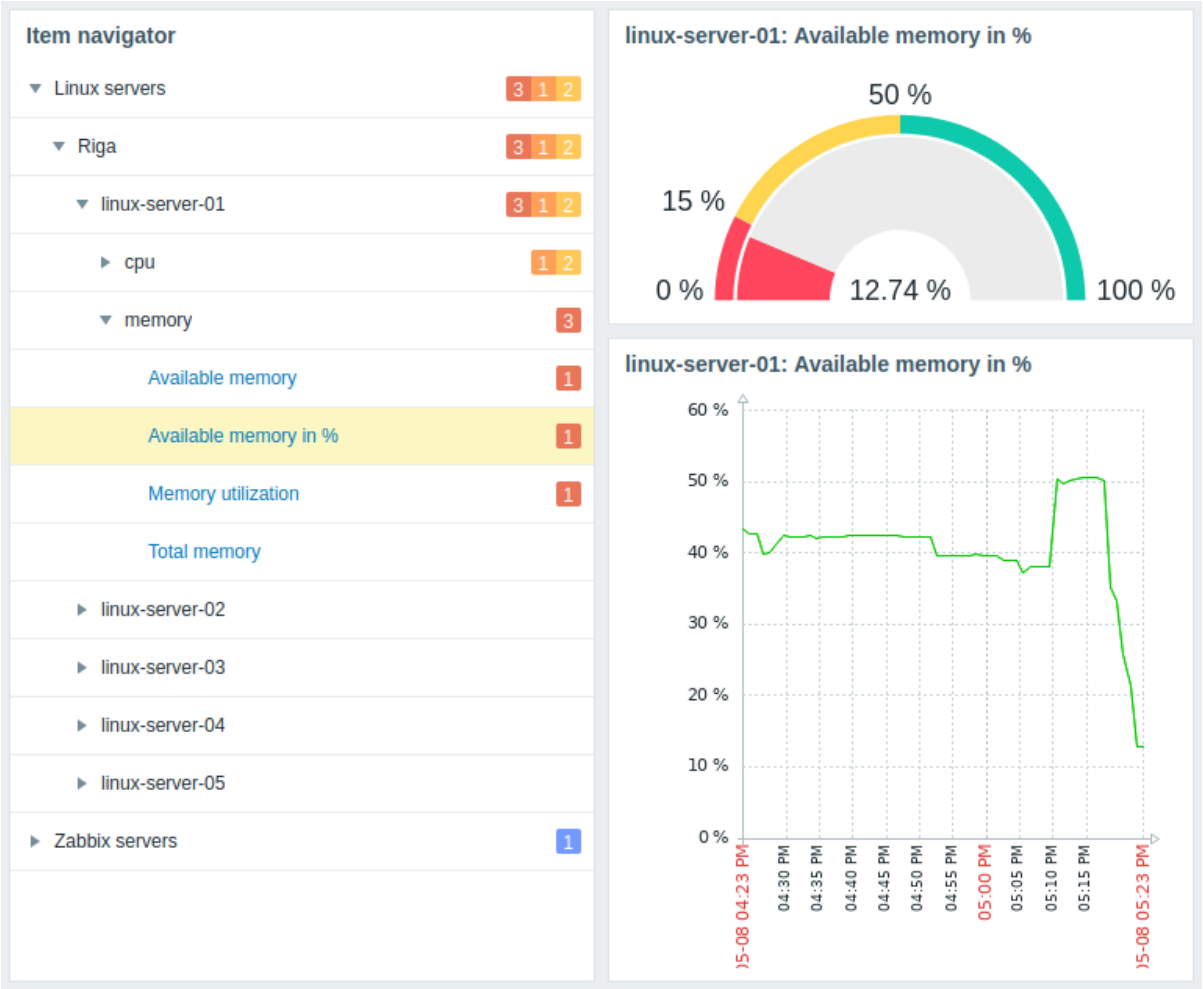
Le widget *Navigateur d'éléments* affiche une liste d'éléments selon diverses options de filtrage et de regroupement.



Les groupes selon lesquels les éléments sont organisés peuvent être développés ou réduits.

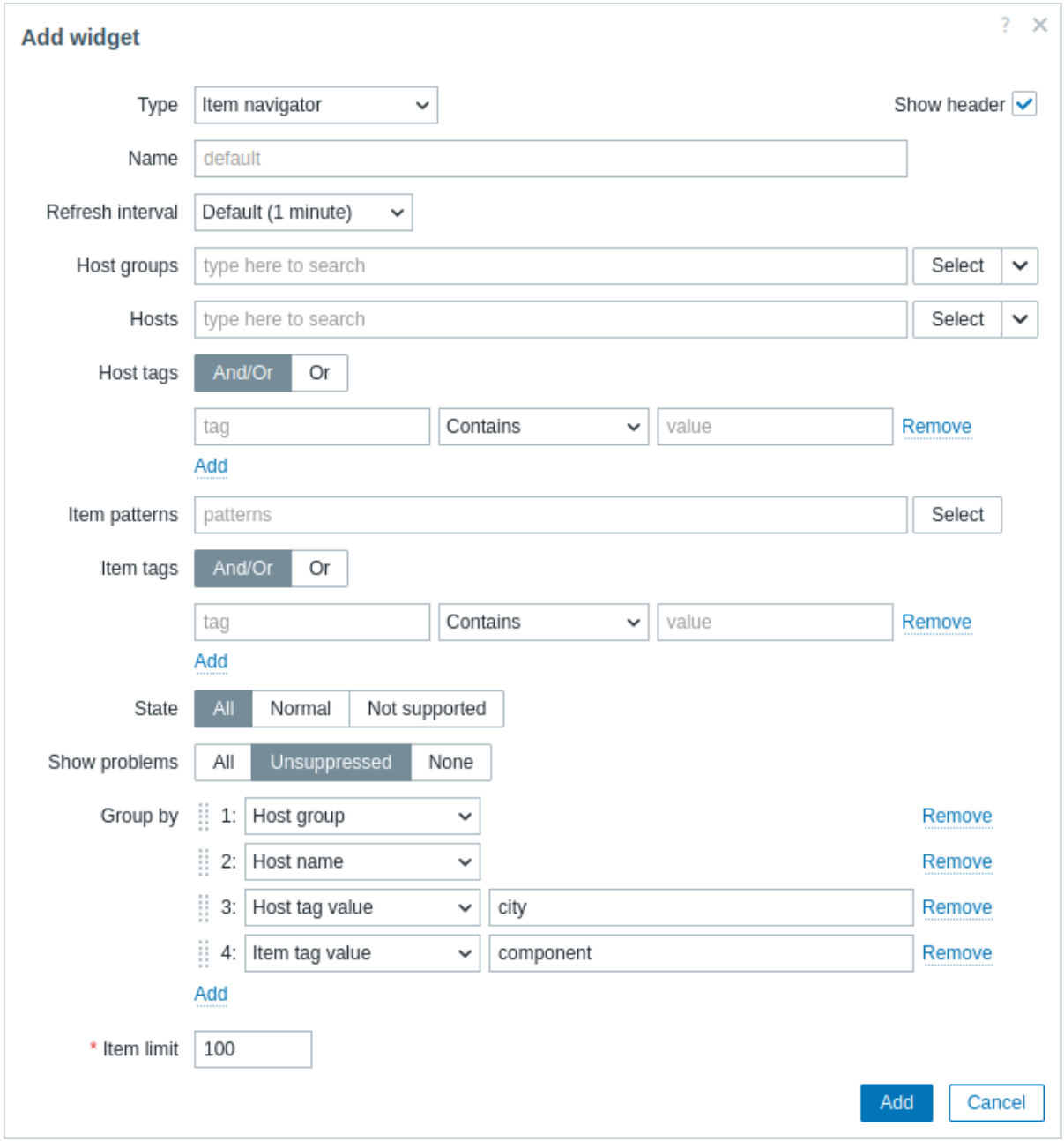
Pour les groupes et les problèmes, des détails supplémentaires sont accessibles via des infobulles au survol de la souris.

Le widget est particulièrement utile pour contrôler ce que les autres widgets affichent en fonction de l'élément sélectionné.



Configuration

Pour configurer le widget, sélectionnez *Navigateur d'éléments* comme type :



En plus des paramètres communs à tous les widgets, vous pouvez définir les options spécifiques suivantes :

Groupes de hôtes	Sélectionnez les groupes de hôtes. Vous pouvez également sélectionner un widget compatible comme source de données pour les groupes de hôtes. Ce champ propose la saisie semi-automatique : commencez à saisir le nom d'un groupe pour afficher une liste déroulante des groupes correspondants. La sélection d'un groupe de hôtes parent sélectionne implicitement tous les groupes de hôtes imbriqués ; si aucun groupe de hôtes n'est sélectionné, le widget affiche les éléments appartenant à tous les hôtes de tous les groupes de hôtes. Ce paramètre n'est pas disponible lors de la configuration du widget sur un tableau de bord de modèle.
Hôtes	Sélectionnez les hôtes. Vous pouvez également sélectionner un widget compatible ou le tableau de bord comme source de données pour les hôtes. Ce champ propose la saisie semi-automatique : commencez à saisir le nom d'un hôte pour afficher une liste déroulante des hôtes correspondants. Si aucun hôte n'est sélectionné, le widget affiche les éléments appartenant à tous les hôtes. Ce paramètre n'est pas disponible lors de la configuration du widget sur un tableau de bord de modèle.

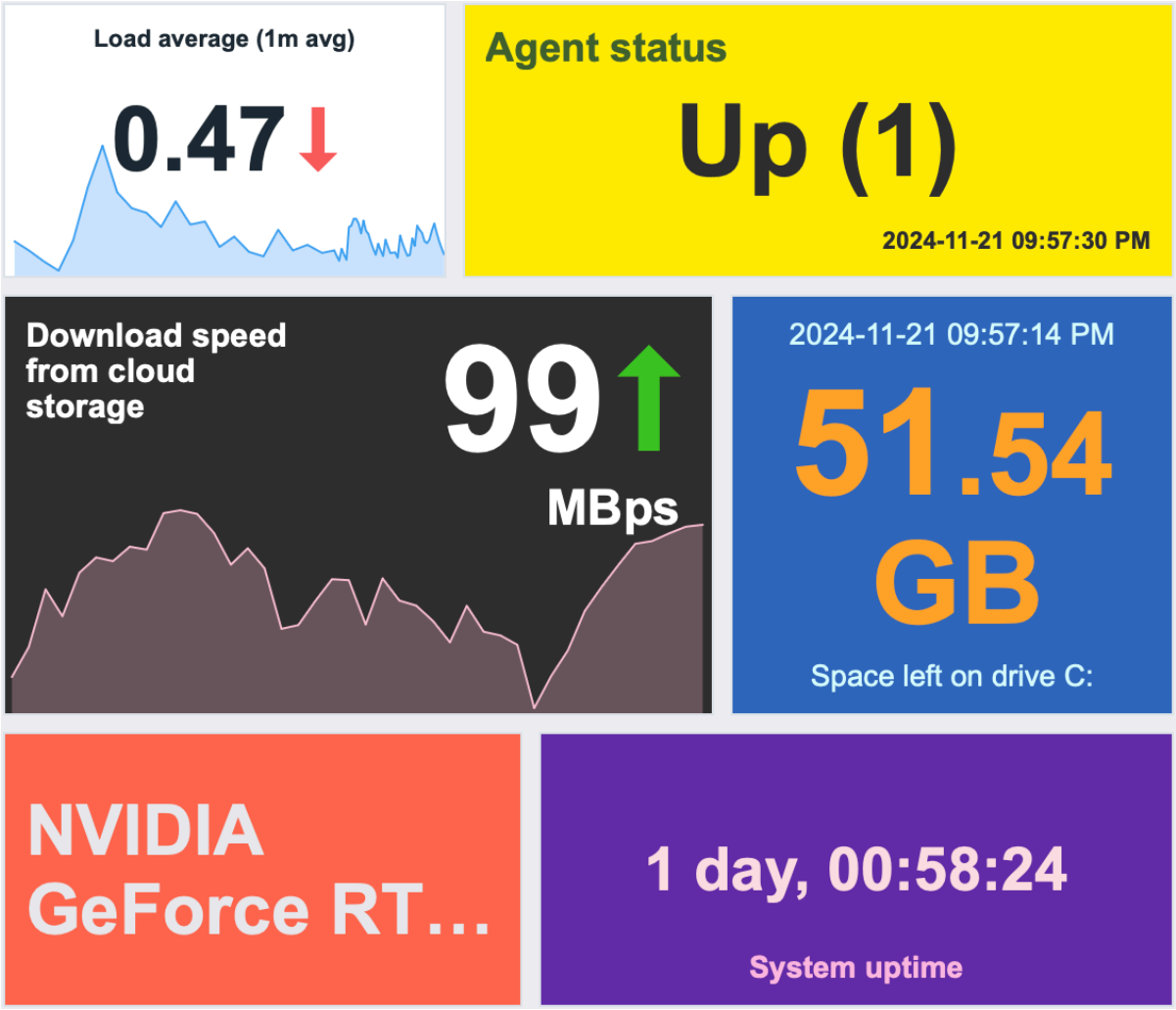
Étiquettes d'hôte	Spécifiez les étiquettes d'hôte pour filtrer les éléments affichés dans le widget. Il est possible d'inclure ou d'exclure des étiquettes et des valeurs d'étiquette spécifiques. Plusieurs conditions peuvent être définies. La correspondance des noms d'étiquette est toujours sensible à la casse. Plusieurs opérateurs sont disponibles pour chaque condition : **Existe** - inclure les noms d'étiquette spécifiés ; **Est égal à** - inclure les noms et les valeurs d'étiquette spécifiés (sensible à la casse) ; **Contient** - inclure les noms d'étiquette spécifiés lorsque les valeurs d'étiquette contiennent la chaîne saisie (correspondance de sous-chaîne, insensible à la casse) ; **N'existe pas** - exclure les noms d'étiquette spécifiés ; **N'est pas égal à** - exclure les noms et les valeurs d'étiquette spécifiés (sensible à la casse) ; **Ne contient pas** - exclure les noms d'étiquette spécifiés lorsque les valeurs d'étiquette contiennent la chaîne saisie (correspondance de sous-chaîne, insensible à la casse). Il existe deux types de calcul pour les conditions : **Et/Ou** - toutes les conditions doivent être remplies ; les conditions ayant le même nom d'étiquette sont regroupées avec la condition Ou ; **Ou** - une seule condition remplie suffit. Ce paramètre n'est pas disponible lors de la configuration du widget sur un tableau de bord de modèle.
Motifs d'élément	Saisissez des motifs d'élément ou sélectionnez des éléments existants comme motifs d'élément. Les éléments correspondant aux motifs spécifiés seront affichés dans le navigateur d'éléments. Des motifs génériques peuvent être utilisés pour la sélection (par exemple, `*` renvoie les éléments correspondant à zéro caractère ou plus ; `Zabbix*` renvoie les éléments commençant par « Zabbix »). Pour spécifier un motif générique, saisissez la chaîne manuellement et appuyez sur *Entrée*. Lorsque vous commencez à saisir du texte, une liste déroulante affiche les éléments correspondants, limitée à ceux appartenant aux *Hôtes* sélectionnés ou aux hôtes des *Groupes de hôtes* sélectionnés, le cas échéant. Le symbole générique est toujours interprété ; il n'est donc pas possible d'ajouter individuellement, par exemple, un élément nommé *item**, s'il existe d'autres éléments correspondants (par exemple, item2, item3). Lors de la configuration du widget sur un tableau de bord de modèle, ce paramètre permet de sélectionner uniquement les éléments configurés sur le modèle.
Étiquettes d'élément	Spécifiez les étiquettes d'élément pour filtrer les éléments affichés dans le widget. Il est possible d'inclure ou d'exclure des étiquettes et des valeurs d'étiquette spécifiques. Plusieurs conditions peuvent être définies. La correspondance des noms d'étiquette est toujours sensible à la casse. Plusieurs opérateurs sont disponibles pour chaque condition : **Existe** - inclure les noms d'étiquette spécifiés ; **Est égal à** - inclure les noms et les valeurs d'étiquette spécifiés (sensible à la casse) ; **Contient** - inclure les noms d'étiquette spécifiés lorsque les valeurs d'étiquette contiennent la chaîne saisie (correspondance de sous-chaîne, insensible à la casse) ; **N'existe pas** - exclure les noms d'étiquette spécifiés ; **N'est pas égal à** - exclure les noms et les valeurs d'étiquette spécifiés (sensible à la casse) ; **Ne contient pas** - exclure les noms d'étiquette spécifiés lorsque les valeurs d'étiquette contiennent la chaîne saisie (correspondance de sous-chaîne, insensible à la casse). Il existe deux types de calcul pour les conditions : **Et/Ou** - toutes les conditions doivent être remplies ; les conditions ayant le même nom d'étiquette sont regroupées avec la condition Ou ; **Ou** - une seule condition remplie suffit.
État	Filtrez les éléments à afficher en fonction de leur état (tous, normal, non pris en charge).
Afficher les problèmes	Filtrez les problèmes à afficher en fonction de leur statut (tous, non supprimés, aucun). Les problèmes sont indiqués par des blocs colorés à la fin de la ligne de l'élément. Les couleurs sont basées sur la gravité du problème, qui peut être personnalisée dans *Administration* > *Général* > *Options d'affichage des déclencheurs*. La gravité des problèmes peut être modifiée lors de la mise à jour des problèmes.

Regrouper par	Ajoutez un attribut de regroupement selon lequel regrouper les éléments : **Groupe d'hôtes** - regrouper les éléments selon les groupes d'hôtes de leurs hôtes ; **Nom d'hôte** - regrouper les éléments selon leurs hôtes ; **Valeur d'étiquette d'hôte** - saisissez un nom d'étiquette pour regrouper les éléments selon les valeurs de cette étiquette d'hôte (par exemple, saisissez « city » pour regrouper les éléments selon les valeurs « Riga », « Tokyo », etc.) ; **Valeur d'étiquette d'élément** - saisissez un nom d'étiquette pour regrouper les éléments selon les valeurs de cette étiquette d'élément (par exemple, saisissez « component » pour regrouper les éléments selon les valeurs « cpu », « memory », etc.). Les attributs de regroupement peuvent être réordonnés en les faisant glisser vers le haut ou vers le bas à l'aide de la poignée située avant le nom du groupe. Notez que l'ordre des attributs de regroupement détermine l'ordre d'imbrication des groupes. Par exemple, la spécification de plusieurs noms d'étiquette d'hôte (1 : color, 2 : city) regroupera les éléments par couleur (red, blue, etc.), puis par ville (Riga, Tokyo, etc.). Un élément peut être affiché dans plusieurs groupes selon les attributs de regroupement configurés (par exemple, lors d'un regroupement par groupe d'hôtes, lorsque l'hôte de l'élément appartient à plusieurs groupes d'hôtes). Cliquer sur ces éléments les sélectionne et les met en évidence dans tous les groupes. Les éléments qui ne correspondent pas aux attributs de regroupement configurés sont affichés dans le groupe *Non catégorisé*. Si *Afficher les problèmes* est configuré pour afficher les problèmes, ceux-ci sont affichés comme suit : - pour chaque groupe, le nombre total de problèmes des éléments subordonnés est affiché ; - pour chaque élément, seul son nombre de problèmes est affiché. Jusqu'à 10 attributs de regroupement peuvent être spécifiés, et tous doivent être uniques. Si aucun attribut de regroupement n'est spécifié, les éléments ne seront pas regroupés.
Limite d'éléments	Saisissez le nombre maximal d'éléments à afficher. Les valeurs possibles vont de 1 à 9999. Lorsque le nombre d'éléments disponibles à afficher est supérieur à la limite définie, un message correspondant est affiché sous les éléments affichés (par exemple, « 100 éléments sur plus de 100 sont affichés »). Notez que la limite d'éléments configurée affecte également l'affichage des groupes configurés ; par exemple, si la limite d'éléments est définie sur 100 et que les éléments sont regroupés selon leurs hôtes (chacun contenant 200 éléments), seul le premier hôte avec ses 100 éléments sera affiché dans le widget. Ce paramètre n'est pas affecté par le paramètre *Limite pour les résultats de recherche et de filtrage* dans *Administration > Général > Interface graphique*.

Valeur de l'élément

Aperçu

Le widget *Valeur d'élément* affiche la valeur d'un seul élément numérique ou de chaîne. Il est utile pour surveiller des métriques importantes, visualiser des seuils et détecter des changements soudains dans les données.



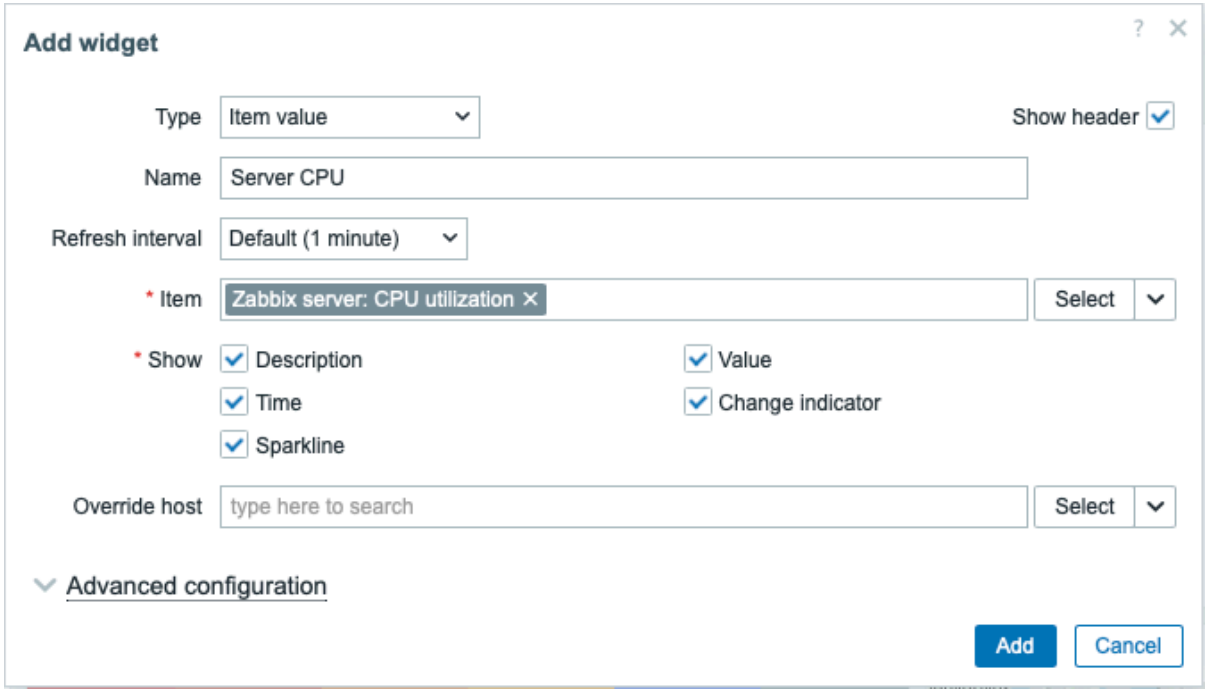
Vous pouvez configurer le widget pour afficher les éléments suivants :

- Horodatage de la métrique (2024-11-21 09:57:30)
- Description de l'élément (Charge moyenne (1m), état de l'agent)
- Valeur de l'élément, indicateur de variation et unités (0.47 ↓, 99 ↑ Mbps)
- Couleur d'arrière-plan
- Graphique sparkline pour les valeurs de la période de temps spécifiée

En cliquant sur le widget, vous ouvrez un graphique ad hoc pour les éléments numériques ou les dernières données pour les éléments de chaîne.

Configuration

Pour configurer, sélectionnez *Valeur de l'élément* comme type de widget :



En plus des paramètres communs à tous les widgets, vous pouvez définir les options spécifiques suivantes :

Élément	Sélectionnez l'élément. Vous pouvez également sélectionner un widget compatible comme source de données pour l'élément. Les éléments renvoyant des données binaires ne sont pas pris en charge.
Afficher	Cochez la case pour afficher l'élément correspondant : description, valeur, heure, indicateur de changement, sparkline. Décochez-la pour le masquer. Au moins un élément doit être sélectionné.
Remplacer l'hôte	Sélectionnez un widget compatible ou le sélecteur d'hôte du tableau de bord comme source de données pour les hôtes. Ce paramètre n'est pas disponible lors de la configuration du widget sur un tableau de bord de modèle.
Configuration avancée	Cliquez sur l'en-tête *Configuration avancée* pour afficher les options de configuration avancée.

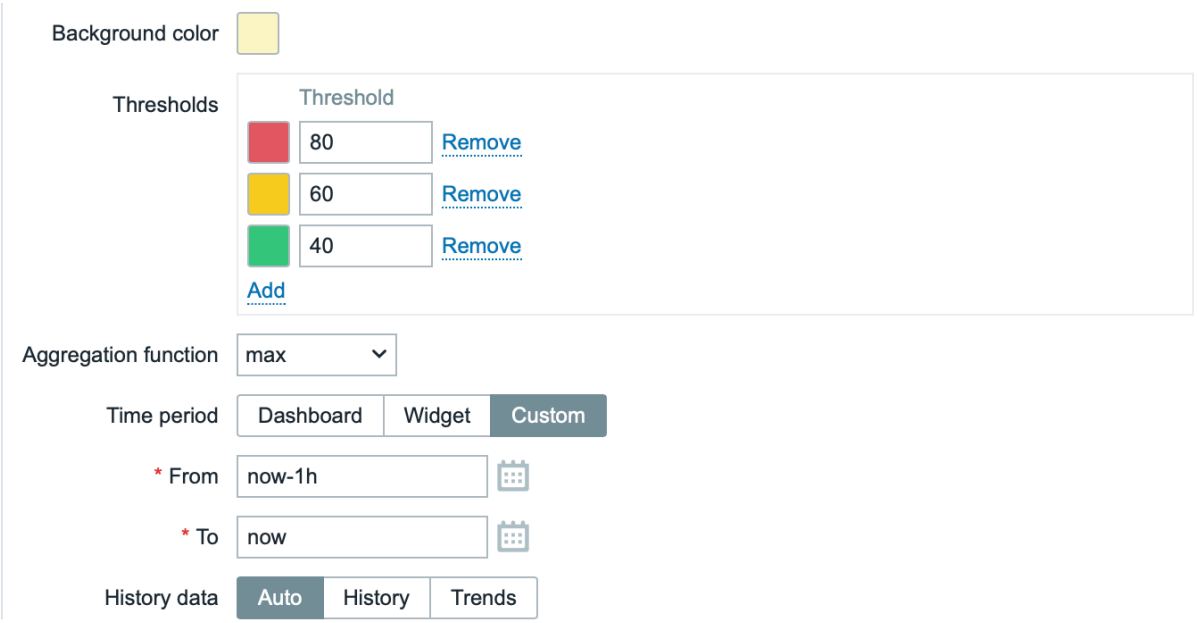
Configuration avancée

Les options de configuration avancée sont disponibles dans la section repliable *Configuration avancée*, et uniquement pour les éléments sélectionnés dans le champ *Afficher* (voir ci-dessus).

Notez que plusieurs éléments ne peuvent pas occuper le même espace ; s'ils sont placés dans le même espace, un message d'erreur s'affichera.

Paramètres génériques du widget

Ces paramètres déterminent la couleur d'arrière-plan (statique ou dynamique) de l'ensemble du widget ainsi qu'une fonction d'agrégation pour l'affichage des valeurs.

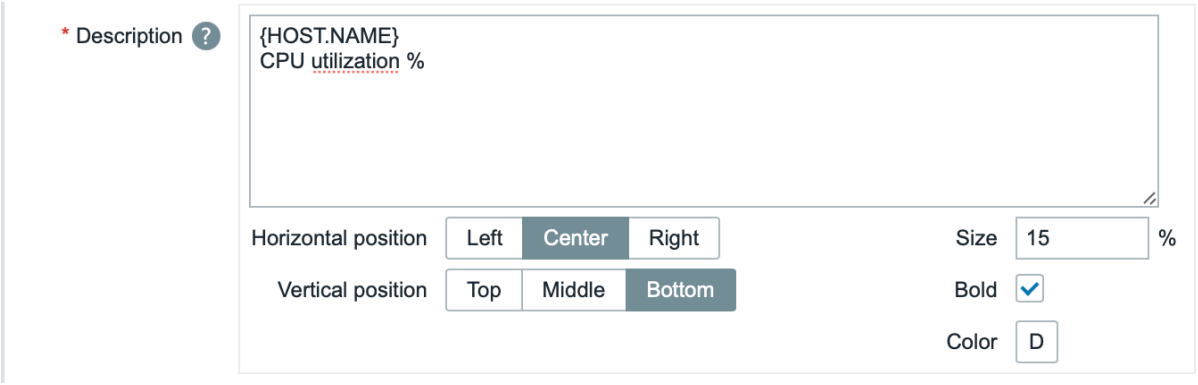


Couleur d'arrière-plan	Sélectionnez la couleur d'arrière-plan de l'ensemble du widget dans le sélecteur de couleurs. D correspond à la couleur par défaut (qui dépend du thème de l'interface). Pour revenir à la valeur par défaut, cliquez sur le bouton *Utiliser la valeur par défaut* dans le sélecteur de couleurs.
Seuils	Configurez la couleur d'arrière-plan dynamique de l'ensemble du widget. Cliquez sur *Ajouter* pour ajouter un seuil, sélectionnez la couleur d'arrière-plan dans le sélecteur de couleurs et indiquez une valeur numérique. Lorsque la valeur de l'élément est égale ou supérieure à la valeur du seuil, la couleur d'arrière-plan change. La liste sera triée par ordre croissant lors de l'enregistrement. Notez que la couleur d'arrière-plan dynamique s'affichera correctement uniquement pour les éléments numériques.
Fonction d'agrégation	Indiquez la fonction d'agrégation à utiliser dans la *Période* sélectionnée : **min** - afficher la plus petite valeur ; **max** - afficher la plus grande valeur ; **avg** - afficher la valeur moyenne ; **count** - afficher le nombre de valeurs ; **sum** - afficher la somme des valeurs ; **first** - afficher la première valeur ; **last** - afficher la dernière valeur. Si **non utilisée** est sélectionné, la valeur la plus récente est affichée (sans agrégation). L'agrégation permet d'afficher une valeur agrégée pour l'intervalle indiqué (5 minutes, une heure, un jour), au lieu de la valeur la plus récente. Seules les données numériques peuvent être affichées pour *min*, *max*, *avg* et *sum*. Pour *count*, les données non numériques seront converties en données numériques.
Période	Période à utiliser pour agréger les valeurs. Sélectionnez la source de données de la période : **Tableau de bord** - utiliser le sélecteur de période du tableau de bord ; **Widget** - utiliser un widget compatible (défini dans le paramètre *Widget*) ; **Personnalisée** - utiliser une période personnalisée définie dans les paramètres *Du* et *Au* ; si elle est définie, une icône d'horloge s'affichera dans le coin supérieur droit du widget et indiquera l'heure définie au survol de la souris. Notez que, quelle que soit la configuration de la *Période* du widget, les widgets compatibles peuvent toujours l'utiliser comme source de données pour la période. Ce paramètre est masqué lorsque la *Fonction d'agrégation* est définie sur « non utilisée ».
Widget	Saisissez ou sélectionnez un widget compatible comme source de données pour la période. Ce paramètre est disponible lorsque *Période* est définie sur « Widget ».
Du	Saisissez ou sélectionnez le début de la période. La syntaxe des périodes relatives (`now`, `now/d`, `now/w-1w`, etc.) est prise en charge. Ce paramètre est disponible lorsque *Période* est définie sur « Personnalisée ».
Au	Saisissez ou sélectionnez la fin de la période. La syntaxe des périodes relatives (`now`, `now/d`, `now/w-1w`, etc.) est prise en charge. Ce paramètre est disponible lorsque *Période* est définie sur « Personnalisée ».

Données d'historique	Prenez les données dans l'historique ou les tendances : **Automatique** - sélection automatique ; **Historique** - prendre les données d'historique ; **Tendances** - prendre les données de tendance. Ce paramètre s'applique uniquement aux données numériques. Les données non numériques seront toujours prises dans l'historique.

Description

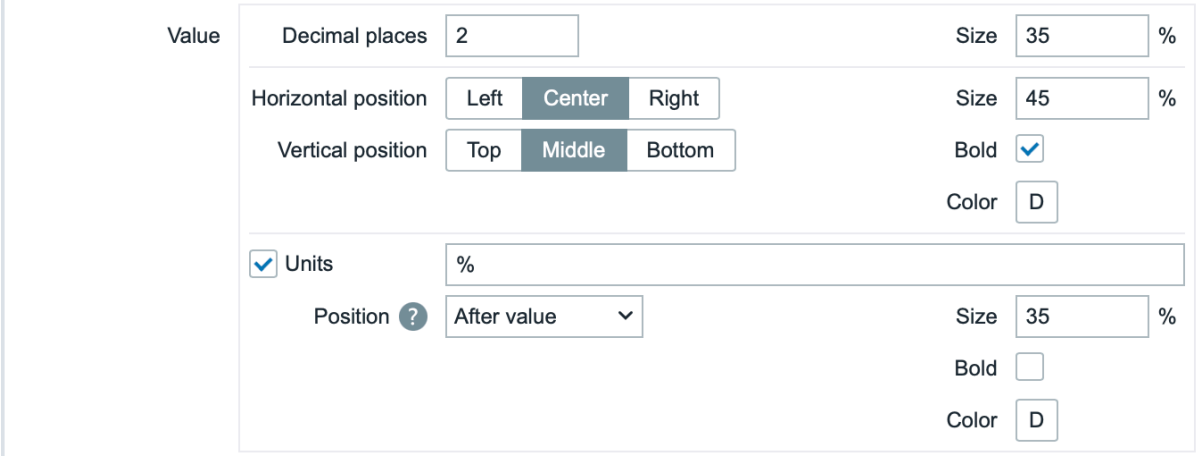
Ces paramètres déterminent comment la description de l'élément doit être affichée.



Description	Saisissez la description de l'élément. Cette description peut remplacer le nom d'élément par défaut. Les descriptions sur plusieurs lignes sont prises en charge. Une combinaison de texte et de macros prises en charge est possible. {HOST.*}, {ITEM.*}, {INVENTORY.*} et les macros utilisateur sont prises en charge.
Position horizontale	Sélectionnez la position horizontale de la description de l'élément : gauche, droite ou centre.
Position verticale	Sélectionnez la position verticale de la description de l'élément : haut, bas ou milieu.
Taille	Saisissez la hauteur de la taille de police pour la description de l'élément (en pourcentage par rapport à la hauteur totale du widget).
Gras	Cochez la case pour afficher la description de l'élément en gras.
Couleur	Sélectionnez la couleur de la description de l'élément à l'aide du sélecteur de couleurs. D correspond à la couleur par défaut (dépend du thème de l'interface web). Pour revenir à la valeur par défaut, cliquez sur le bouton *Use default* dans le sélecteur de couleurs.

Valeur

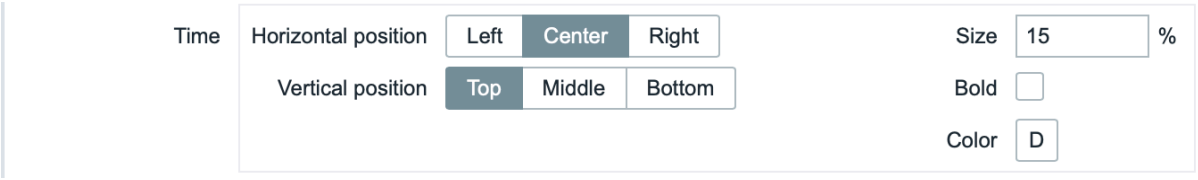
Ces paramètres déterminent comment la valeur de l'élément doit être affichée.



Décimales	Sélectionnez le nombre de décimales à afficher avec la valeur. Cette valeur n'affecte que les éléments de type flottant. Pour les éléments dont l'unité est définie sur « s », si cette valeur est supérieure à 0 (la valeur par défaut est 2), le widget arrondit l'unité de temps la plus significative et affiche une valeur numérique (par exemple, « 10.43m ») au lieu de convertir la valeur en une chaîne de temps complète. Lorsqu'elle est définie sur 0, les secondes sont converties en une chaîne de temps lisible par l'homme (par exemple, « 4h 56m 30s »).
Taille	Saisissez la hauteur de la taille de police pour les décimales (en pourcentage par rapport à la hauteur totale du widget).
Position horizontale	Sélectionnez la position horizontale de la valeur de l'élément : gauche, droite ou centre.
Position verticale	Sélectionnez la position verticale de la valeur de l'élément : haut, bas ou milieu.
Taille	Saisissez la hauteur de la taille de police pour la valeur de l'élément (en pourcentage par rapport à la hauteur totale du widget). Notez que la taille de la valeur de l'élément est prioritaire ; les autres éléments doivent lui céder de l'espace. Toutefois, avec l'indicateur de changement, si la valeur est trop grande, elle sera tronquée afin d'afficher l'indicateur de changement.
Gras	Cochez la case pour afficher la valeur de l'élément en gras.
Couleur	Sélectionnez la couleur de la valeur de l'élément dans le sélecteur de couleurs. D signifie couleur par défaut (dépend du thème de l'interface web). Pour revenir à la valeur par défaut, cliquez sur le bouton *Utiliser la valeur par défaut* dans le sélecteur de couleurs.
Unités	
Unités	Cochez la case pour afficher les unités avec la valeur de l'élément. Si vous saisissez un nom d'unité, il remplacera l'unité définie dans la configuration de l'élément.
Position	Sélectionnez la position de l'unité de l'élément : au-dessus, en dessous, avant ou après la valeur.
Taille	Saisissez la hauteur de la taille de police pour l'unité de l'élément (en pourcentage par rapport à la hauteur totale du widget).
Gras	Cochez la case pour afficher l'unité de l'élément en gras.
Couleur	Sélectionnez la couleur de l'unité de l'élément dans le sélecteur de couleurs. D signifie couleur par défaut (dépend du thème de l'interface web). Pour revenir à la valeur par défaut, cliquez sur le bouton *Utiliser la valeur par défaut* dans le sélecteur de couleurs.

Heure

Ces paramètres déterminent comment l'heure (valeur d'horloge provenant de l'historique de l'élément) doit être affichée.



Position horizontale	Sélectionnez la position horizontale de l'heure : gauche, droite ou centre.
Position verticale	Sélectionnez la position verticale de l'heure : haut, bas ou milieu.
Taille	Saisissez la hauteur de la taille de police pour l'heure (en pourcentage par rapport à la hauteur totale du widget).
Gras	Cochez la case pour afficher l'heure en gras.
Couleur	Sélectionnez la couleur de l'heure à l'aide du sélecteur de couleurs. D correspond à la couleur par défaut (dépend du thème de l'interface). Pour revenir à la valeur par défaut, cliquez sur le bouton *Use default* dans le sélecteur de couleurs.

Indicateur de changement

Cette section vous permet de sélectionner la couleur des indicateurs de changement à l'aide du sélecteur de couleurs.



Les indicateurs de changement sont les suivants :

- ↑ - la valeur de l'élément a augmenté (pour les éléments numériques)
- ↓ - la valeur de l'élément a diminué (pour les éléments numériques)
- ↕ - la valeur de l'élément a changé (pour les éléments de type chaîne et les éléments avec une correspondance de valeurs)

Remarque :

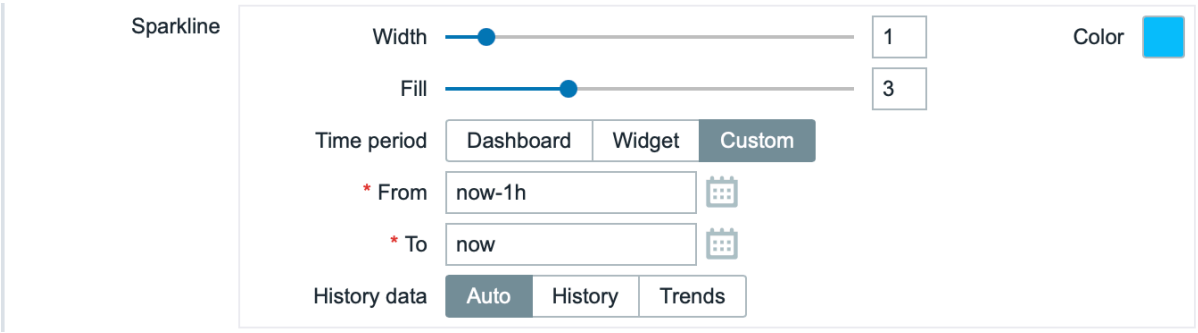
- Les indicateurs d'augmentation (↑) et de diminution (↓) ne sont pas affichés s'il n'y a qu'une seule valeur.
- Les valeurs sont toujours comparées dans le même périmètre ; par exemple :
  - La dernière valeur est comparée à la valeur précédente.
  - Une valeur mensuelle est comparée au mois précédent.
  - Pour les agrégations, la période précédente a la même durée que la période sélectionnée et se termine immédiatement avant le début de la période sélectionnée.

D représente la couleur par défaut (qui dépend du thème de l'interface). Pour rétablir la valeur par défaut, cliquez sur le bouton *Utiliser la valeur par défaut* dans le sélecteur de couleurs.

La taille verticale de l'indicateur de changement est égale à la taille de la valeur (la partie entière de la valeur pour les éléments numériques).

Mini-graphique

Ces paramètres déterminent comment le mini-graphique doit être affiché.

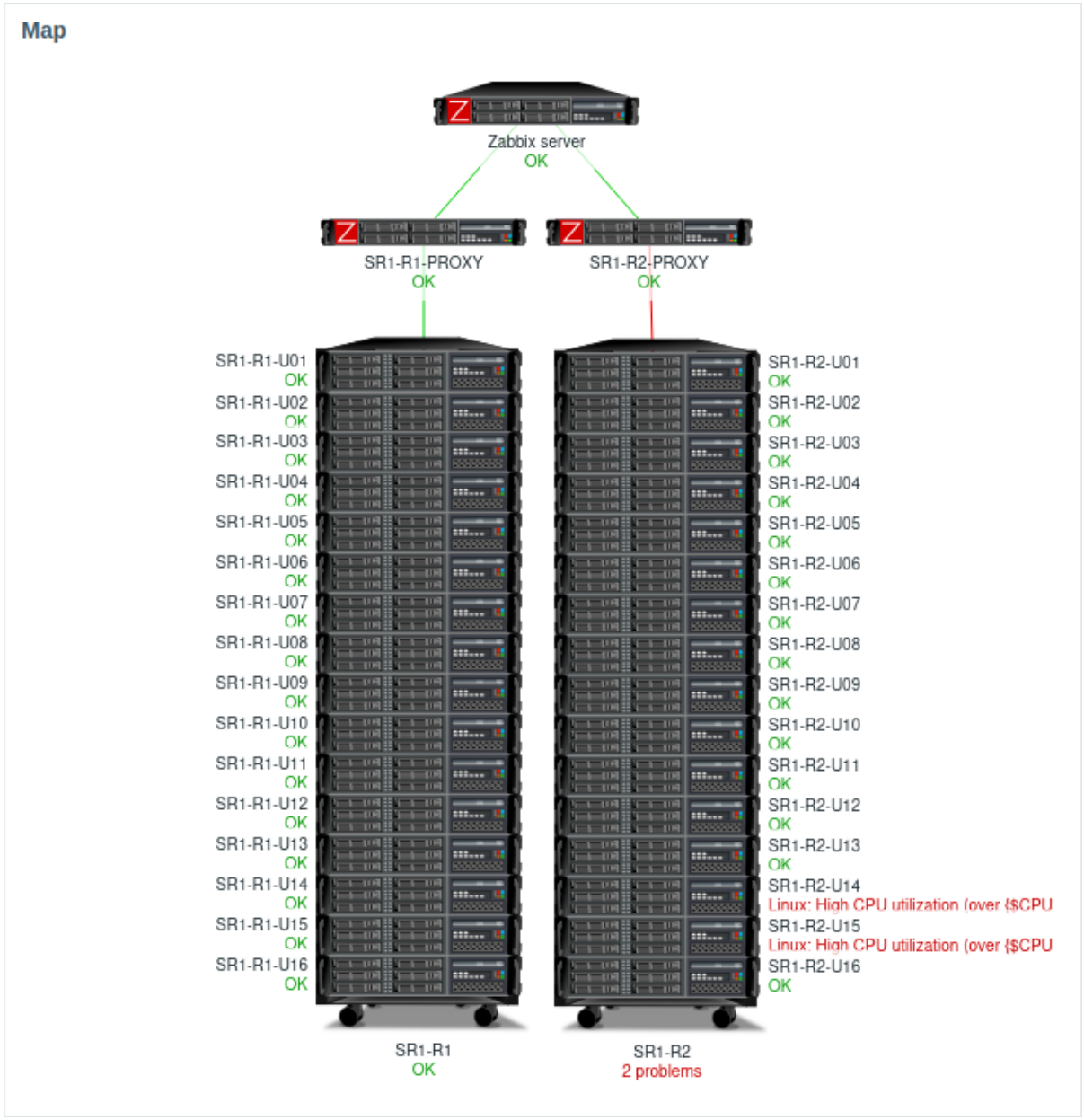


Largeur	Définissez l'épaisseur de la ligne du graphique à l'aide du curseur ou en saisissant manuellement une valeur comprise entre 0 et 10.
Couleur	Sélectionnez la couleur de la ligne et du remplissage.
Remplissage	Définissez le niveau de transparence de la couleur de remplissage à l'aide du curseur ou en saisissant manuellement une valeur comprise entre 0 et 10.
Période	Spécifiez la période pour les valeurs à inclure dans le mini-graphique. Sélectionnez la source de données pour la période : **Tableau de bord** - utiliser le sélecteur de période du tableau de bord ; **Widget** - utiliser un widget compatible (défini dans le paramètre *Widget*) ; **Personnalisée** - utiliser une période personnalisée définie dans les paramètres *De* et *À* ; si elle est définie, une icône d'horloge sera affichée dans le coin supérieur droit du widget, indiquant l'heure définie au survol de la souris. Notez que, quelle que soit la configuration de *Période* du widget, les widgets compatibles peuvent toujours l'utiliser comme source de données pour la période. Ce paramètre n'est pas disponible si *Fonction d'agrégation* est définie sur « non utilisée ».
Widget	Saisissez ou sélectionnez un widget compatible comme source de données pour la période. Ce paramètre est disponible si *Période* est définie sur « Widget ».
De	Saisissez ou sélectionnez le début de la période. La syntaxe de temps relatif (`now`, `now/d`, `now/w-1w`, etc.) est prise en charge. Ce paramètre est disponible si *Période* est définie sur « Personnalisée ».
À	Saisissez ou sélectionnez la fin de la période. La syntaxe de temps relatif (`now`, `now/d`, `now/w-1w`, etc.) est prise en charge. Ce paramètre est disponible si *Période* est définie sur « Personnalisée ».
Données d'historique	Utilisez les données de l'historique ou des tendances : **Auto** - sélection automatique ; **Historique** - utiliser les données d'historique ; **Tendances** - utiliser les données de tendance.

Carte

Vue d'ensemble

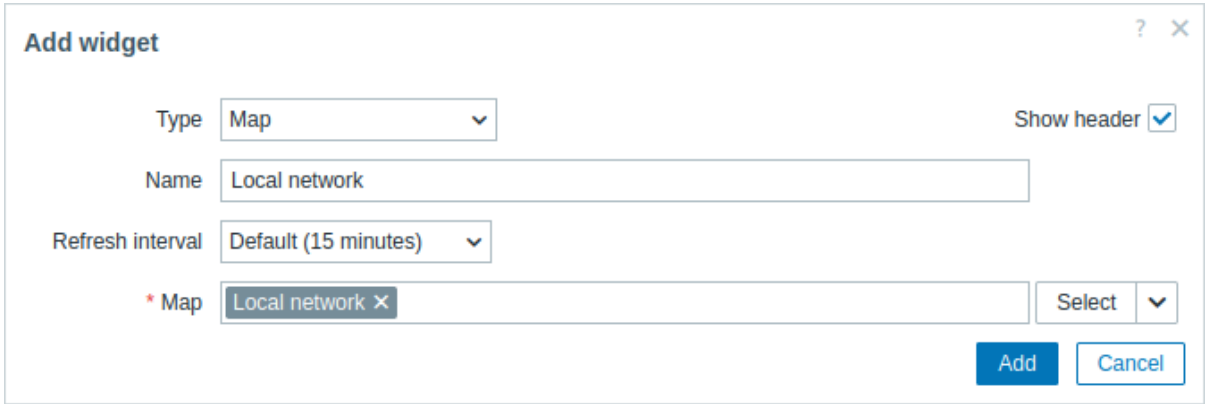
Le widget *Map* peut afficher une carte du réseau ou une visualisation similaire, vous offrant une vue d'ensemble dynamique de votre réseau.



Vous pouvez l'associer au widget Map navigation tree pour afficher dynamiquement la carte sélectionnée dans l'arborescence.

Configuration

Pour configurer, sélectionnez *Map* comme type :



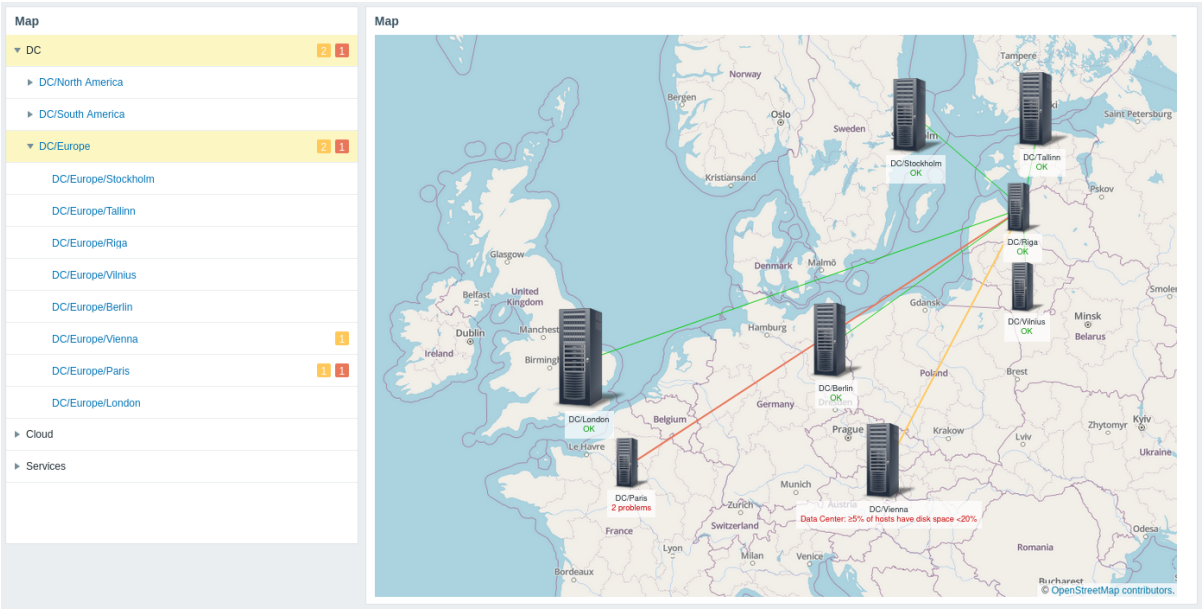
En plus des paramètres communs à tous les widgets, vous pouvez définir les options spécifiques suivantes :

Map	Définissez une carte à afficher. Vous pouvez également sélectionner un widget compatible comme source de données pour la carte à afficher. Ce champ prend en charge l'autocomplétion ; ainsi, lorsque vous commencez à saisir le nom de la carte ou du widget, une liste déroulante des cartes ou widgets correspondants est proposée.

Arborescence de navigation de la carte

Vue d'ensemble

Le widget *Arborescence de navigation des cartes* affiche une hiérarchie des cartes existantes ainsi que le nombre de problèmes pour chaque carte et groupe de cartes.

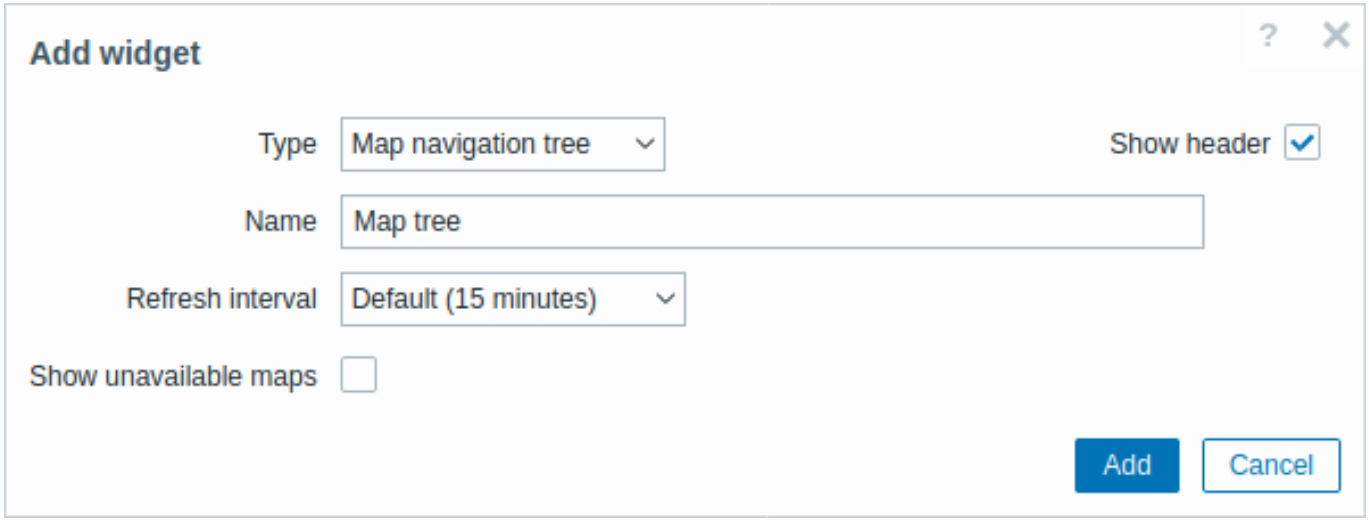


Vous pouvez l'associer au widget Map pour afficher dynamiquement la carte sélectionnée dans l'arborescence.

Le nombre de problèmes pour la carte de niveau supérieur dans la hiérarchie inclut tous les problèmes des sous-cartes ainsi que les siens.

Configuration

Pour configurer, sélectionnez *Map navigation tree* comme type :

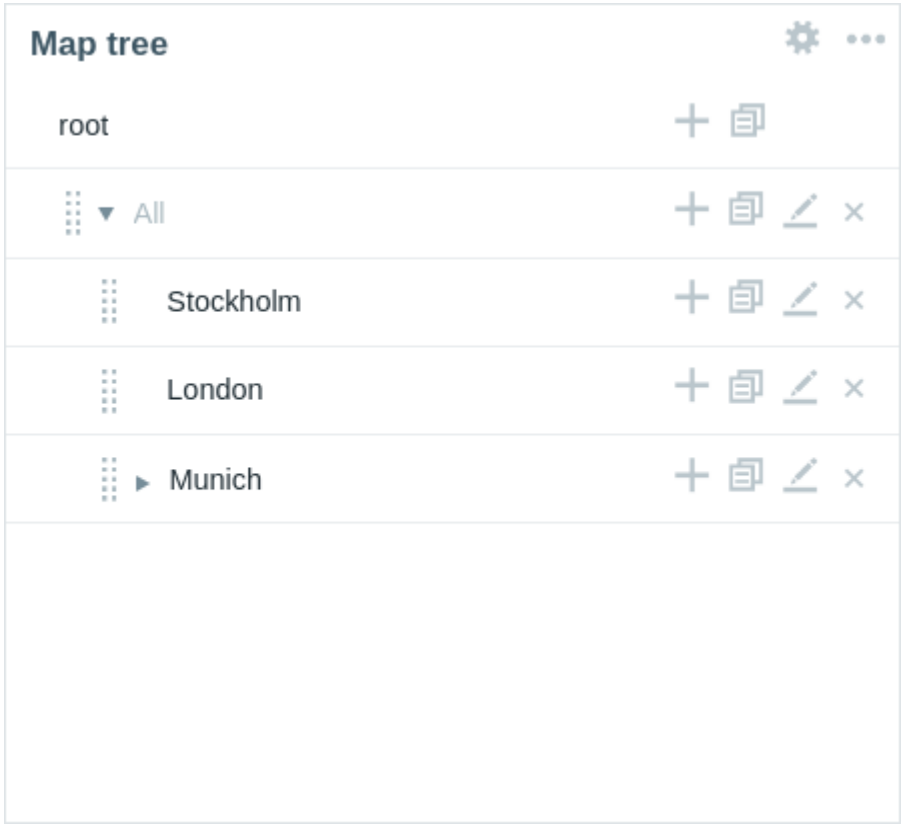


En plus des paramètres communs à tous les widgets, vous pouvez définir les options spécifiques suivantes :

Show unavailable maps	Cochez cette case pour afficher les cartes pour lesquelles l'utilisateur ne dispose pas de l'autorisation de lecture. Les cartes non disponibles dans l'arborescence de navigation seront affichées avec une icône grisée. Notez que si cette case est cochée, les sous-cartes disponibles sont affichées même si la carte de niveau parent n'est pas disponible. Si elle n'est pas cochée, les sous-cartes disponibles d'une carte parent non disponible ne seront pas affichées du tout. Le nombre de problèmes est calculé sur la base des cartes disponibles et des éléments de carte disponibles.

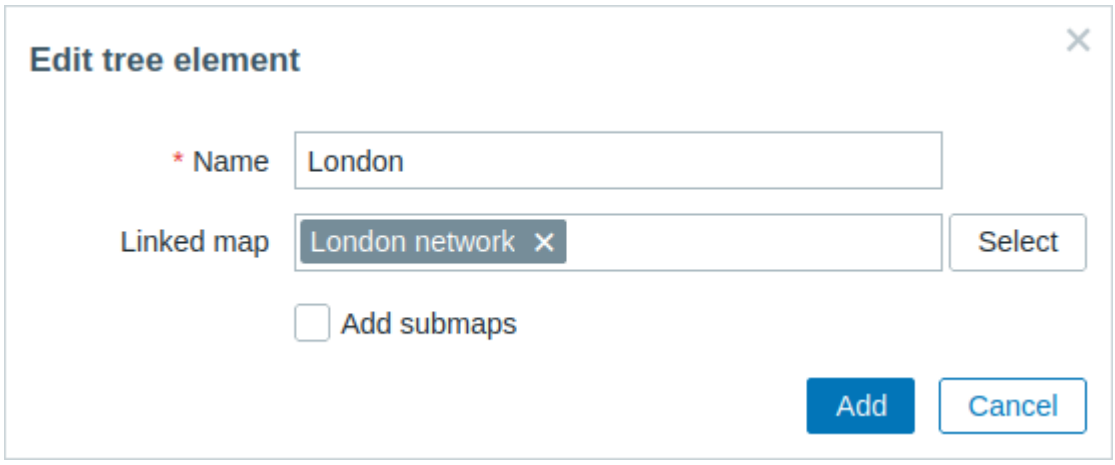
Les éléments de l'arborescence de navigation sont affichés dans une liste. Vous pouvez :

- faire glisser un élément (y compris ses éléments enfants) vers un nouvel emplacement dans la liste ;
- développer ou réduire un élément pour afficher ou masquer ses éléments enfants ;
- ajouter un élément enfant (avec ou sans carte liée) à un élément ;
- ajouter plusieurs éléments enfants (avec des cartes liées) à un élément ;
- modifier un élément ;
- supprimer un élément (y compris ses éléments enfants).



Configuration de l'élément

To configure a navigation tree element, either add a new element or edit an existing element.



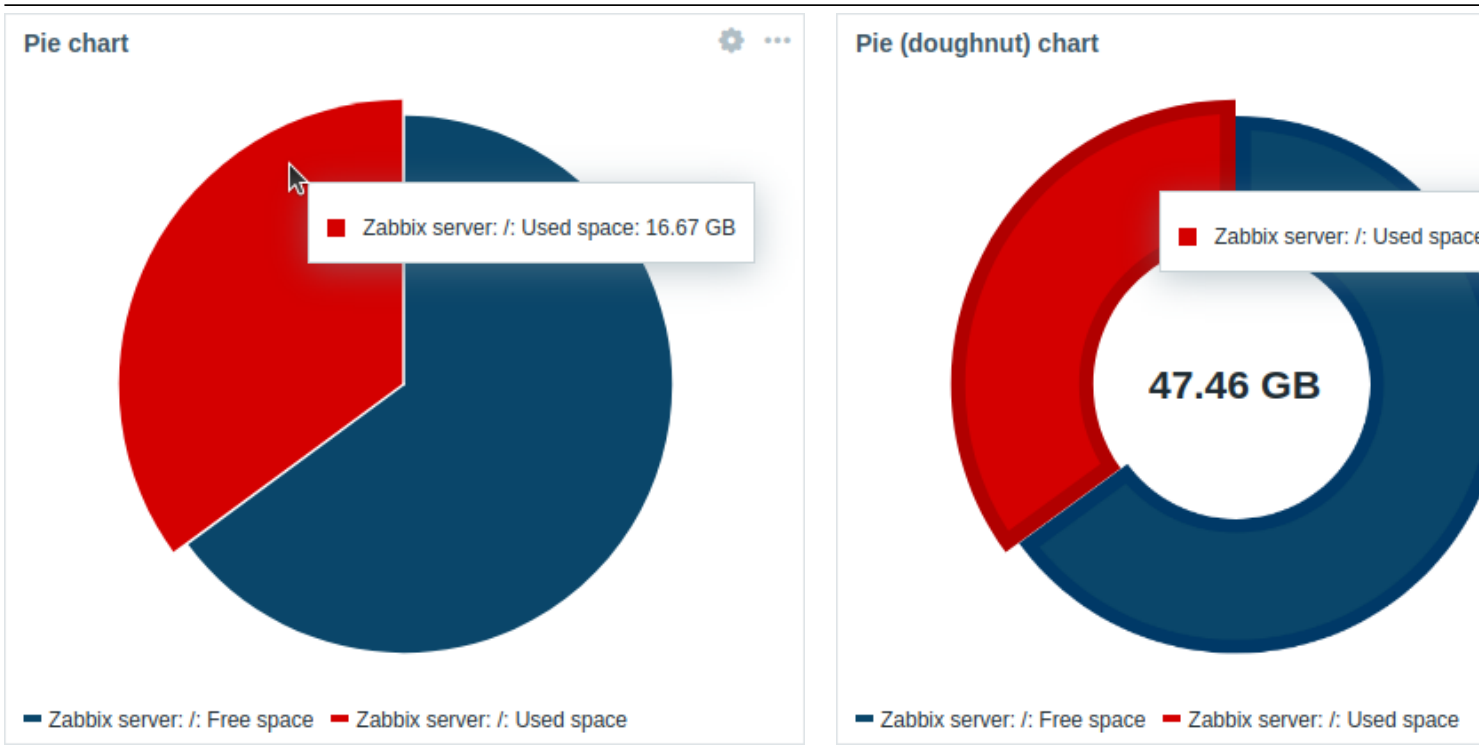
The following navigation tree element configuration parameters are available:

Name	Enter the navigation tree element name.
Linked map	Select the map to link to the navigation tree element. This field is auto-complete so starting to type the name of a map will offer a dropdown of matching maps.
Add submaps	Mark this checkbox to add the submaps of the linked map as child elements to the navigation tree element.

Diagramme circulaire

Présentation

Le widget *Graphique circulaire* affiche les données numériques des éléments sous forme de graphique circulaire ou en anneau basé sur des vecteurs. Ce widget peut vous aider à visualiser la contribution des éléments ou des hôtes à l'ensemble des données.

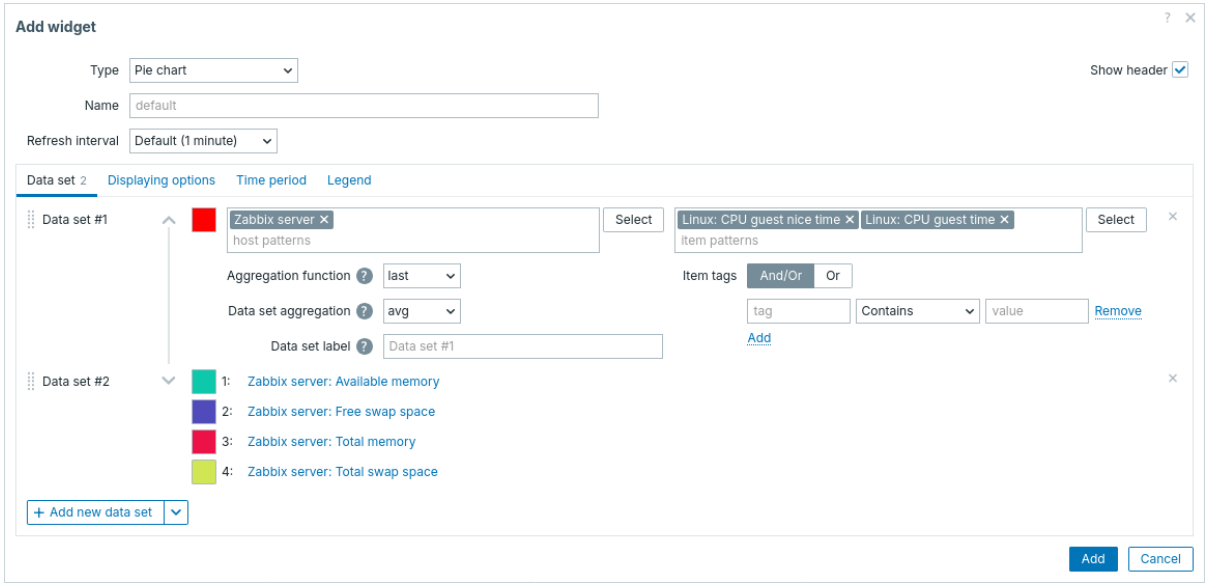


Au survol de la souris, le secteur survolé s'agrandit et affiche sa légende ; cliquer sur le secteur le maintient agrandi et sa légende visible jusqu'à sa fermeture.

Les informations affichées dans le widget *Graphique circulaire* peuvent être téléchargées sous forme d'image PNG en sélectionnant l'option *Télécharger l'image* dans le menu *Actions* du widget ⋯, qui apparaît lorsque vous survolez le widget. Selon les paramètres de votre navigateur, l'image est soit enregistrée automatiquement dans le dossier `Downloads`, soit une boîte de dialogue *Enregistrer* s'ouvre, vous permettant de choisir l'emplacement.

Configuration

Pour configurer, sélectionnez *Pie chart* comme type :



Jeu de données

L'onglet **Jeu de données** permet de sélectionner les données du graphique circulaire en ajoutant des jeux de données. Deux types de jeux de données peuvent être ajoutés :

- *Modèles d'éléments* - les données des éléments correspondants sont affichées. Vous pouvez choisir une couleur de base unique ou sélectionner une ligne de palette pour attribuer des couleurs distinctes à chaque élément correspondant.
- *Liste d'éléments* - les données des éléments sélectionnés sont affichées. Vous pouvez choisir individuellement la couleur de chaque élément à l'aide du sélecteur.

Par défaut, un jeu de données *Modèles d'éléments* est ajouté.

Jeu de données	Pour le jeu de données **Modèles d'éléments** : Sélectionnez ou saisissez des modèles d'hôtes et d'éléments ; les données des éléments correspondant aux modèles saisis seront affichées sur le graphique circulaire ; jusqu'à 50 éléments peuvent être affichés. Des modèles avec caractères génériques peuvent être utilisés pour la sélection (par exemple, * renvoie les résultats correspondant à zéro caractère ou plus). Pour spécifier un modèle avec caractère générique, saisissez manuellement la chaîne et appuyez sur *Entrée*. Le symbole générique est toujours interprété ; il n'est donc pas possible d'ajouter individuellement, par exemple, un élément nommé *item** s'il existe d'autres éléments correspondants (par exemple, *item2*, *item3*). La spécification de modèles d'hôtes et d'éléments est obligatoire pour les jeux de données « Modèles d'éléments ». Voir également : Détails de la configuration du jeu de données. Pour le jeu de données **Liste d'éléments** : Sélectionnez les éléments du graphique circulaire en cliquant sur le bouton *Ajouter un élément*. La liste déroulante *Type* située après le nom de l'élément permet de sélectionner le type d'affichage de chaque élément : **Normal** - la valeur de l'élément est représentée proportionnellement sur le graphique circulaire (par défaut) ; **Total** - la valeur de l'élément occupe la totalité du graphique circulaire. Notez qu'un seul élément « Total » peut exister par graphique circulaire et qu'il sera placé en premier dans la légende du graphique circulaire. Si un élément est défini sur « Total », le paramètre *Agrégation du jeu de données* (voir ci-dessous) sera désactivé et défini sur « non utilisé ». Vous pouvez également sélectionner des widgets compatibles comme source de données pour les éléments en cliquant sur le bouton *Ajouter un widget*. Les mêmes options s'appliquent aux widgets et aux éléments individuels. La spécification d'éléments ou de widgets est obligatoire pour les jeux de données « Liste d'éléments ». Voir également : Détails de la configuration du jeu de données. Notez que seuls les types d'éléments numériques sont autorisés. Lors de la configuration du widget sur un tableau de bord de modèle, le paramètre permettant de spécifier des modèles d'hôtes n'est pas disponible et le paramètre permettant de spécifier une liste d'éléments vous permet de sélectionner uniquement les éléments configurés sur le modèle.
Fonction d'agrégation	Spécifiez la fonction d'agrégation à utiliser pour chaque élément du jeu de données dans la *Période* sélectionnée : **min** - afficher la plus petite valeur ; **max** - afficher la plus grande valeur ; **avg** - afficher la valeur moyenne ; **sum** - afficher la somme des valeurs ; **count** - afficher le nombre de valeurs ; **first** - afficher la première valeur ; **last** - afficher la dernière valeur (par défaut). L'agrégation permet d'afficher une valeur agrégée pour l'intervalle spécifié (5 minutes, une heure, un jour). Seules les données numériques peuvent être affichées pour *min*, *max*, *avg* et *sum*. Pour *count*, les données non numériques sont converties en données numériques.

Agrégation du jeu de données	Spécifiez la fonction d'agrégation à utiliser pour l'ensemble du jeu de données : **non utilisé** - aucune agrégation, les éléments sont affichés séparément (par défaut) ; **min** - afficher la plus petite valeur ; **max** - afficher la plus grande valeur ; **avg** - afficher la valeur moyenne ; **sum** - afficher la somme des valeurs ; **count** - afficher le nombre de valeurs. L'agrégation permet d'afficher une valeur agrégée pour l'intervalle spécifié (5 minutes, une heure, un jour) sélectionné dans l'onglet *Période* ou utilisé pour l'ensemble du tableau de bord.
Libellé du jeu de données	Spécifiez un libellé personnalisé pour le jeu de données. Le libellé est affiché dans la configuration du jeu de données et dans la légende du graphique circulaire (pour les jeux de données agrégés). Tous les jeux de données sont numérotés, y compris ceux auxquels un *Libellé du jeu de données* a été attribué. Si aucun libellé n'est spécifié, le jeu de données reçoit automatiquement un libellé correspondant à son numéro (par exemple, « Jeu de données n° 2 », « Jeu de données n° 3 », etc.). La numérotation des jeux de données est recalculée après leur réorganisation ou leur déplacement par glisser-déposer. Les libellés de jeux de données trop longs sont raccourcis pour s'adapter à l'espace disponible (par exemple, « Nombre de proc... »).
Étiquettes des éléments	Spécifiez les étiquettes permettant de filtrer les éléments affichés dans le widget. Il est possible d'inclure ou d'exclure des étiquettes et des valeurs d'étiquettes spécifiques. Plusieurs conditions peuvent être définies. La correspondance des noms d'étiquettes est toujours sensible à la casse. Plusieurs opérateurs sont disponibles pour chaque condition : **Existe** - inclure les noms d'étiquettes spécifiés ; **Est égal à** - inclure les noms et valeurs d'étiquettes spécifiés (sensible à la casse) ; **Contient** - inclure les noms d'étiquettes spécifiés lorsque les valeurs d'étiquettes contiennent la chaîne saisie (correspondance de sous-chaîne, insensible à la casse) ; **N'existe pas** - exclure les noms d'étiquettes spécifiés ; **N'est pas égal à** - exclure les noms et valeurs d'étiquettes spécifiés (sensible à la casse) ; **Ne contient pas** - exclure les noms d'étiquettes spécifiés lorsque les valeurs d'étiquettes contiennent la chaîne saisie (correspondance de sous-chaîne, insensible à la casse). Deux types de calcul sont disponibles pour les conditions : **Et/Ou** - toutes les conditions doivent être remplies ; les conditions ayant le même nom d'étiquette sont regroupées avec la condition Ou ; **Ou** - une seule condition remplie suffit.

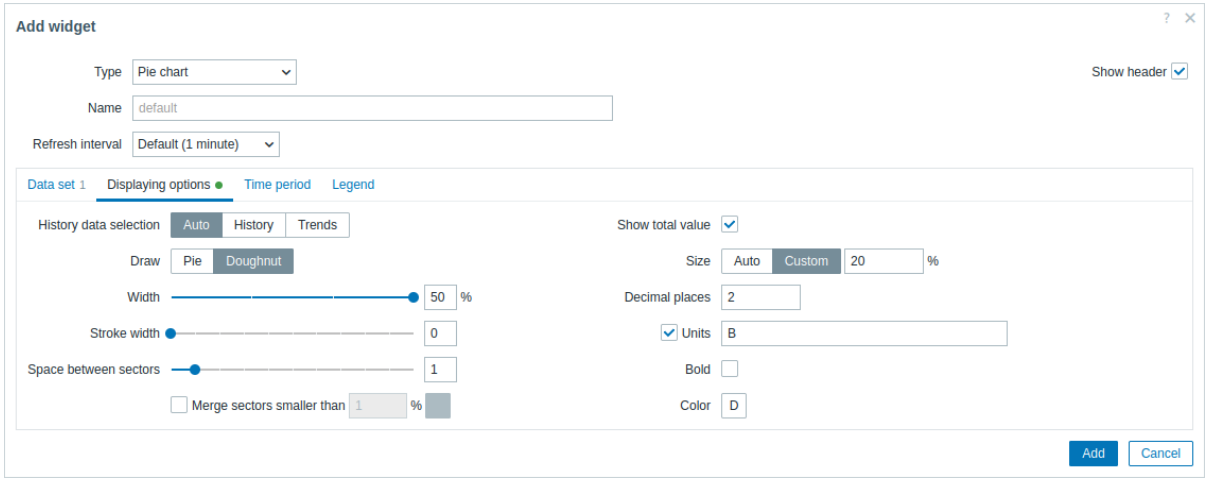
Détails de configuration des jeux de données

Les jeux de données existants sont affichés dans une liste. Vous pouvez réorganiser, développer/réduire, modifier les couleurs et cloner ces jeux de données.

Pour plus d'informations, consultez les détails de configuration des jeux de données dans le widget *Graph*. Ces détails s'appliquent également au widget *Pie chart*.

Options d'affichage

L'onglet **Options d'affichage** vous permet de définir les options de sélection et de visualisation des données d'historique pour le graphique circulaire :

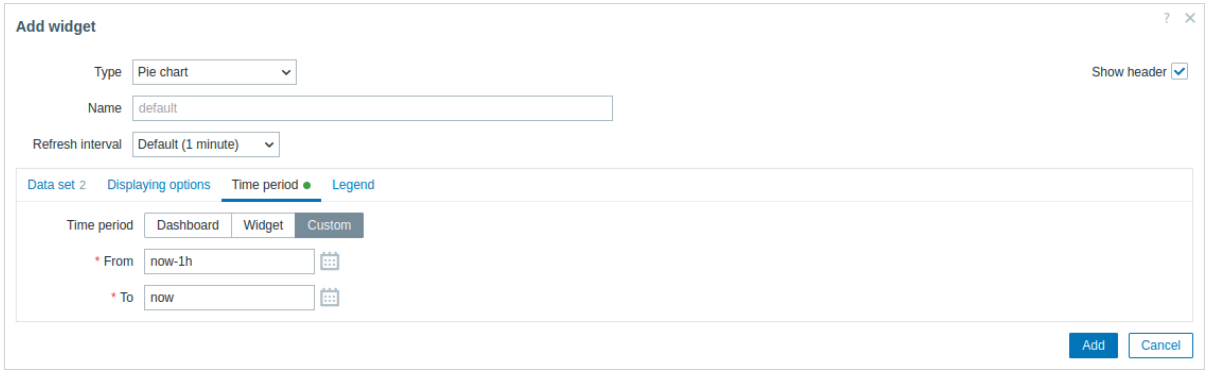


Sélection des données d'historique	Sélectionnez la source de données : **Auto** - les données sont obtenues conformément à l'algorithme classique (par défaut) ; **Historique** - données provenant de l'historique ; **Tendances** - données provenant des tendances.
Tracer	Sélectionnez le style de visualisation du graphique circulaire : **Circulaire** - un cercle plein (les secteurs occupent 100 % du rayon) ; **Anneau** - un cercle avec un espace vide au centre (les secteurs occupent jusqu'à 50 % du rayon).
Espace entre les secteurs	Sélectionnez la taille de l'espace (en unités de 0 à 10) entre les secteurs (la valeur par défaut est « 1 »).
Fusionner les secteurs inférieurs à N %	Cochez la case pour fusionner les secteurs inférieurs à N %. Si cette option est activée, sélectionnez la couleur des secteurs fusionnés et le seuil de pourcentage (N) pour la fusion des petits secteurs.
Style de tracé : ***Anneau***	
Largeur	Sélectionnez la largeur de l'anneau : 20, 30, 40 ou 50 % (valeur par défaut) du rayon.
Largeur du contour	Sélectionnez la largeur de la bordure du secteur de l'anneau (0 à 10).
Afficher la valeur totale	Cochez la case pour afficher la valeur totale au centre du graphique en anneau.
Taille	Sélectionnez l'option de taille pour la valeur totale : **Auto** - la taille du texte est ajustée automatiquement afin qu'il soit lisible au centre de l'anneau ; **Personnalisée** - indiquez la taille du texte sous forme de pourcentage de la hauteur totale du widget.
Décimales	Indiquez le nombre de décimales pour la valeur totale (0 à 6).
Unités	Indiquez les unités de la valeur totale.
Gras	Cochez la case pour afficher la valeur totale en gras.

Couleur	Sélectionnez la couleur de la valeur totale.

Période

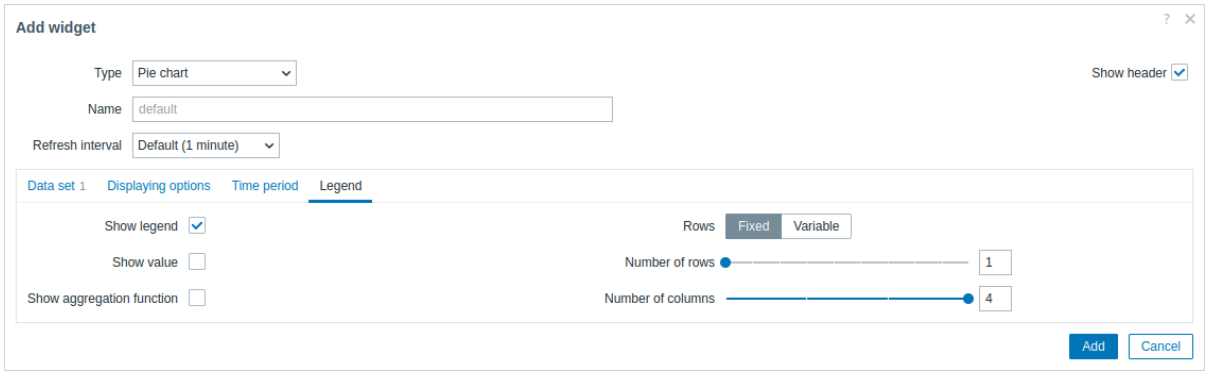
L'onglet **Période** vous permet de définir une période personnalisée pour les paramètres d'agrégation du graphique circulaire :



Période	Sélectionnez la source de données pour la période : **Tableau de bord** - utilisez le sélecteur de période du tableau de bord ; **Widget** - utilisez un widget compatible (défini dans le paramètre *Widget*) ; **Personnalisée** - utilisez une période personnalisée définie dans les paramètres *Du* et *Au* ; si elle est définie, une icône d'horloge s'affiche dans l'angle supérieur droit du widget et indique l'heure définie au survol de la souris. Notez que, quelle que soit la configuration de la *Période* du widget, les widgets compatibles peuvent toujours l'utiliser comme source de données pour la période.
Widget	Saisissez ou sélectionnez un widget compatible (*Graphique*, *Graphique (classique)*, *Prototype de graphique*) comme source de données pour la période. Ce paramètre est disponible si *Période* est définie sur « Widget ».
Du	Saisissez ou sélectionnez le début de la période. La syntaxe des périodes relatives (`now`, `now/d`, `now/w-1w`, etc.) est prise en charge. Ce paramètre est disponible si *Période* est définie sur « Personnalisée ».
Au	Saisissez ou sélectionnez la fin de la période. La syntaxe des périodes relatives (`now`, `now/d`, `now/w-1w`, etc.) est prise en charge. Ce paramètre est disponible si *Période* est définie sur « Personnalisée ».

Légende

L'onglet **Légende** vous permet de personnaliser la légende du graphique circulaire :



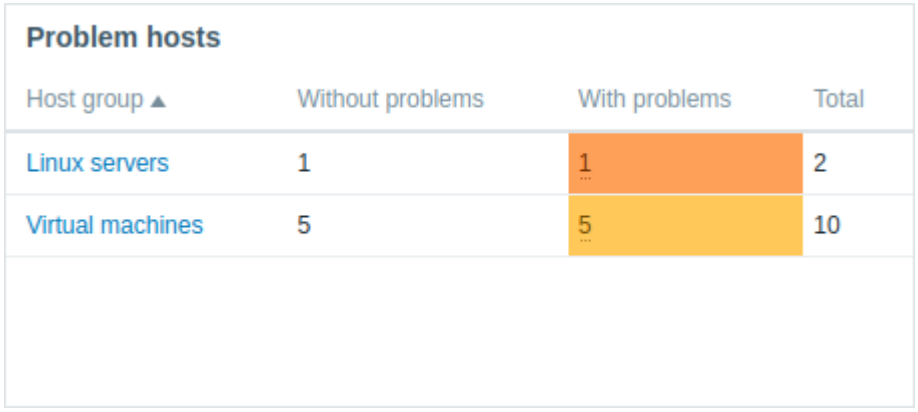
Afficher la légende	Décochez cette case pour masquer la légende sur le graphique circulaire (cochée par défaut).
Afficher la valeur	Cochez cette case pour afficher la valeur de l'élément dans la légende.
Afficher la fonction d'agrégation	Cochez cette case pour afficher la fonction d'agrégation dans la légende.
Lignes	Sélectionnez le mode d'affichage des lignes de la légende : **Fixe** - le nombre de lignes affichées est déterminé par la valeur du paramètre *Nombre de lignes* ; **Variable** - le nombre de lignes affichées est déterminé par le nombre d'éléments configurés, sans dépasser la valeur du paramètre *Nombre maximal de lignes*.

Nombre de lignes/ Nombre maximal de lignes	Si *Lignes* est défini sur « Fixe », définissez le nombre de lignes de légende à afficher (1-10). Si *Lignes* est défini sur « Variable », définissez le nombre maximal de lignes de légende à afficher (1-10).
Nombre de colonnes	Définissez le nombre de colonnes de légende à afficher (1-4). Ce paramètre est disponible si *Afficher la valeur* n'est pas coché.

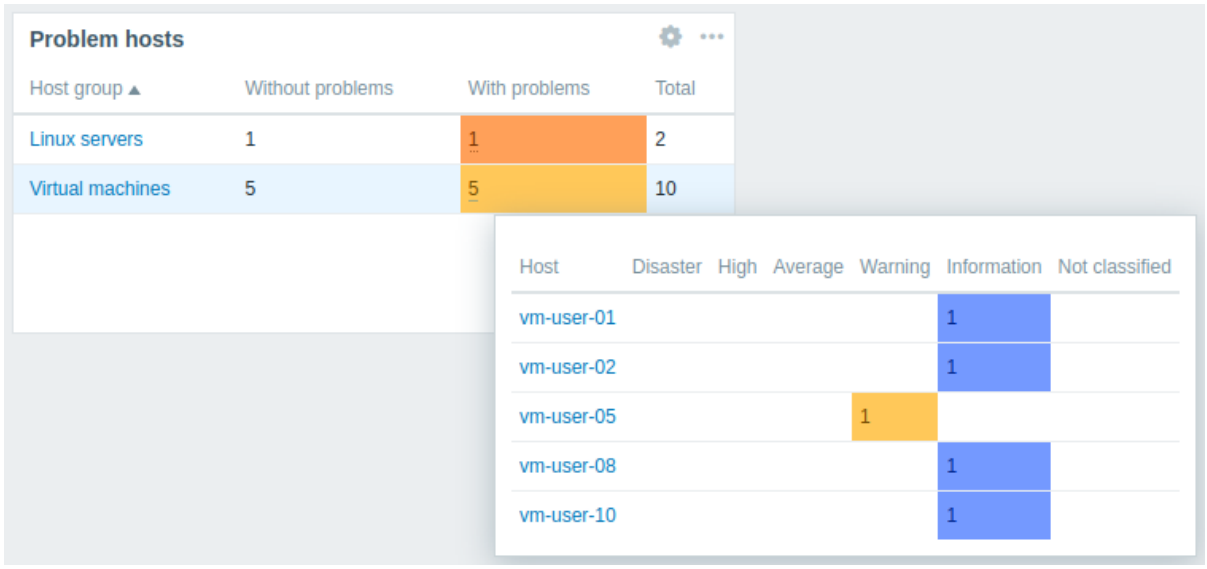
Hôtes en problème

Aperçu

Le widget *Hôtes avec problèmes* affiche le nombre de problèmes causes par groupe d'hôtes et indique la gravité de problème la plus élevée dans chaque groupe. Il permet de filtrer les problèmes selon différents paramètres (groupes d'hôtes, hôtes, noms de problèmes, etc.).

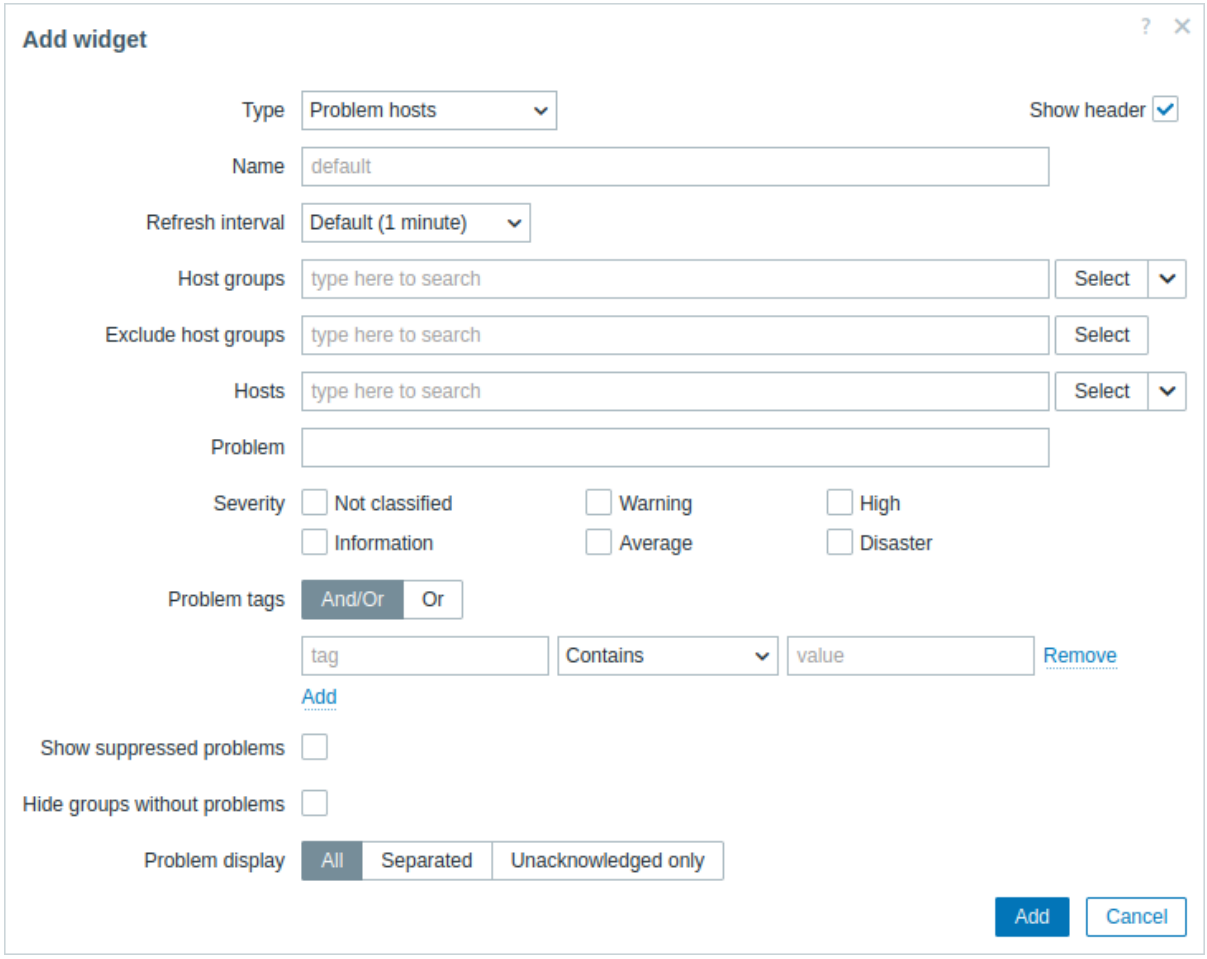


En survolant un nombre de problèmes, la liste des hôtes présentant des problèmes s'affiche :



Configuration

Pour configurer, sélectionnez *Hôtes avec problèmes* comme type :



En plus des paramètres communs à tous les widgets, vous pouvez définir les options spécifiques suivantes :

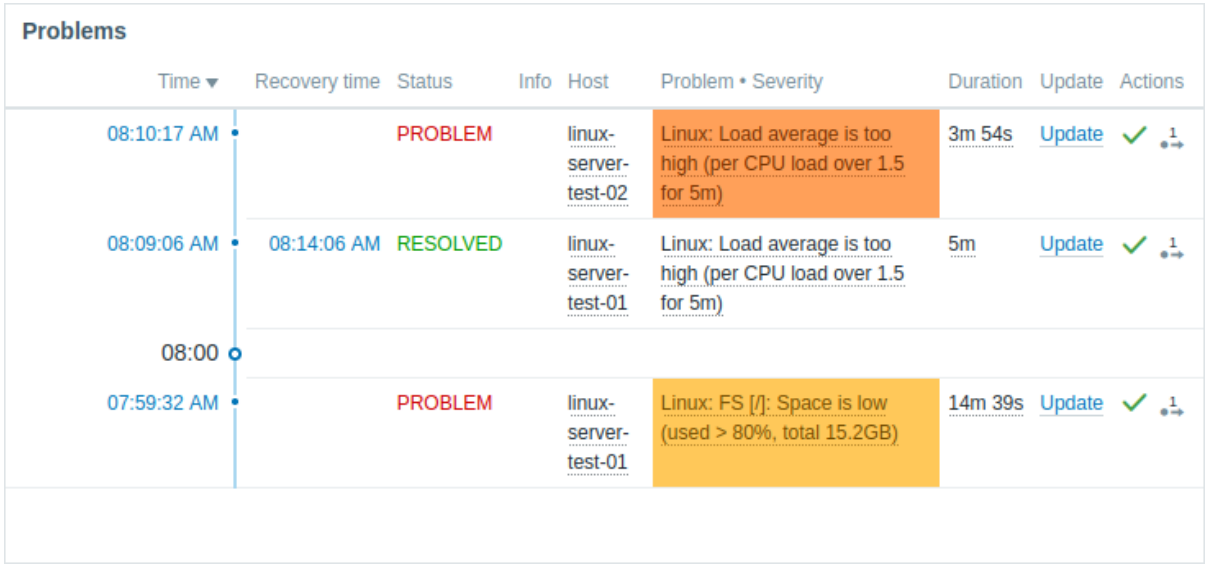
Groupes d'hôtes	Sélectionnez les groupes d'hôtes à afficher dans le widget. Vous pouvez également sélectionner un widget compatible comme source de données pour les groupes d'hôtes. Ce champ prend en charge l'autocomplétion ; ainsi, lorsque vous commencez à saisir le nom d'un groupe, une liste déroulante des groupes correspondants est proposée. La spécification d'un groupe d'hôtes parent sélectionne implicitement tous les groupes d'hôtes imbriqués. Les données des hôtes de ces groupes d'hôtes seront affichées dans le widget ; si aucun groupe d'hôtes n'est saisi, tous les groupes d'hôtes seront affichés. Ce paramètre n'est pas disponible lors de la configuration du widget sur un tableau de bord de modèle.
Exclure les groupes d'hôtes	Sélectionnez les groupes d'hôtes à masquer dans le widget. Ce champ prend en charge l'autocomplétion ; ainsi, lorsque vous commencez à saisir le nom d'un groupe, une liste déroulante des groupes correspondants est proposée. La spécification d'un groupe d'hôtes parent sélectionne implicitement tous les groupes d'hôtes imbriqués. Les données des hôtes de ces groupes d'hôtes ne seront pas affichées dans le widget. Par exemple, les hôtes 001, 002, 003 peuvent appartenir au Groupe A et les hôtes 002, 003 également au Groupe B. Si nous choisissons d'*afficher* le Groupe A et d'*exclure* le Groupe B en même temps, seules les données de l'hôte 001 seront affichées dans le tableau de bord. Ce paramètre n'est pas disponible lors de la configuration du widget sur un tableau de bord de modèle.
Hôtes	Sélectionnez les hôtes à afficher dans le widget. Vous pouvez également sélectionner un widget compatible ou le tableau de bord comme source de données pour les hôtes. Ce champ prend en charge l'autocomplétion ; ainsi, lorsque vous commencez à saisir le nom d'un hôte, une liste déroulante des hôtes correspondants est proposée. Si aucun hôte n'est saisi, tous les hôtes seront affichés. Ce paramètre n'est pas disponible lors de la configuration du widget sur un tableau de bord de modèle.

Problème	Vous pouvez limiter le nombre d'hôtes avec problèmes affichés en fonction du nom du problème. Si vous saisissez une chaîne ici, seuls les hôtes ayant des problèmes dont le nom contient la chaîne saisie seront affichés. Les macros ne sont pas développées.
Sévérité	Cochez les sévérités de problème pour filtrer les problèmes à afficher dans le widget. Si aucune sévérité n'est cochée, tous les problèmes seront affichés.
Étiquettes de problème	Spécifiez des étiquettes de problème pour limiter le nombre de problèmes affichés dans le widget. Il est possible d'inclure ou d'exclure des étiquettes et des valeurs d'étiquette spécifiques. Plusieurs conditions peuvent être définies. La correspondance des noms d'étiquette est toujours sensible à la casse. Plusieurs opérateurs sont disponibles pour chaque condition : **Existe** - inclure les noms d'étiquette spécifiés ; **Égal à** - inclure les noms d'étiquette et les valeurs spécifiés (sensible à la casse) ; **Contient** - inclure les noms d'étiquette spécifiés lorsque les valeurs d'étiquette contiennent la chaîne saisie (correspondance de sous-chaîne, insensible à la casse) ; **N'existe pas** - exclure les noms d'étiquette spécifiés ; **N'est pas égal à** - exclure les noms d'étiquette et les valeurs spécifiés (sensible à la casse) ; **Ne contient pas** - exclure les noms d'étiquette spécifiés lorsque les valeurs d'étiquette contiennent la chaîne saisie (correspondance de sous-chaîne, insensible à la casse). Deux types de calcul sont disponibles pour les conditions : **Et/Ou** - toutes les conditions doivent être remplies ; les conditions ayant le même nom d'étiquette seront regroupées par la condition *Ou* ; **Ou** - il suffit qu'une seule condition soit remplie.
Afficher les problèmes supprimés	Cochez la case pour afficher les problèmes qui seraient autrement supprimés (non affichés) en raison de la maintenance de l'hôte.
Masquer les groupes sans problèmes	Cochez l'option *Masquer les groupes sans problèmes* pour masquer dans le widget les données des groupes d'hôtes sans problèmes. Ce paramètre n'est pas disponible lors de la configuration du widget sur un tableau de bord de modèle.
Affichage des problèmes	Affichez le nombre de problèmes comme suit : **Tous** - le nombre total de problèmes sera affiché ; **Séparé** - le nombre de problèmes non acquittés sera affiché séparément, en tant que partie du nombre total de problèmes ; **Non acquittés uniquement** - seul le nombre de problèmes non acquittés sera affiché.

Problèmes

Vue d'ensemble

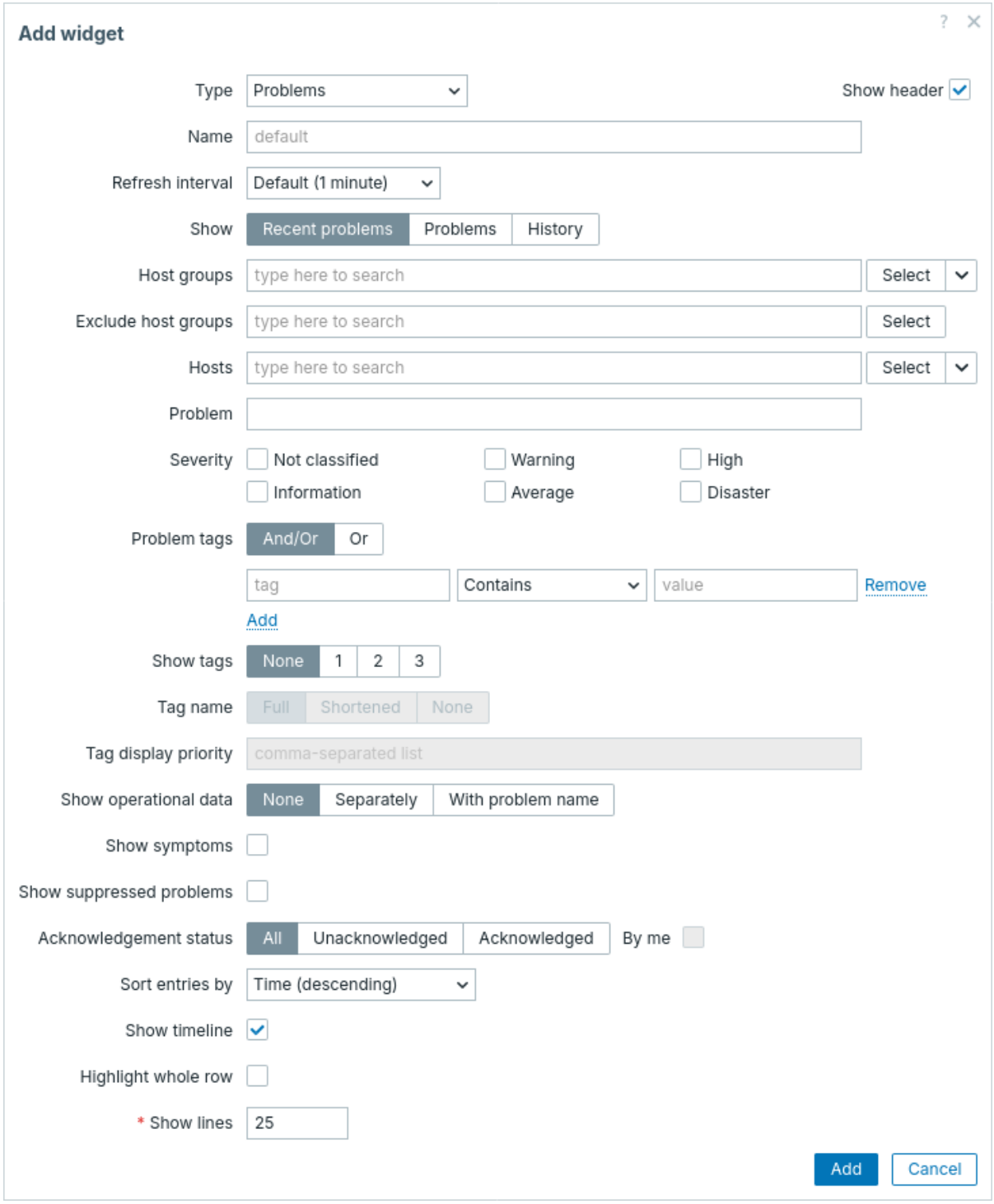
Le widget *Problems* affiche les problèmes actuels filtrés selon différents paramètres (groupes d'hôtes, hôtes, noms de problèmes, etc.), vous offrant une vue claire de ce qui nécessite votre attention.



Il affiche les mêmes données que *Monitoring* > *Problems* et peut afficher jusqu'à 1000 enregistrements.

Configuration

Pour configurer, sélectionnez *Problems* comme type :



Vous pouvez limiter le nombre de problèmes affichés dans le widget de différentes manières - par statut du problème, nom du problème, gravité, groupe d'hôtes, hôte, tag d'événement, statut d'acquittement, etc.

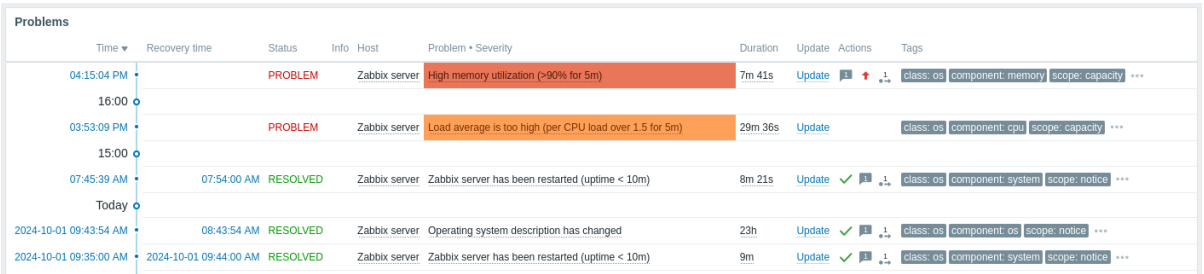
En plus des paramètres communs à tous les widgets, vous pouvez définir les options spécifiques suivantes :

Show	Filtrer par statut du problème : **Recent problems** - les problèmes non résolus et récemment résolus sont affichés (par défaut) ; **Problems** - les problèmes non résolus sont affichés ; **History** - l'historique de tous les événements est affiché.

Host groups	Sélectionnez les groupes d'hôtes dont les problèmes doivent être affichés dans le widget. Vous pouvez également sélectionner un widget compatible comme source de données pour les groupes d'hôtes. Ce champ est à saisie semi-automatique ; commencer à taper le nom d'un groupe proposera une liste déroulante des groupes correspondants. Spécifier un groupe d'hôtes parent sélectionne implicitement tous les groupes d'hôtes imbriqués. Les problèmes provenant de ces groupes d'hôtes seront affichés dans le widget ; si aucun groupe d'hôtes n'est saisi, les problèmes de tous les groupes d'hôtes seront affichés. Ce paramètre n'est pas disponible lors de la configuration du widget sur un tableau de bord de modèle.
Exclude host groups	Sélectionnez les groupes d'hôtes dont les problèmes doivent être masqués dans le widget. Ce champ est à saisie semi-automatique ; commencer à taper le nom d'un groupe proposera une liste déroulante des groupes correspondants. Spécifier un groupe d'hôtes parent sélectionne implicitement tous les groupes d'hôtes imbriqués. Les problèmes provenant de ces groupes d'hôtes ne seront pas affichés dans le widget. Par exemple, les hôtes 001, 002, 003 peuvent appartenir au groupe A et les hôtes 002, 003 également au groupe B. Si nous choisissons d'*afficher* le groupe A et d'*exclure* le groupe B en même temps, seuls les problèmes de l'hôte 001 seront affichés dans le widget. Ce paramètre n'est pas disponible lors de la configuration du widget sur un tableau de bord de modèle.
Hosts	Sélectionnez les hôtes dont les problèmes doivent être affichés dans le widget. Vous pouvez également sélectionner un widget compatible ou le tableau de bord comme source de données pour les hôtes. Ce champ est à saisie semi-automatique ; commencer à taper le nom d'un hôte proposera une liste déroulante des hôtes correspondants. Si aucun hôte n'est saisi, les problèmes de tous les hôtes seront affichés. Ce paramètre n'est pas disponible lors de la configuration du widget sur un tableau de bord de modèle.
Problem	Vous pouvez limiter le nombre de problèmes affichés par leur nom. Si vous saisissez une chaîne ici, seuls les problèmes dont le nom contient la chaîne saisie seront affichés. Les macros ne sont pas développées.
Severity	Cochez les gravités des problèmes pour filtrer les problèmes affichés dans le widget. Si aucune gravité n'est cochée, tous les problèmes seront affichés.
Problem tags	Spécifiez des tags de problème pour limiter le nombre de problèmes affichés dans le widget. Il est possible d'inclure ainsi que d'exclure des tags et des valeurs de tag spécifiques. Plusieurs conditions peuvent être définies. La correspondance du nom du tag est toujours sensible à la casse. Plusieurs opérateurs sont disponibles pour chaque condition : **Exists** - inclure les noms de tag spécifiés ; **Equals** - inclure les noms et valeurs de tag spécifiés (sensible à la casse) ; **Contains** - inclure les noms de tag spécifiés dont les valeurs de tag contiennent la chaîne saisie (correspondance de sous-chaîne, insensible à la casse) ; **Does not exist** - exclure les noms de tag spécifiés ; **Does not equal** - exclure les noms et valeurs de tag spécifiés (sensible à la casse) ; **Does not contain** - exclure les noms de tag spécifiés dont les valeurs de tag contiennent la chaîne saisie (correspondance de sous-chaîne, insensible à la casse). Il existe deux types de calcul pour les conditions : **And/Or** - toutes les conditions doivent être remplies, les conditions ayant le même nom de tag seront regroupées par la condition *Or* ; **Or** - il suffit qu'une condition soit remplie. Lors du filtrage, les tags spécifiés ici seront affichés en premier avec le problème, sauf s'ils sont remplacés par la liste *Tag display priority* (voir ci-dessous).
Show tags	Sélectionnez le nombre de tags affichés : **None** - aucune colonne *Tags* ; **1** - la colonne *Tags* contient un tag ; **2** - la colonne *Tags* contient deux tags ; **3** - la colonne *Tags* contient trois tags. Pour voir tous les tags du problème, survolez l'icône à trois points avec la souris.

Tag name	Sélectionnez le mode d'affichage des noms de tag : **Full** - les noms et valeurs des tags sont affichés en entier ; **Shortened** - les noms des tags sont abrégés à 3 symboles, mais les valeurs des tags sont affichées en entier ; **None** - seules les valeurs des tags sont affichées ; aucun nom.
Tag display priority	Saisissez la priorité d'affichage des tags pour un problème, sous forme de liste de tags séparés par des virgules. Seuls les noms de tag doivent être utilisés, pas les valeurs. Exemple : `Services,Applications,Application` Les tags de cette liste seront toujours affichés en premier, en remplaçant l'ordre naturel alphabétique.
Show operational data	Sélectionnez le mode d'affichage des données opérationnelles : **None** - aucune donnée opérationnelle n'est affichée ; **Separately** - les données opérationnelles sont affichées dans une colonne séparée ; **With problem name** - ajoute les données opérationnelles au nom du problème, en utilisant des parenthèses pour les données opérationnelles. Pour plus de détails, voir la description du champ *Operational data* dans la configuration du déclencheur.
Show symptoms	Cochez la case pour afficher sur sa propre ligne les problèmes classés comme symptômes.
Show suppressed problems	Cochez la case pour afficher les problèmes qui seraient autrement supprimés (non affichés) en raison de la maintenance de l'hôte ou d'une suppression de problème unique.
Acknowledgement status	Filtrer pour afficher tous les problèmes, uniquement les problèmes non acquittés, ou uniquement les problèmes acquittés. Cochez la case supplémentaire pour exclure les problèmes que vous avez déjà acquittés.
Sort entries by	Trier les entrées par : **Time** (croissant ou décroissant) ; **Severity** (croissant ou décroissant) ; **Problem name** (croissant ou décroissant) ; **Host** (croissant ou décroissant). Le tri des entrées par **Host** (croissant ou décroissant) n'est pas disponible lors de la configuration du widget sur un tableau de bord de modèle.
Show timeline	Cochez la case pour afficher une chronologie visuelle.
Highlight whole row	Cochez la case pour mettre en surbrillance la ligne entière pour les problèmes non résolus. La couleur de gravité du problème est utilisée pour la mise en surbrillance. *Highlight whole row* n'est pas disponible dans les thèmes à contraste élevé.
Show lines	Spécifiez le nombre de lignes de problèmes à afficher.

Utilisation du widget



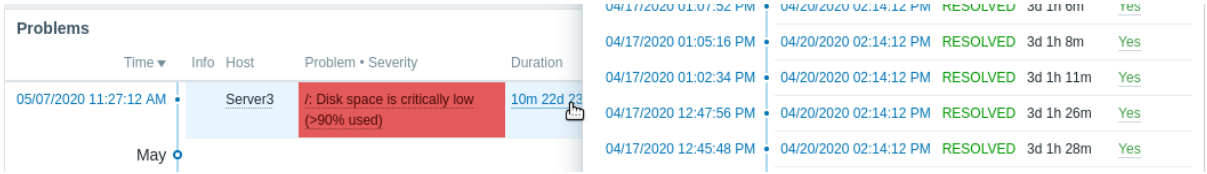
Le widget Problems offre un accès rapide à des informations supplémentaires :

- Cliquez sur la date du problème (dans la colonne Time) ou sur la date de rétablissement (dans la colonne Recovery Time) pour afficher les détails de l'événement.
- Si le champ Info n'est pas vide, survolez l'icône affichée pour voir des détails supplémentaires.
- Cliquez sur le nom de l'hôte pour ouvrir le menu de l'hôte.
- Cliquez sur le nom du problème pour ouvrir le menu de l'événement.
- Survolez ou cliquez sur la durée du problème pour afficher la fenêtre contextuelle de l'événement de problème.
- Cliquer sur *Update* ouvre une fenêtre Mettre à jour le problème.
- Survolez ou cliquez sur l'icône de flèche grise dans la colonne Actions pour afficher la liste des actions exécutées.

Fenêtre contextuelle de l'événement de problème

La fenêtre contextuelle de l'événement de problème inclut la liste des événements de problème pour ce déclencheur et, si elle est

définie, la description du déclencheur ainsi qu'une URL cliquable.



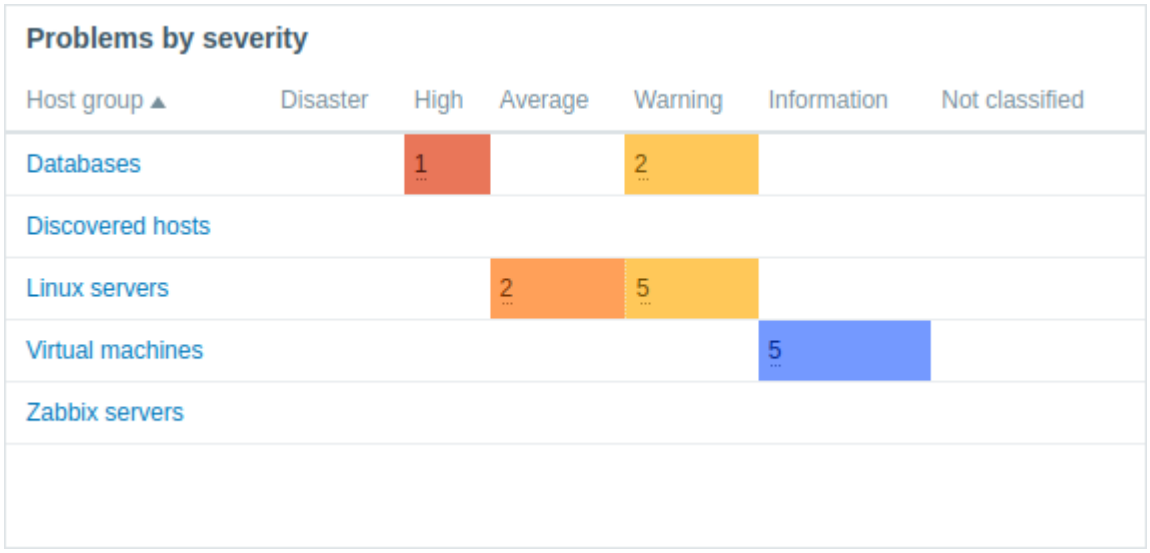
Pour afficher la fenêtre contextuelle de l'événement de problème :

- Passez la souris sur la durée du problème dans la colonne *Duration* du widget *Problems*. La fenêtre contextuelle disparaît dès que vous retirez la souris de la durée.
- Cliquez sur la durée dans la colonne *Duration* du widget *Problems*. La fenêtre contextuelle disparaît uniquement si vous cliquez à nouveau sur la durée.

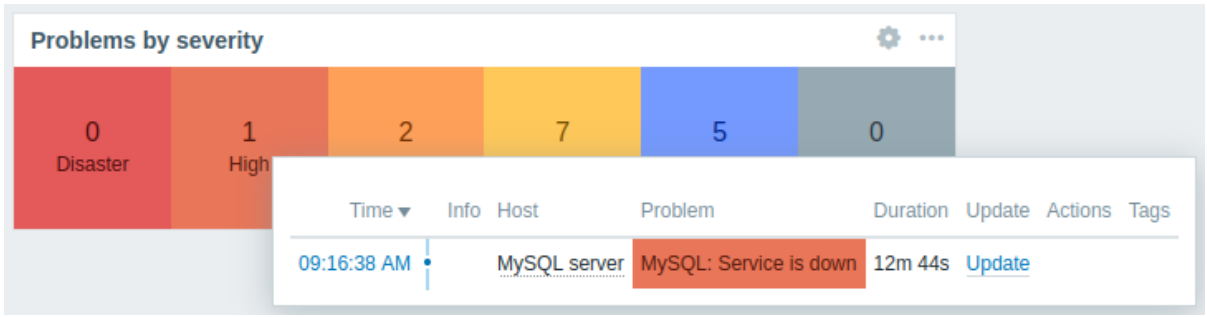
Problèmes par gravité

Vue d'ensemble

Le widget *Problems by severity* affiche le nombre de problèmes causes regroupés par gravité. Il permet de filtrer les problèmes selon différents paramètres (groupes d'hôtes, hôtes, noms de problèmes, etc.).

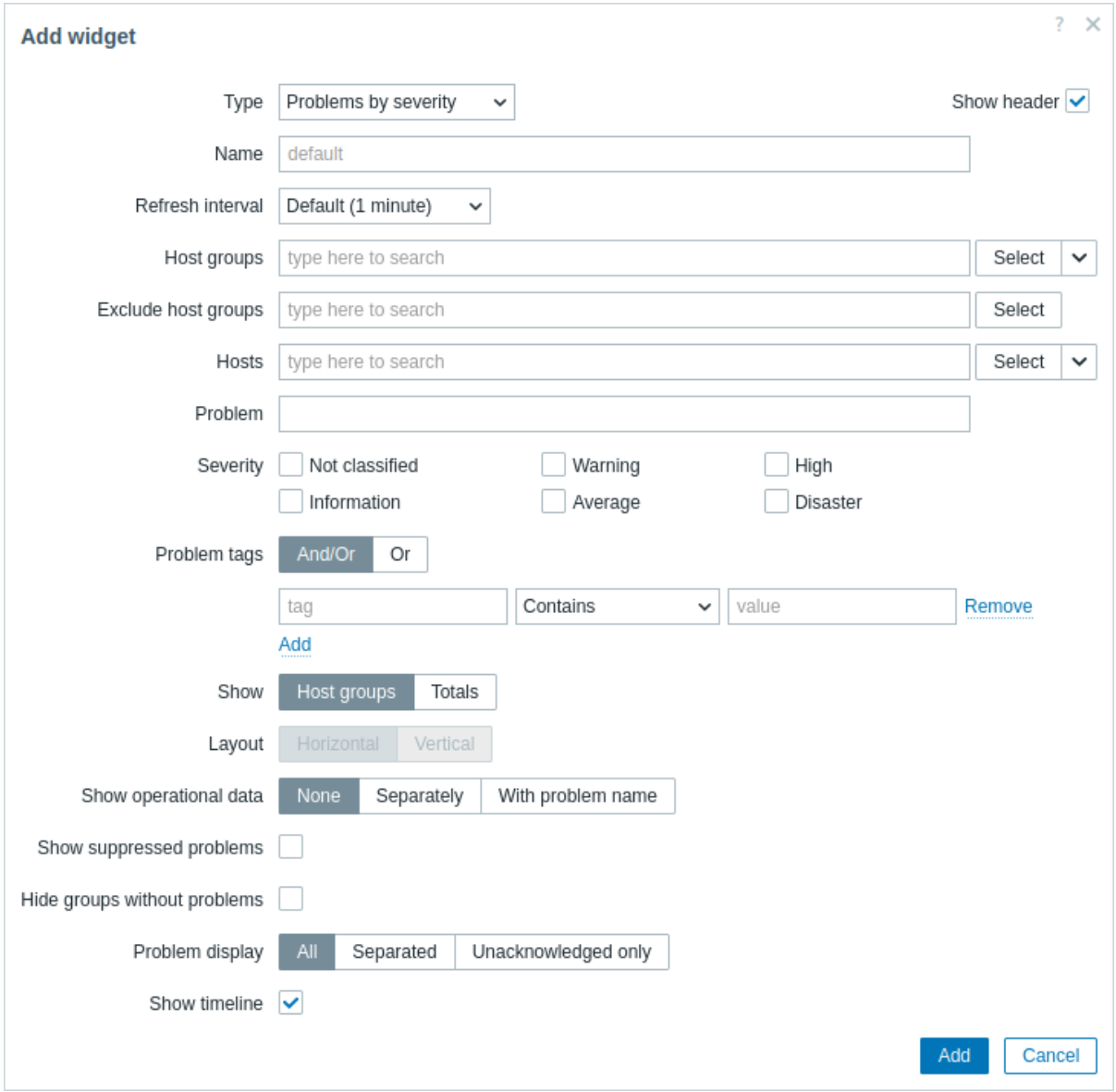


En survolant un nombre de problèmes, la liste des problèmes associés s'affiche :



Configuration

Pour configurer, sélectionnez *Problems by severity* comme type :



En plus des paramètres communs à tous les widgets, vous pouvez définir les options spécifiques suivantes :

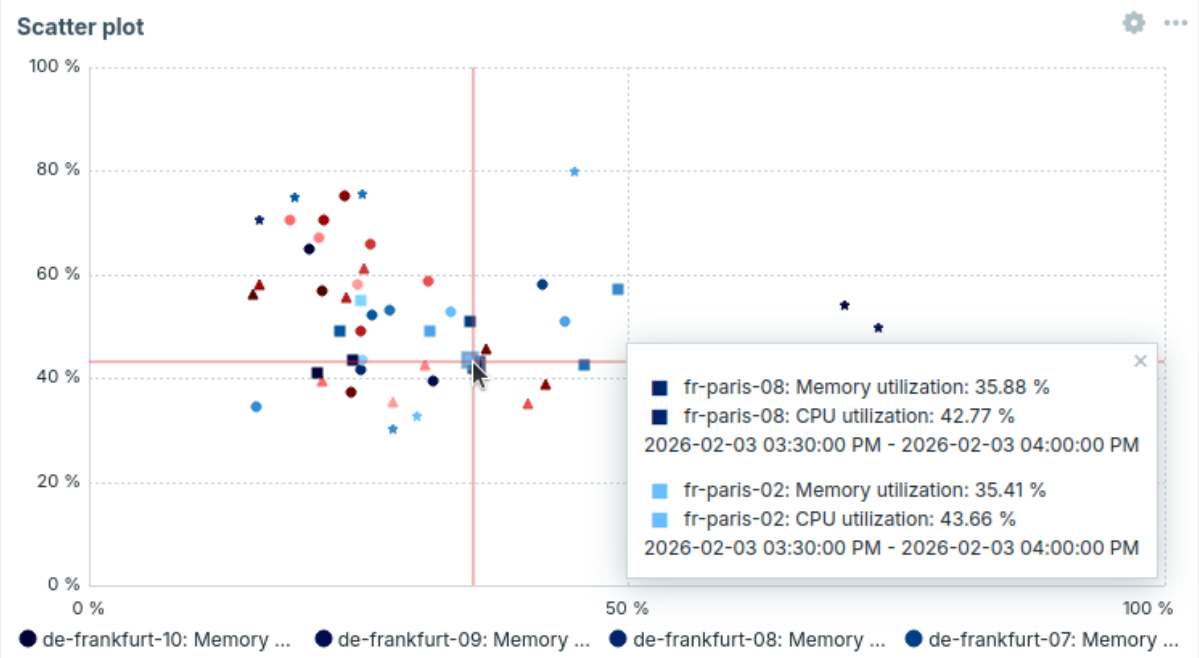
Host groups	Sélectionnez les groupes d'hôtes à afficher dans le widget. Vous pouvez également sélectionner un widget compatible comme source de données pour les groupes d'hôtes. Ce champ est à saisie semi-automatique : commencer à taper le nom d'un groupe affichera une liste déroulante des groupes correspondants. Spécifier un groupe d'hôtes parent sélectionne implicitement tous les groupes d'hôtes imbriqués. Les données des hôtes provenant de ces groupes d'hôtes seront affichées dans le widget ; si aucun groupe d'hôtes n'est saisi, tous les groupes d'hôtes seront affichés. Ce paramètre n'est pas disponible lors de la configuration du widget sur un tableau de bord de modèle.
Exclude host groups	Sélectionnez les groupes d'hôtes à masquer dans le widget. Ce champ est à saisie semi-automatique : commencer à taper le nom d'un groupe affichera une liste déroulante des groupes correspondants. Spécifier un groupe d'hôtes parent sélectionne implicitement tous les groupes d'hôtes imbriqués. Les données des hôtes provenant de ces groupes d'hôtes ne seront pas affichées. Par exemple, les hôtes 001, 002, 003 peuvent appartenir au groupe A et les hôtes 002, 003 également au groupe B. Si nous choisissons d'*afficher* le groupe A et d'*exclure* le groupe B en même temps, seules les données de l'hôte 001 seront affichées dans le tableau de bord. Ce paramètre n'est pas disponible lors de la configuration du widget sur un tableau de bord de modèle.

Hosts	Sélectionnez les hôtes à afficher dans le widget. Vous pouvez également sélectionner un widget compatible ou le tableau de bord comme source de données pour les hôtes. Ce champ est à saisie semi-automatique : commencer à taper le nom d'un hôte affichera une liste déroulante des hôtes correspondants. Si aucun hôte n'est saisi, tous les hôtes seront affichés. Ce paramètre n'est pas disponible lors de la configuration du widget sur un tableau de bord de modèle.
Problem	Vous pouvez limiter le nombre d'hôtes en problème affichés par le nom du problème. Si vous saisissez une chaîne ici, seuls les hôtes ayant des problèmes dont le nom contient la chaîne saisie seront affichés. Les macros ne sont pas développées.
Severity	Cochez les niveaux de gravité des problèmes pour filtrer les problèmes à afficher dans le widget. Si aucun niveau de gravité n'est coché, tous les problèmes seront affichés.
Problem tags	Spécifiez des tags de problème pour limiter le nombre de problèmes affichés dans le widget. Il est possible d'inclure ou d'exclure des tags et des valeurs de tag spécifiques. Plusieurs conditions peuvent être définies. La correspondance du nom du tag est toujours sensible à la casse. Plusieurs opérateurs sont disponibles pour chaque condition : **Exists** - inclure les noms de tag spécifiés ; **Equals** - inclure les noms et valeurs de tag spécifiés (sensible à la casse) ; **Contains** - inclure les noms de tag spécifiés lorsque les valeurs de tag contiennent la chaîne saisie (correspondance de sous-chaîne, insensible à la casse) ; **Does not exist** - exclure les noms de tag spécifiés ; **Does not equal** - exclure les noms et valeurs de tag spécifiés (sensible à la casse) ; **Does not contain** - exclure les noms de tag spécifiés lorsque les valeurs de tag contiennent la chaîne saisie (correspondance de sous-chaîne, insensible à la casse). Il existe deux types de calcul pour les conditions : **And/Or** - toutes les conditions doivent être remplies, les conditions ayant le même nom de tag seront regroupées par la condition *Or* ; **Or** - une seule condition suffit.
Show	Sélectionnez l'option d'affichage : **Host groups** - afficher les problèmes par groupe d'hôtes ; **Totals** - afficher un total des problèmes pour tous les groupes d'hôtes sélectionnés dans des blocs colorés correspondant à la gravité du problème. Ce paramètre n'est pas disponible lors de la configuration du widget sur un tableau de bord de modèle, et seul un total des problèmes sera affiché.
Layout	Sélectionnez l'option de disposition : **Horizontal** - les blocs colorés des totaux seront affichés horizontalement ; **Vertical** - les blocs colorés des totaux seront affichés verticalement. Ce paramètre est disponible si *Show* est défini sur "Totals".
Show operational data	Afficher les données opérationnelles comme suit : **None** - les données opérationnelles ne seront pas affichées **Separately** - les données opérationnelles seront affichées sur une ligne séparée **With problem name** - les données opérationnelles seront affichées avec le nom du problème. Pour plus de détails, consultez la description du champ *Operational data* dans la configuration du déclencheur.
Show suppressed problems	Cochez la case pour afficher les problèmes qui seraient autrement supprimés (non affichés) en raison de la maintenance de l'hôte.
Hide groups without problems	Cochez l'option *Hide groups without problems* pour masquer dans le widget les données des groupes d'hôtes sans problèmes. Ce paramètre n'est pas disponible lors de la configuration du widget sur un tableau de bord de modèle.
Problem display	Afficher le nombre de problèmes comme suit : **All** - le nombre total de problèmes sera affiché ; **Separated** - le nombre de problèmes non reconnus sera affiché séparément du nombre total de problèmes ; **Unacknowledged only** - seul le nombre de problèmes non reconnus sera affiché.
Show timeline	Cochez la case pour afficher une chronologie visuelle.

Nuage de points

Vue d'ensemble

Le widget *Nuage de points* affiche la relation entre deux métriques en représentant des points de données individuels le long des axes X et Y.
Cela permet de révéler les tendances, les regroupements, les corrélations et les valeurs aberrantes dans l'ensemble de données.



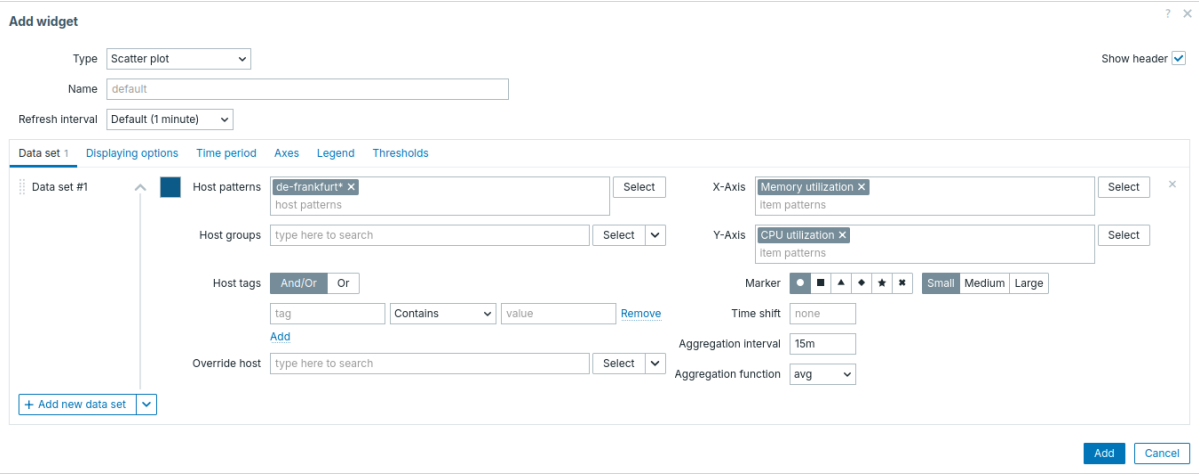
Lorsque vous survolez un point de données, une info-bulle affiche une liste des éléments associés à ce point de données ainsi qu'aux autres points situés dans un rayon de 6 px ; les points sont d'abord triés selon les valeurs de l'axe X du nuage de points, par ordre croissant, puis selon les valeurs de l'axe Y, par ordre décroissant.

La sélection d'un élément dans l'info-bulle diffuse ses données vers les autres widgets.

Les informations affichées dans le widget *Nuage de points* peuvent être téléchargées sous forme d'image PNG en sélectionnant l'option *Télécharger l'image* dans le menu *Actions* du widget ⋯, qui apparaît lorsque vous survolez le widget.
Selon les paramètres de votre navigateur, l'image est soit enregistrée automatiquement dans le dossier `Downloads`, soit une boîte de dialogue *Enregistrer* s'ouvre et vous permet de choisir l'emplacement.

Configuration

Pour configurer, sélectionnez *Scatter plot* comme type :



Ensemble de données

L'onglet **Ensemble de données** permet de sélectionner les données du nuage de points en ajoutant des ensembles de données.
Deux types d'ensembles de données peuvent être ajoutés :

- *Modèles d'éléments* - les hôtes contenant des éléments correspondants sont affichés comme points de données. Vous pouvez choisir une couleur de base unique ou sélectionner une ligne de palette afin d'attribuer des couleurs distinctes à chaque hôte contenant les éléments correspondants.
- *Liste d'éléments* - les éléments sélectionnés sont affichés comme points de données, chaque point représentant une paire d'éléments (X et Y). Vous pouvez choisir individuellement la couleur de chaque ensemble de données à l'aide du sélecteur.

Les points de données sont créés uniquement pour les paires hôte/élément qui possèdent au moins une valeur sur chacun des deux axes.

Par défaut, un ensemble de données *Modèles d'éléments* est ajouté.

Ensemble de données	Pour l'ensemble de données **Modèles d'éléments** : Sélectionnez ou saisissez des modèles d'hôtes et d'éléments ; les données des éléments correspondant à ces modèles seront affichées dans le nuage de points. Notez que si le nombre de points de données dépasse 8k, `memory_limit` doit être augmenté dans le fichier de configuration `php.ini`. La limite de mémoire recommandée pour 20k points de données est de 512 Mo. Des modèles avec caractères génériques peuvent être utilisés pour la sélection (par exemple, ∗ renvoie les résultats correspondant à zéro caractère ou plus). Pour spécifier un modèle avec caractère générique, saisissez manuellement la chaîne et appuyez sur *Entrée*. Le symbole générique est toujours interprété ; il n'est donc pas possible d'ajouter individuellement, par exemple, un élément nommé *item** s'il existe d'autres éléments correspondants (par exemple, *item2*, *item3*). La spécification de modèles d'hôtes et d'éléments est obligatoire pour les ensembles de données « Modèles d'éléments ». Voir également : Détails de la configuration de l'ensemble de données. Pour l'ensemble de données **Liste d'éléments** : Sélectionnez les éléments du nuage de points en cliquant sur le bouton *Ajouter un élément*. Vous pouvez également sélectionner des widgets compatibles comme source de données pour les éléments en cliquant sur le bouton *Ajouter un widget*. La spécification d'éléments ou de widgets est obligatoire pour les ensembles de données « Liste d'éléments ». Voir également : Détails de la configuration de l'ensemble de données. Notez que seuls les types d'éléments numériques sont autorisés. Lors de la configuration du widget sur un tableau de bord de modèle, le paramètre permettant de spécifier des modèles d'hôtes n'est pas disponible, et le paramètre permettant de spécifier une liste d'éléments autorise uniquement la sélection des éléments configurés sur le modèle.
Groupes d'hôtes	Sélectionnez des groupes d'hôtes. Vous pouvez également sélectionner un widget compatible comme source de données pour les groupes d'hôtes. Ce champ propose la saisie semi-automatique : lorsque vous commencez à saisir le nom d'un groupe, une liste déroulante de groupes correspondants s'affiche. La spécification d'un groupe d'hôtes parent sélectionne implicitement tous les groupes d'hôtes imbriqués. Les données des hôtes appartenant à ces groupes d'hôtes seront affichées dans le widget ; si aucun groupe d'hôtes n'est saisi, tous les groupes d'hôtes seront affichés. Ce paramètre est disponible uniquement pour les ensembles de données *Modèles d'éléments* et n'est pas disponible lors de la configuration du widget sur un tableau de bord de modèle.

Étiquettes d'hôtes	Spécifiez les étiquettes permettant de filtrer les hôtes affichés dans le widget. Il est possible d'inclure ou d'exclure des étiquettes et des valeurs d'étiquettes spécifiques. Plusieurs conditions peuvent être définies. La correspondance des noms d'étiquettes est toujours sensible à la casse. Plusieurs opérateurs sont disponibles pour chaque condition : **Existe** - inclure les noms d'étiquettes spécifiés ; **Est égal à** - inclure les noms et valeurs d'étiquettes spécifiés (sensible à la casse) ; **Contient** - inclure les noms d'étiquettes dont les valeurs contiennent la chaîne saisie (correspondance de sous-chaîne, insensible à la casse) ; **N'existe pas** - exclure les noms d'étiquettes spécifiés ; **N'est pas égal à** - exclure les noms et valeurs d'étiquettes spécifiés (sensible à la casse) ; **Ne contient pas** - exclure les noms d'étiquettes dont les valeurs contiennent la chaîne saisie (correspondance de sous-chaîne, insensible à la casse). Il existe deux types de calcul pour les conditions : **Et/Ou** - toutes les conditions doivent être remplies ; les conditions ayant le même nom d'étiquette sont regroupées avec la condition Ou ; **Ou** - une seule condition remplie suffit. Ce paramètre est disponible uniquement pour les ensembles de données *Modèles d'éléments* et n'est pas disponible lors de la configuration du widget sur un tableau de bord de modèle.
Remplacer l'hôte	Sélectionnez un widget compatible ou le sélecteur d'hôtes du tableau de bord comme source de données pour les hôtes. Ce paramètre n'est pas disponible lors de la configuration du widget sur un tableau de bord de modèle.
Marqueur	Choisissez le type et la taille du marqueur pour les points de données des hôtes dans l'ensemble de données.
Décalage temporel	Spécifiez un décalage temporel si nécessaire. Vous pouvez utiliser des suffixes temporels dans ce champ. Les valeurs négatives sont autorisées.
Fonction d'agrégation	Spécifiez la fonction d'agrégation à utiliser pour chaque élément de l'ensemble de données dans l'*intervalle d'agrégation* sélectionné : **min** - afficher la plus petite valeur ; **max** - afficher la plus grande valeur ; **avg** - afficher la valeur moyenne ; **sum** - afficher la somme des valeurs ; **count** - afficher le nombre de valeurs (si au moins une valeur existe sur chacun des deux axes) ; **first** - afficher la première valeur ; **last** - afficher la dernière valeur. L'agrégation permet d'afficher une valeur agrégée pour l'intervalle spécifié (5 minutes, une heure, un jour), au lieu de toutes les valeurs. Seules les données numériques peuvent être affichées pour *min*, *max*, *avg* et *sum*. Pour *count*, les données non numériques sont converties en données numériques.

Intervalle d'agrégation	Spécifiez l'intervalle d'agrégation des valeurs. Vous pouvez utiliser des suffixes temporels dans ce champ. Une valeur numérique sans suffixe est interprétée comme un nombre de secondes. Notez que si le widget est configuré pour afficher des données historiques basées sur les tendances (*Sélection des données historiques* est défini sur *Tendances* ou *Automatique*), il est recommandé d'utiliser un intervalle d'agrégation multiple d'une heure (par exemple, 3600, 60m, 1h, 3h, etc.). Les tendances stockent des valeurs agrégées par heure ; l'utilisation d'un intervalle d'agrégation qui n'est pas un multiple d'une heure (par exemple, 100s, 7min, 15min, 90min, etc.) peut donc produire des résultats difficiles à interpréter. La spécification d'un intervalle d'agrégation est obligatoire. L'intervalle d'agrégation détermine l'intervalle temporel et les coordonnées d'un point de données ; par exemple, si les éléments de l'axe X possèdent 4 valeurs et ceux de l'axe Y une valeur, les coordonnées reflètent l'agrégation de 4 valeurs sur X et d'une valeur sur Y dans l'intervalle spécifié.

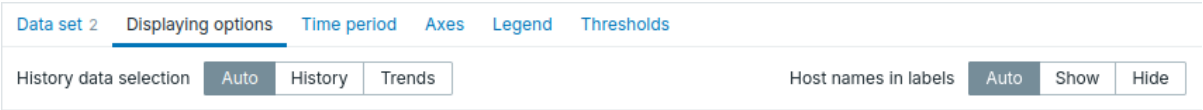
Détails de configuration des ensembles de données

Les ensembles de données existants sont affichés dans une liste. Vous pouvez réorganiser, développer/réduire, modifier les couleurs et cloner ces ensembles de données.

Pour plus d'informations, consultez les détails de configuration des ensembles de données dans le widget *Graph*. Ces détails s'appliquent également au widget *Scatter plot*.

Options d'affichage

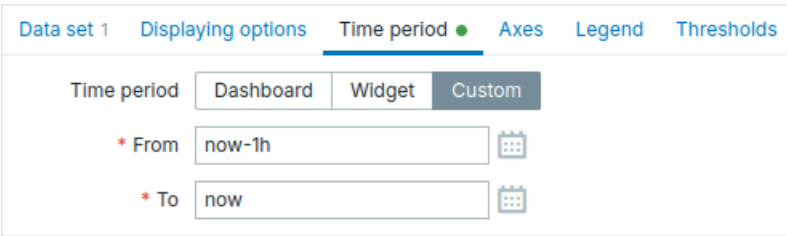
L'onglet **Options d'affichage** vous permet de définir la sélection des données d'historique :



Sélection des données d'historique	Définissez la source des données du graphique de dispersion : **Auto** - les données sont obtenues conformément à l'algorithme classique du graphique (par défaut). **Historique** - données provenant de l'historique. **Tendances** - données provenant des tendances.
Noms des hôtes dans les libellés	Choisissez d'afficher ou non les noms des hôtes dans la légende du graphique et dans l'infobulle qui apparaît lorsque vous survolez les données du graphique : **Auto** - les noms des hôtes sont affichés uniquement si plusieurs hôtes sont présents dans l'ensemble des jeux de données (par défaut). **Afficher** - les noms des hôtes sont affichés. **Masquer** - les noms des hôtes sont masqués. Ce paramètre n'est pas disponible lors de la configuration du widget sur un tableau de bord de modèle.

Période

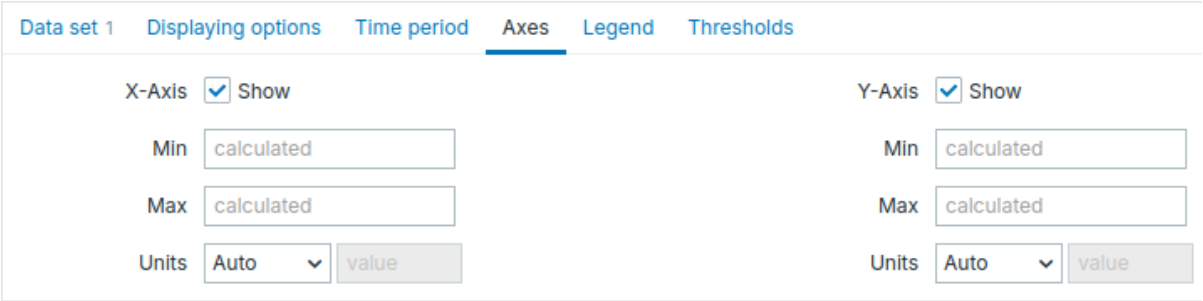
L'onglet **Période** vous permet de définir une période pour laquelle afficher les données dans le graphique de dispersion :



Période	Sélectionnez la source de données pour la période : **Tableau de bord** - utilisez le sélecteur de période du tableau de bord ; **Widget** - utilisez un widget compatible (défini dans le paramètre *Widget*) ; **Personnalisée** - utilisez une période personnalisée définie dans les paramètres *Du* et *Au* ; si elle est définie, une icône d'horloge s'affiche dans le coin supérieur droit du widget et indique l'heure définie au survol de la souris. Notez que, quelle que soit la configuration de la *Période* du widget, les widgets compatibles peuvent toujours l'utiliser comme source de données pour la période.
Widget	Saisissez ou sélectionnez un widget compatible comme source de données pour la période. Ce paramètre est disponible si *Période* est définie sur « Widget ».
Du	Saisissez ou sélectionnez le début de la période. La syntaxe des temps relatifs (`now`, `now/d`, `now/w-1w`, etc.) est prise en charge. Ce paramètre est disponible si *Période* est définie sur « Personnalisée ».
Au	Saisissez ou sélectionnez la fin de la période. La syntaxe des temps relatifs (`now`, `now/d`, `now/w-1w`, etc.) est prise en charge. Ce paramètre est disponible si *Période* est définie sur « Personnalisée ».

Axes

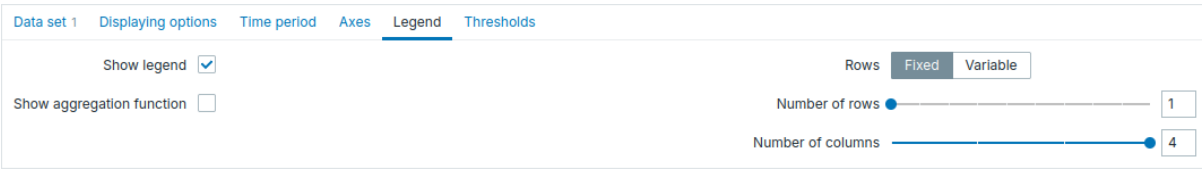
L'onglet **Axes** vous permet de personnaliser l'affichage des axes :



Axe X/Axe Y	Désélectionnez cette case pour masquer l'axe X/l'axe Y (sélectionnée par défaut).
Min	Définissez la valeur minimale de l'axe correspondant.
Max	Définissez la valeur maximale de l'axe correspondant.
Unités	Choisissez l'unité des valeurs des axes du graphique de dispersion dans la liste déroulante : **Auto** - les valeurs des axes sont affichées avec l'unité du premier élément de l'ensemble de données ; **Statique** - les valeurs des axes sont affichées avec l'unité indiquée dans le champ de saisie *value* ; si le champ est laissé vide, seules les valeurs numériques sont affichées.

Légende

L'onglet **Légende** vous permet de personnaliser la légende du nuage de points :

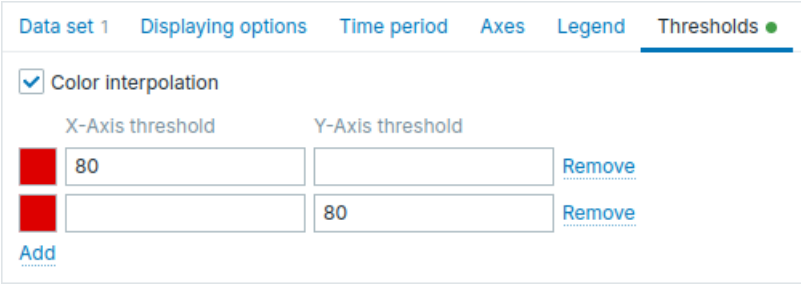


Afficher la légende	Désélectionnez cette case pour masquer la légende sur le nuage de points (sélectionnée par défaut).
Afficher la fonction d'agrégation	Sélectionnez cette case pour afficher la fonction d'agrégation dans la légende.
Lignes	Sélectionnez le mode d'affichage des lignes de la légende : **Fixe** - le nombre de lignes affichées est déterminé par la valeur du paramètre *Nombre de lignes* ; **Variable** - le nombre de lignes affichées est déterminé par le nombre d'éléments configurés, sans dépasser la valeur du paramètre *Nombre maximal de lignes*.
Nombre de lignes/ Nombre maximal de lignes	Si *Lignes* est défini sur « Fixe », définissez le nombre de lignes de légende à afficher (1-10). Si *Lignes* est défini sur « Variable », définissez le nombre maximal de lignes de légende à afficher (1-10).

Nombre de colonnes	Définissez le nombre de colonnes de légende à afficher (1-4). Ce paramètre est disponible si *Afficher min/moy/max* n'est pas sélectionné.

Seuils

L'onglet **Seuils** permet de configurer les seuils qui déterminent les couleurs des marqueurs des hôtes :

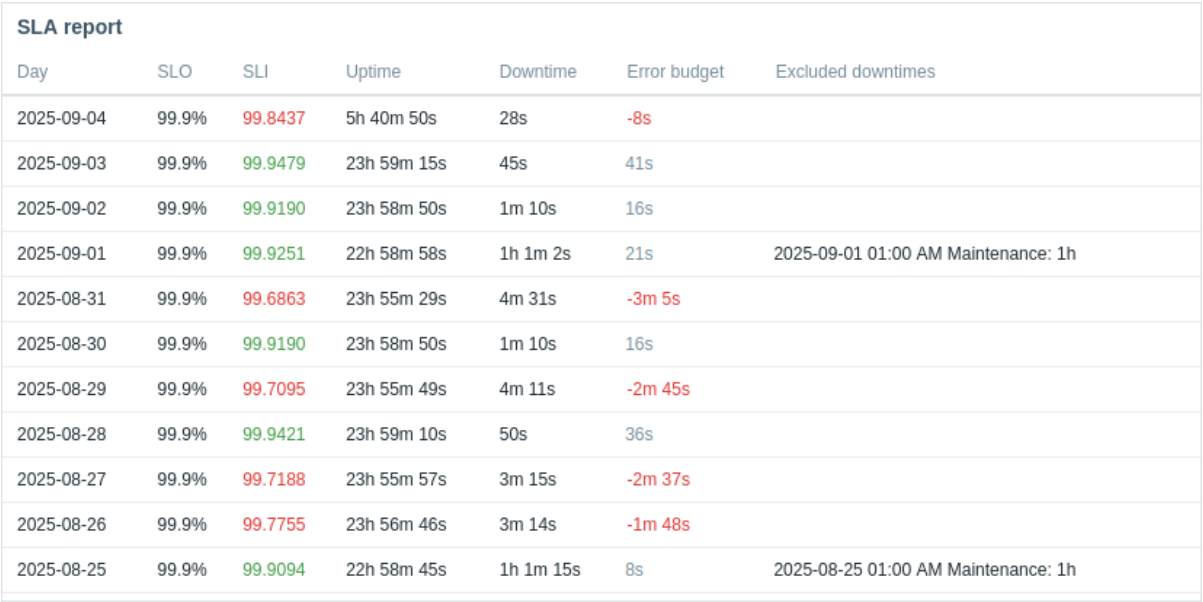


Interpolation des couleurs	Cochez la case pour activer une transition progressive entre les couleurs des seuils pour les marqueurs des hôtes. Ce paramètre s'applique si au moins deux seuils sont définis pour l'axe X, l'axe Y ou les deux axes simultanément.
Seuil de l'axe X/de l'axe Y	Cliquez sur *Ajouter* pour ajouter un seuil, sélectionnez une couleur de seuil dans le sélecteur de couleurs et indiquez une valeur numérique. Les seuils peuvent être appliqués à un seul axe, aux deux axes séparément ou aux deux axes simultanément. Par exemple, un seul seuil avec X=80 et Y=80 colorera les marqueurs lorsque les deux conditions sont remplies. La liste des seuils sera triée par ordre croissant lors de l'enregistrement. Les suffixes (par exemple, « 1d », « 2w », « 4K », « 8G ») sont pris en charge. Les mappages de valeurs sont pris en charge.

Rapport SLA

Aperçu

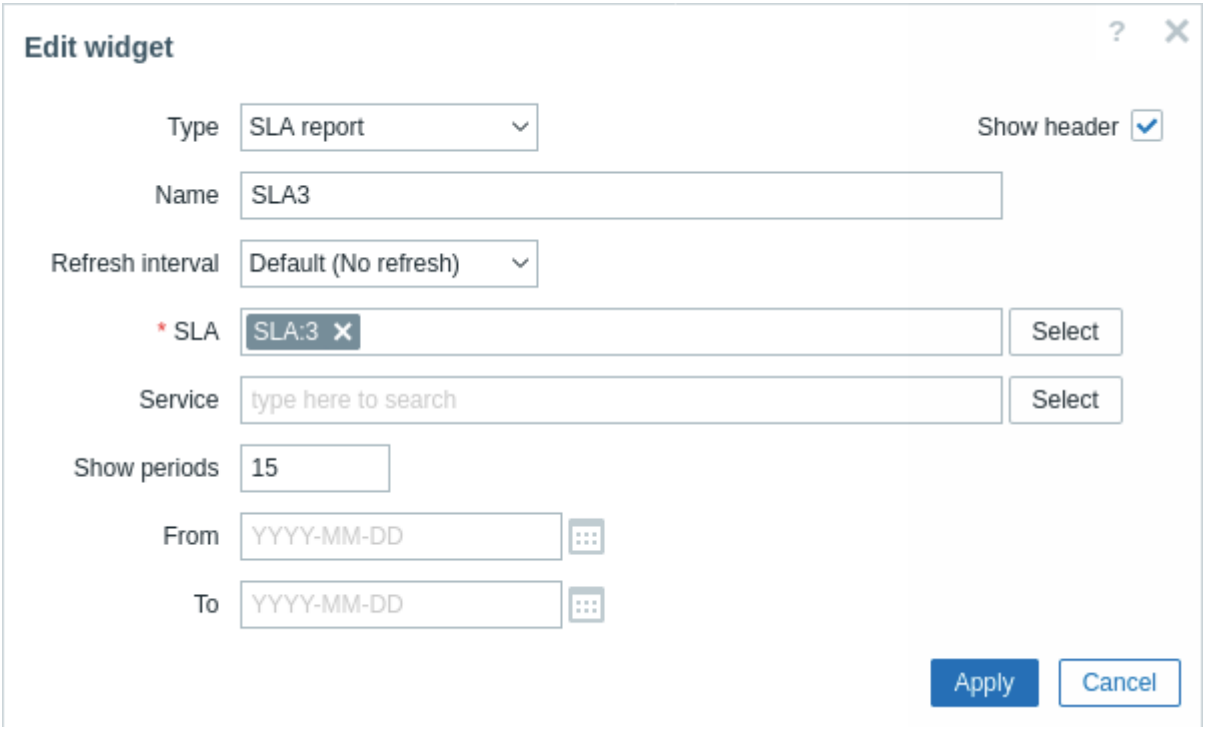
Le widget *Rapport SLA* affiche les rapports SLA, vous aidant à surveiller les performances des services par rapport aux objectifs définis.



Il affiche les mêmes données que *Services* > *Rapport SLA*.

Configuration

Pour configurer, sélectionnez *Rapport de SLA* comme type :



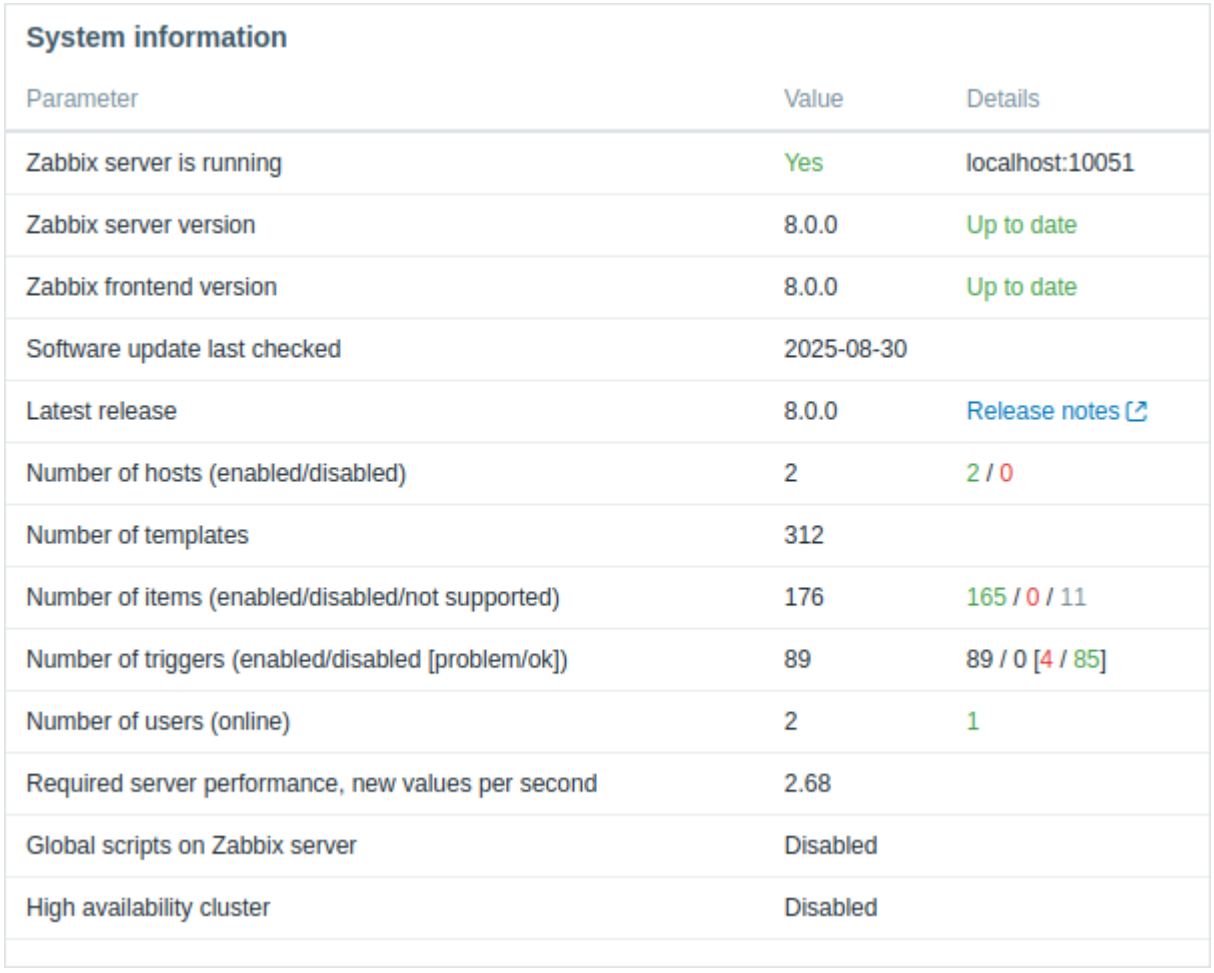
En plus des paramètres communs à tous les widgets, vous pouvez définir les options spécifiques suivantes :

SLA	Sélectionnez le SLA pour le rapport.
Service	Sélectionnez le service pour le rapport.
Afficher les périodes	Définissez combien de périodes seront affichées dans le widget (20 par défaut, 100 maximum).
De	Sélectionnez la date de début du rapport. Les dates relatives sont prises en charge : `now`, `now/d`, `now/w-1w`, etc. ; modificateurs de date pris en charge : d, w, M, y.
À	Sélectionnez la date de fin du rapport. Les dates relatives sont prises en charge : `now`, `now/d`, `now/w-1w`, etc. ; modificateurs de date pris en charge : d, w, M, y.

Informations système

Vue d'ensemble

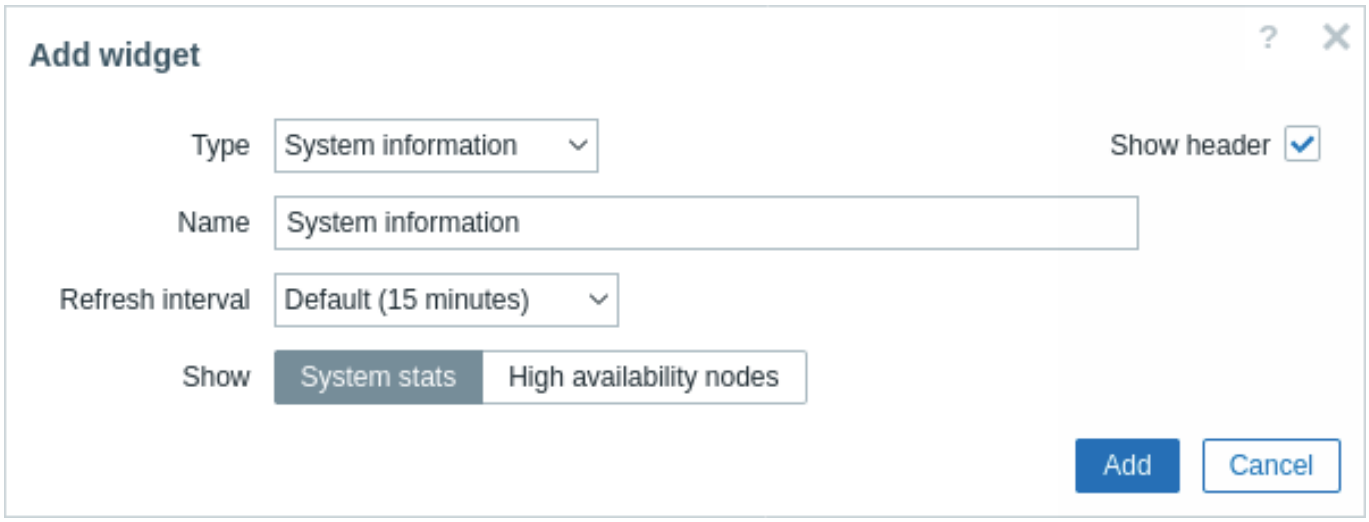
Le widget *System information* affiche un résumé des données clés du serveur Zabbix et du système, et, lorsque la haute disponibilité est activée, des détails sur les nœuds de haute disponibilité.



Il affiche les mêmes données que *Reports* > *System information*, à l'exception de *Zabbix server ID*.

Configuration

Pour configurer le widget, sélectionnez *Informations système* comme type :



En plus des paramètres communs à tous les widgets, vous pouvez définir les options spécifiques suivantes :

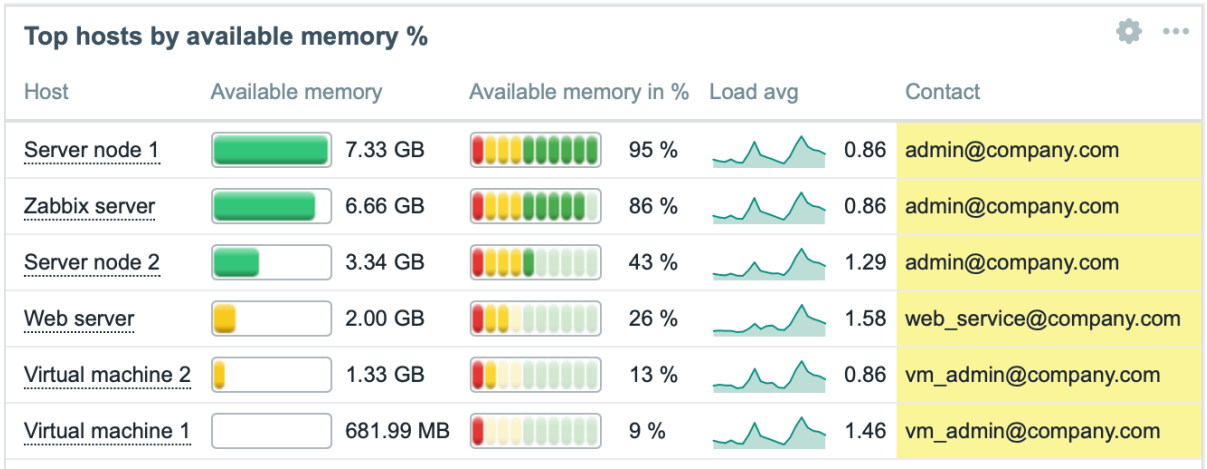
Afficher	Sélectionnez les éléments à afficher : **Statistiques système** - affiche un résumé des principales données du serveur Zabbix et du système ; **Nœuds de haute disponibilité** - affiche l'état des nœuds de haute disponibilité (si un cluster de haute disponibilité est activé).

Afficher les détails de la vérification des mises à jour logicielles	Cochez cette case pour afficher les détails de la vérification des mises à jour logicielles de Zabbix. Cette option est disponible uniquement si la vérification des mises à jour logicielles est activée dans la configuration du serveur Zabbix et si « Statistiques système » est sélectionné dans le champ *Afficher*.

Principaux hôtes

Vue d'ensemble

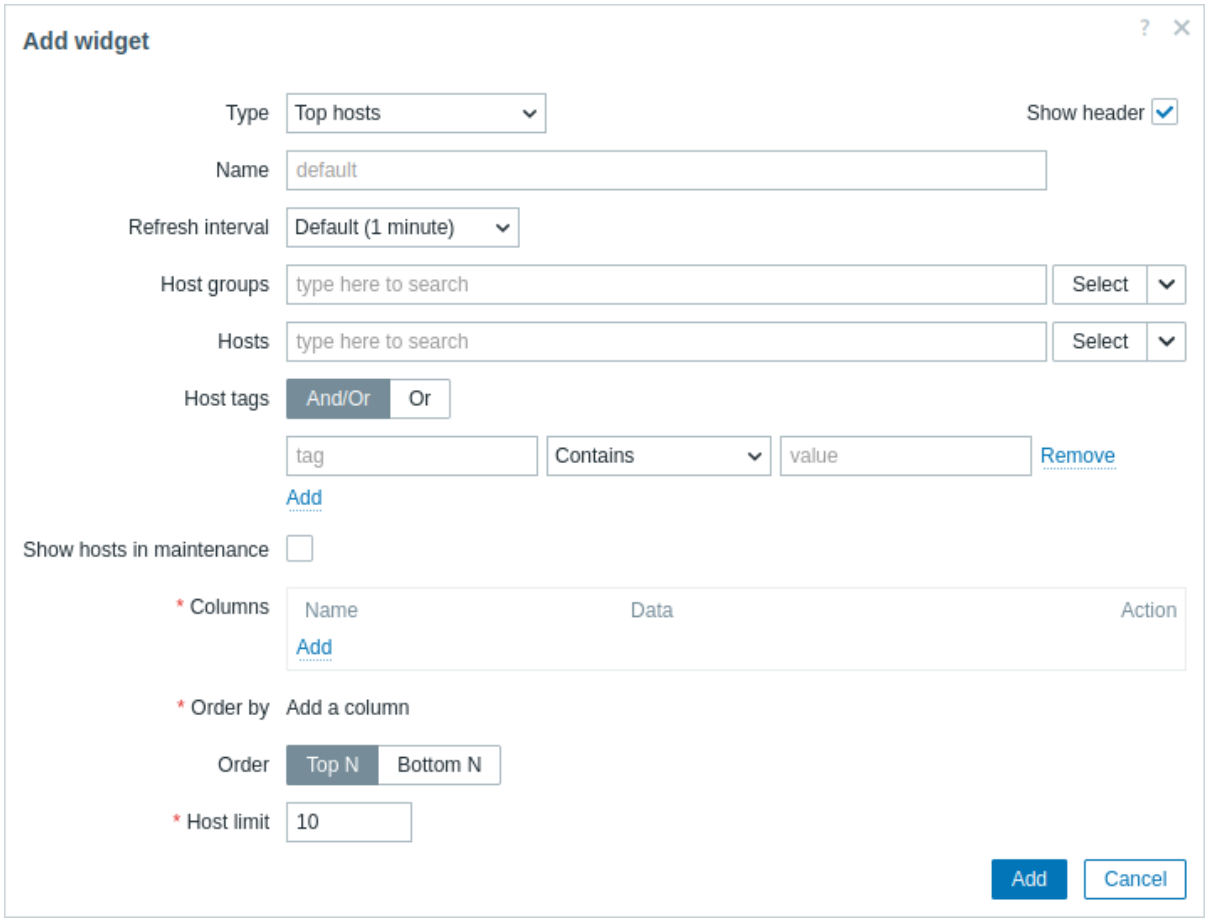
Le widget *Top hosts* affiche les valeurs les plus élevées/les plus basses pour les éléments sélectionnés de plusieurs hôtes.



Le widget prend en charge l'affichage simultané de jusqu'à 1000 hôtes.

Configuration

Pour configurer, sélectionnez *Top hosts* comme type :



En plus des paramètres communs à tous les widgets, vous pouvez définir les options spécifiques suivantes :

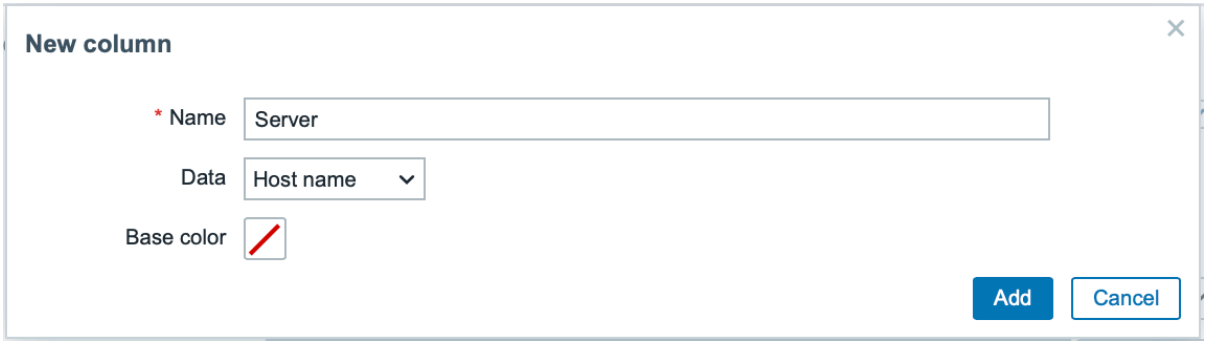
Groupes d'hôtes	Sélectionnez les groupes d'hôtes à afficher dans le widget. Vous pouvez également sélectionner un widget compatible comme source de données pour les groupes d'hôtes. Ce champ prend en charge l'autocomplétion ; ainsi, lorsque vous commencez à saisir le nom d'un groupe, une liste déroulante des groupes correspondants s'affiche. Ce paramètre n'est pas disponible lors de la configuration du widget sur un tableau de bord de modèle.
Hôtes	Sélectionnez les hôtes à afficher dans le widget. Vous pouvez également sélectionner un widget compatible ou le tableau de bord comme source de données pour les hôtes. Ce champ prend en charge l'autocomplétion ; ainsi, lorsque vous commencez à saisir le nom d'un hôte, une liste déroulante des hôtes correspondants s'affiche. Ce paramètre n'est pas disponible lors de la configuration du widget sur un tableau de bord de modèle.
Tags d'hôte	Spécifiez des tags pour limiter le nombre d'hôtes affichés dans le widget. Il est possible d'inclure ou d'exclure des tags et des valeurs de tag spécifiques. Plusieurs conditions peuvent être définies. La correspondance des noms de tag est toujours sensible à la casse. Plusieurs opérateurs sont disponibles pour chaque condition : **Existe** - inclure les noms de tag spécifiés ; **Égal à** - inclure les noms de tag et les valeurs spécifiés (sensible à la casse) ; **Contient** - inclure les noms de tag spécifiés lorsque les valeurs de tag contiennent la chaîne saisie (correspondance de sous-chaîne, insensible à la casse) ; **N'existe pas** - exclure les noms de tag spécifiés ; **N'est pas égal à** - exclure les noms de tag et les valeurs spécifiés (sensible à la casse) ; **Ne contient pas** - exclure les noms de tag spécifiés lorsque les valeurs de tag contiennent la chaîne saisie (correspondance de sous-chaîne, insensible à la casse). Deux types de calcul sont disponibles pour les conditions : **Et/Ou** - toutes les conditions doivent être remplies ; les conditions ayant le même nom de tag seront regroupées par la condition Ou ; **Ou** - il suffit qu'une seule condition soit remplie. Ce paramètre n'est pas disponible lors de la configuration du widget sur un tableau de bord de modèle.
Afficher les hôtes en maintenance	Cochez cette case pour que les hôtes en maintenance soient également affichés (dans ce cas, l'icône de maintenance sera affichée à côté du nom de l'hôte). Décochée par défaut.
Colonnes	Ajoutez des colonnes de données à afficher. L'ordre des colonnes détermine leur affichage de gauche à droite. Les colonnes peuvent être réorganisées en les faisant glisser vers le haut ou vers le bas à l'aide de la poignée située avant le nom de la colonne.
Trier par	Spécifiez la colonne de la liste *Colonnes* définie à utiliser pour le tri *Top N* ou *Bottom N*.
Ordre	Spécifiez l'ordre des lignes : **Top N** - par ordre décroissant selon la valeur agrégée de *Trier par* ; **Bottom N** - par ordre croissant selon la valeur agrégée de *Trier par*.
Limite d'hôtes	Nombre de lignes d'hôtes à afficher (1-1000). Ce paramètre n'est pas disponible lors de la configuration du widget sur un tableau de bord de modèle.

Configuration des colonnes

Il est possible d'ajouter des colonnes de trois types de données : nom d'hôte, valeur d'élément ou texte. La liste des paramètres de colonne disponibles dépend du type de données de la colonne et, pour le type valeur d'élément, du format de la valeur.

Colonne Nom d'hôte

La colonne Nom d'hôte est utilisée pour afficher le nom de l'hôte.

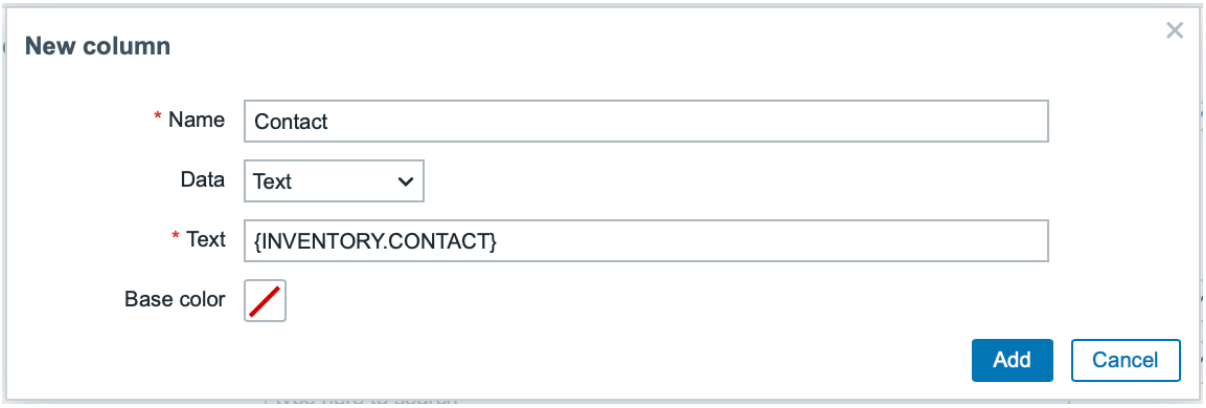


Paramètres pris en charge :

Name	Nom de la colonne.
Data	Type de données à afficher dans la colonne, sélectionnez **Host name**.
Base color	Couleur d'arrière-plan de la colonne.

Colonne de texte

La colonne de texte est utilisée pour afficher toute chaîne de texte spécifiée.

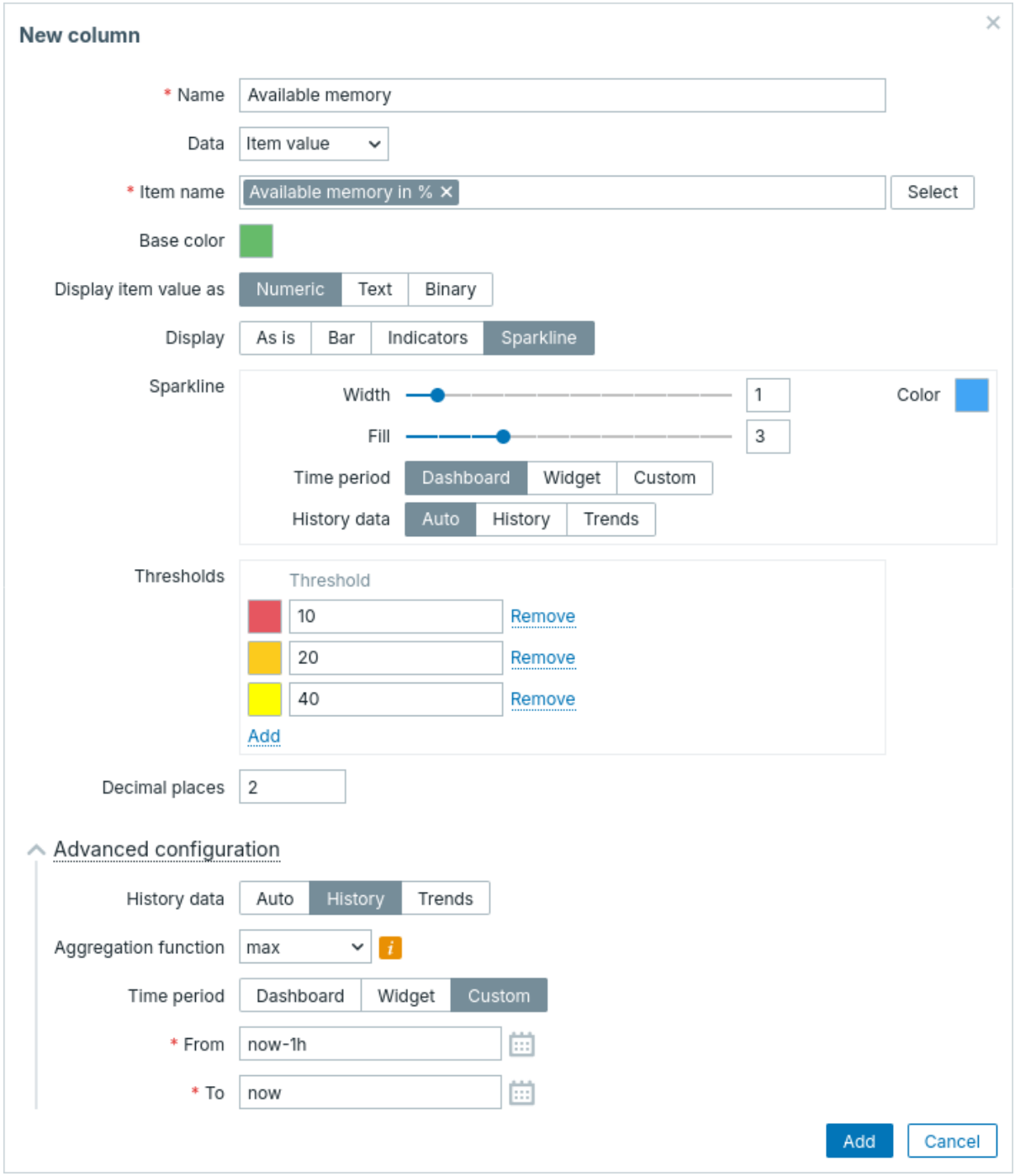


Paramètres pris en charge :

Name	Nom de la colonne.
Data	Type de données à afficher dans la colonne, sélectionnez **Text**.
Base color	Couleur d'arrière-plan de la colonne.
Text	Saisissez la chaîne à afficher. Peut contenir des macros d'hôte et d'inventaire.

Colonne de valeur de l'élément

La colonne de valeur de l'élément permet d'afficher la valeur de l'élément spécifié ; elle prend en charge plusieurs formats et options d'affichage des valeurs.



Paramètres pris en charge :

Name	Nom de la colonne.
Data	Type de données à afficher dans la colonne, sélectionnez **Item value**.
Item name	Sélectionnez un élément ; son nom est utilisé pour rechercher et afficher les éléments portant le même nom sur tous les hôtes sélectionnés. Si un hôte possède plusieurs éléments de ce type, le widget affiche l'élément dont la clé vient en premier dans l'ordre alphabétique (par exemple, *CPU utilization* avec la clé `proc.cpu.util` plutôt que `system.cpu.util`). Lors de la configuration du widget sur un tableau de bord de modèle, seuls les éléments configurés sur le modèle peuvent être sélectionnés.
Base color	Couleur d'arrière-plan de la colonne ; couleur de remplissage si les données *Item value* sont affichées sous forme de barre ou d'indicateurs. La couleur par défaut peut être remplacée par une couleur personnalisée si la valeur de l'élément dépasse l'un des seuils spécifiés.
Display item value as	Format d'affichage de la valeur de l'élément : **Numeric**, **Text** ou **Binary**. L'option sélectionnée détermine les paramètres supplémentaires disponibles. Consultez la liste des paramètres correspondant à chaque format ci-dessous.
Advanced configuration	

History data	Prendre les données de l'historique ou des tendances : **Auto** - sélection automatique ; **History** - prendre les données de l'historique ; **Trends** - prendre les données des tendances. Ce paramètre est disponible uniquement pour les valeurs d'éléments numériques. Pour les valeurs textuelles et binaires, les données sont toujours prises dans l'historique.
Aggregation function	Spécifiez la fonction d'agrégation à utiliser dans la *Time period* sélectionnée : **min** - afficher la plus petite valeur ; **max** - afficher la plus grande valeur ; **avg** - afficher la valeur moyenne ; **count** - afficher le nombre de valeurs ; **sum** - afficher la somme des valeurs ; **first** - afficher la première valeur ; **last** - afficher la dernière valeur. Si **not used** est sélectionné, la valeur la plus récente est affichée (sans agrégation). L'agrégation permet d'afficher une valeur agrégée pour l'intervalle spécifié (5 minutes, une heure, un jour), au lieu de la valeur la plus récente. Seules les données numériques peuvent être affichées pour *min*, *max*, *avg* et *sum*. Pour *count*, les données non numériques sont converties en données numériques.
Time period	Spécifiez la période à utiliser pour agréger les valeurs. Sélectionnez la source de données de la période : **Dashboard** - utiliser le sélecteur de période du tableau de bord ; **Widget** - utiliser un widget compatible (défini dans le paramètre *Widget*) ; **Custom** - utiliser une période personnalisée définie dans les paramètres *From* et *To* ; si elle est définie, une icône d'horloge s'affiche dans l'angle supérieur droit du widget et indique la période définie au survol de la souris. Notez que, quelle que soit la configuration *Time period* du widget, les widgets compatibles peuvent toujours l'utiliser comme source de données pour la période. Ce paramètre n'est pas disponible si *Aggregation function* est défini sur « not used ».
Widget	Saisissez ou sélectionnez un widget compatible comme source de données pour la période. Ce paramètre est disponible si *Time period* est défini sur « Widget ».
From	Saisissez ou sélectionnez le début de la période. La syntaxe des périodes relatives (`now`, `now/d`, `now/w-1w`, etc.) est prise en charge. Ce paramètre est disponible si *Time period* est défini sur « Custom ».
To	Saisissez ou sélectionnez la fin de la période. La syntaxe des périodes relatives (`now`, `now/d`, `now/w-1w`, etc.) est prise en charge. Ce paramètre est disponible si *Time period* est défini sur « Custom ».

Paramètres de **Numeric item value** :

Display	Définissez le mode d'affichage de la valeur : **As is** - sous forme de texte standard ; **Bar** - sous forme de barre pleine avec remplissage coloré ; **Indicators** - sous forme de barre segmentée avec remplissage coloré ; **Sparkline** - mini-graphique linéaire.
Thresholds	Spécifiez les valeurs seuils à partir desquelles la couleur d'arrière-plan ou de remplissage doit changer. La liste est triée par ordre croissant lors de l'enregistrement. Pour les graphiques sparkline, les seuils s'appliquent uniquement à la dernière valeur de l'élément.
Decimal places	Spécifiez le nombre de décimales à afficher avec la valeur.
Bar/Indicators	
Min	Valeur minimale.
Max	Valeur maximale.
Sparkline	
Width	Définissez l'épaisseur de la ligne du graphique à l'aide du curseur ou en saisissant manuellement une valeur comprise entre 0 et 10.
Color	Sélectionnez la couleur de la ligne et du remplissage.
Fill	Définissez le niveau de transparence de la couleur de remplissage à l'aide du curseur ou en saisissant manuellement une valeur comprise entre 0 et 10.

Time period	Spécifiez la période dont les valeurs doivent être incluses dans le graphique sparkline. Sélectionnez la source de données de la période : **Dashboard** - utiliser le sélecteur de période du tableau de bord **Widget** - utiliser un widget compatible (défini dans le paramètre *Widget*) ; **Custom** - utiliser une période personnalisée définie dans les paramètres *From* et *To* ; si elle est définie, une icône d'horloge s'affiche dans l'angle supérieur droit du widget et indique la période définie au survol de la souris. Notez que, quelle que soit la configuration *Time period* du widget, les widgets compatibles peuvent toujours l'utiliser comme source de données pour la période.
Widget	Saisissez ou sélectionnez un widget compatible comme source de données pour la période. Ce paramètre est disponible si *Time period* est défini sur « Widget ».
From	Saisissez ou sélectionnez le début de la période. La syntaxe des périodes relatives (`now`, `now/d`, `now/w-1w`, etc.) est prise en charge. Ce paramètre est disponible si *Time period* est défini sur « Custom ».
To	Saisissez ou sélectionnez la fin de la période. La syntaxe des périodes relatives (`now`, `now/d`, `now/w-1w`, etc.) est prise en charge. Ce paramètre est disponible si *Time period* est défini sur « Custom ».
History data	Prendre les données de l'historique ou des tendances : **Auto** - sélection automatique ; **History** - prendre les données de l'historique ; **Trends** - prendre les données des tendances.

Paramètres de **Text item value** :

Highlights	Spécifiez les expressions régulières dont la correspondance doit entraîner la modification de la couleur d'arrière-plan ou de remplissage.

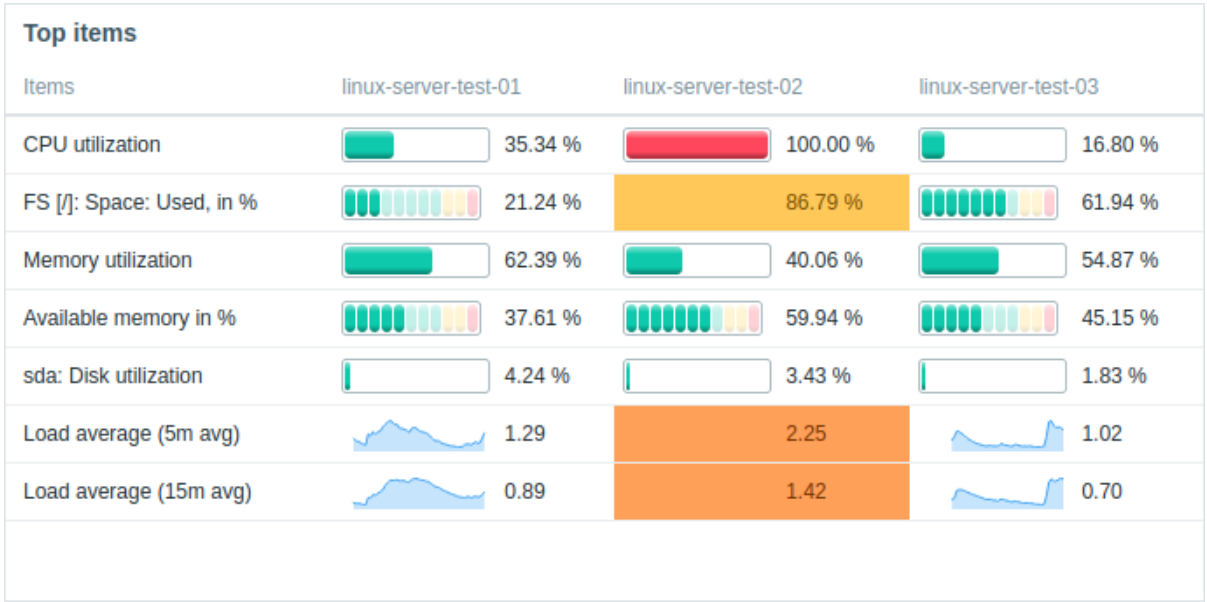
Paramètres de **Binary item value** :

Show thumbnail	Indiquez s'il faut créer et afficher une miniature de l'image contenant les données binaires ou afficher un lien hypertexte *Show* menant à l'image en taille réelle dans la colonne de valeur.

Principaux éléments

Aperçu

Le widget *Top items* affiche les valeurs les plus élevées/les plus basses pour les éléments sélectionnés, offrant un aperçu rapide de leurs performances.

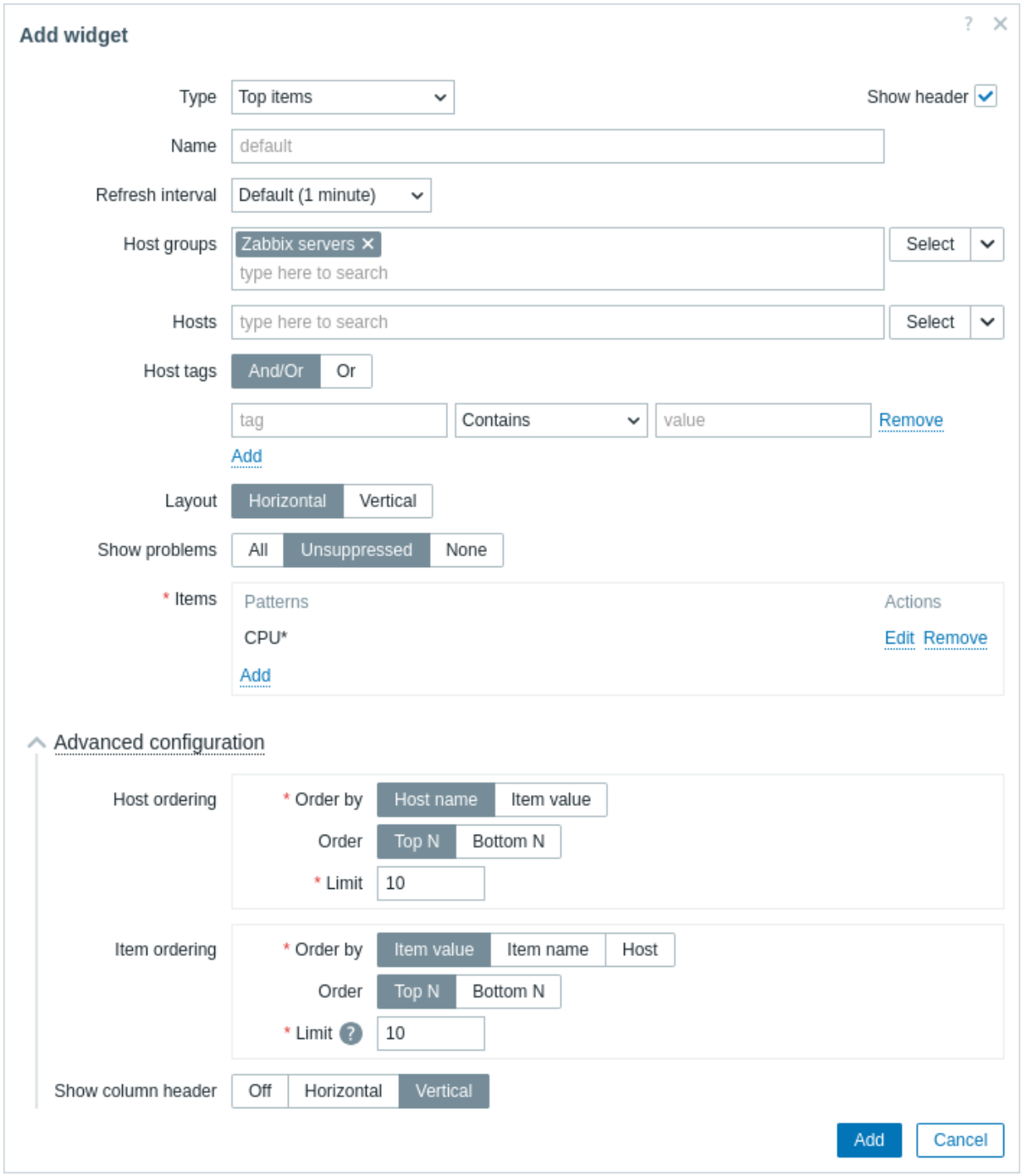


Par défaut, le widget affiche les valeurs des dernières 24 heures. Cette période peut être modifiée dans *Administration* > *General* > *GUI* (*Max history display period*).

Un clic sur la valeur de l'élément ouvre le menu de l'élément.

Configuration

Pour configurer, sélectionnez *Top items* comme type :



En plus des paramètres communs à tous les widgets, vous pouvez définir les options spécifiques suivantes :

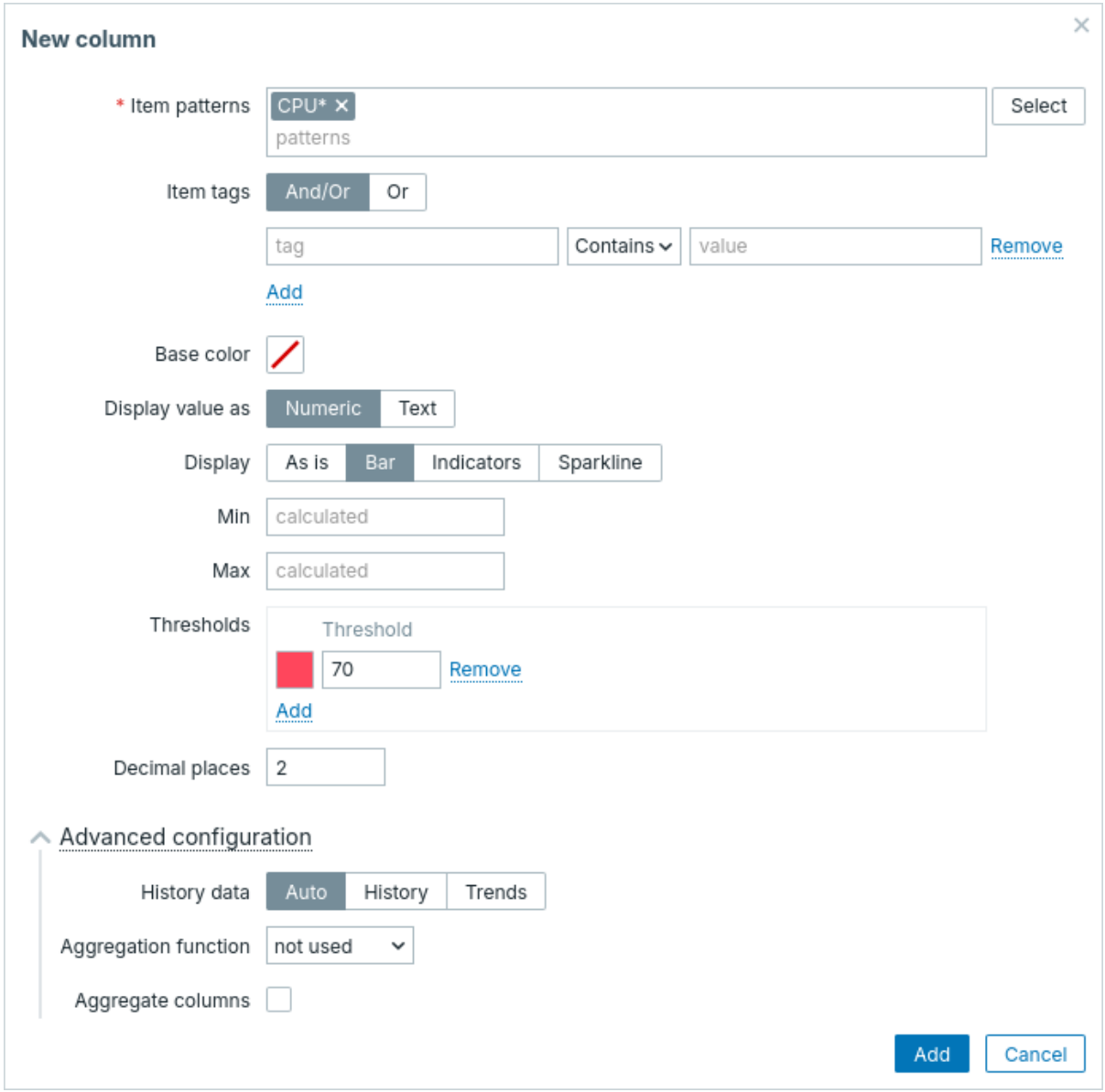
Groupes d'hôtes	Sélectionnez des groupes d'hôtes. Vous pouvez également sélectionner un widget compatible comme source de données pour les groupes d'hôtes. Ce champ est à saisie semi-automatique, donc commencer à taper le nom d'un groupe proposera une liste déroulante des groupes correspondants. Ce paramètre n'est pas disponible lors de la configuration du widget sur un tableau de bord de modèle.

Hôtes	Sélectionnez des hôtes. Vous pouvez également sélectionner un widget compatible ou le tableau de bord comme source de données pour les hôtes. Ce champ est à saisie semi-automatique, donc commencer à taper le nom d'un hôte proposera une liste déroulante des hôtes correspondants. Ce paramètre n'est pas disponible lors de la configuration du widget sur un tableau de bord de modèle.
Balises d'hôte	Spécifiez des balises pour limiter le nombre de données d'hôte affichées dans le widget. Il est possible d'inclure ainsi que d'exclure des balises et des valeurs de balise spécifiques. Plusieurs conditions peuvent être définies. La correspondance du nom de balise est toujours sensible à la casse. Plusieurs opérateurs sont disponibles pour chaque condition : **Exists** - inclure les noms de balise spécifiés ; **Equals** - inclure les noms et valeurs de balise spécifiés (sensible à la casse) ; **Contains** - inclure les noms de balise spécifiés lorsque les valeurs de balise contiennent la chaîne saisie (correspondance de sous-chaîne, insensible à la casse) ; **Does not exist** - exclure les noms de balise spécifiés ; **Does not equal** - exclure les noms et valeurs de balise spécifiés (sensible à la casse) ; **Does not contain** - exclure les noms de balise spécifiés lorsque les valeurs de balise contiennent la chaîne saisie (correspondance de sous-chaîne, insensible à la casse). Il existe deux types de calcul pour les conditions : **And/Or** - toutes les conditions doivent être remplies, les conditions ayant le même nom de balise seront regroupées par la condition *Or* ; **Or** - une seule condition remplie suffit.
Disposition	Sélectionnez l'option de disposition : **Horizontal** - les noms d'hôte seront affichés à gauche ; **Vertical** - les noms d'hôte seront affichés en haut.
Afficher les problèmes	Filtrez les problèmes à afficher en fonction de leur statut (tous, non supprimés, aucun). Les problèmes sont indiqués par des valeurs d'élément colorées, remplaçant les paramètres d'affichage de la colonne d'élément. Les couleurs sont basées sur la gravité du problème, qui peut être personnalisée dans *Administration* > *Général* > *Options d'affichage des déclencheurs*. Les gravités des problèmes peuvent être modifiées lors de la mise à jour des problèmes.
Éléments	Ajoutez des motifs d'élément ou des éléments spécifiques à afficher (voir Configuration des colonnes).
Configuration avancée	
Ordre des hôtes	Sélectionnez les options de tri pour la colonne/ligne des hôtes.
Trier par	Trier les hôtes par : **Host name** - les hôtes seront triés par nom d'hôte ; **Item value** - les hôtes seront triés par la valeur des éléments sélectionnés.
Ordre	Sélectionnez si vous souhaitez afficher les valeurs les plus élevées ou les plus faibles : **TopN** - les N valeurs les plus élevées ; **BottomN** - les N valeurs les plus faibles. La valeur de N est sélectionnée dans le champ *Limit*.
Limite	Saisissez la limite pour les hôtes affichables (plage 1-1000 ; valeur par défaut 10). Cette valeur deviendra la valeur de N dans le champ *Ordre*.
Ordre des éléments	Sélectionnez les options de tri pour la colonne/ligne des éléments.
Trier par	Trier les éléments par : **Item value** - les éléments seront triés par valeur d'élément ; **Item name** - les éléments seront triés par nom d'élément ; **Host** - les éléments seront triés par le motif d'hôte sélectionné.

Ordre	Sélectionnez si vous souhaitez afficher les valeurs les plus élevées ou les plus faibles : **TopN** - les N valeurs les plus élevées ; **BottomN** - les N valeurs les plus faibles. La valeur de N est sélectionnée dans le champ *Limit*.
Limite	Saisissez la limite pour les éléments affichables (plage 1-1000 ; valeur par défaut 10). Cette valeur deviendra la valeur de N dans le champ *Ordre*.
Afficher l'en-tête de colonne	Sélectionnez les options d'affichage de l'en-tête de colonne : **Off** - n'affiche pas l'en-tête de colonne ; **Horizontal** - affiche le texte horizontalement dans l'en-tête ; **Vertical** - affiche le texte verticalement dans l'en-tête.

Configuration des colonnes

Pour configurer les colonnes d'éléments, cliquez sur *Ajouter* dans le paramètre *Éléments* :



Paramètres communs des colonnes :

Modèles d'éléments	Spécifiez un ou plusieurs modèles d'éléments à l'aide du caractère générique. Vous pouvez également sélectionner les éléments. Lors de la configuration du widget sur un tableau de bord de modèle, seuls les éléments configurés sur le modèle peuvent être sélectionnés.

Tags d'élément	Spécifiez des tags pour limiter le nombre de données d'élément affichées dans le widget. Il est possible d'inclure ou d'exclure des tags et des valeurs de tag spécifiques. Plusieurs conditions peuvent être définies. La correspondance des noms de tag est toujours sensible à la casse. Plusieurs opérateurs sont disponibles pour chaque condition : **Existe** - inclure les noms de tag spécifiés ; **Égal à** - inclure les noms de tag et les valeurs spécifiés (sensible à la casse) ; **Contient** - inclure les noms de tag spécifiés lorsque les valeurs de tag contiennent la chaîne saisie (correspondance de sous-chaîne, insensible à la casse) ; **N'existe pas** - exclure les noms de tag spécifiés ; **N'est pas égal à** - exclure les noms de tag et les valeurs spécifiés (sensible à la casse) ; **Ne contient pas** - exclure les noms de tag spécifiés lorsque les valeurs de tag contiennent la chaîne saisie (correspondance de sous-chaîne, insensible à la casse). Deux types de calcul sont disponibles pour les conditions : **Et/Ou** - toutes les conditions doivent être remplies ; les conditions ayant le même nom de tag seront regroupées par la condition *Ou* ; **Ou** - il suffit qu'une seule condition soit remplie.
Couleur de base	Sélectionnez la couleur d'arrière-plan de la colonne ou la couleur de remplissage si *Affichage* est défini sur « Bar » ou « Indicators ». Notez que la couleur de base peut être remplacée par les couleurs de seuil ou de mise en évidence.
Afficher la valeur comme	Format d'affichage de la valeur de l'élément : **Numérique** ou **Texte**. L'option sélectionnée détermine quels paramètres supplémentaires seront disponibles. Reportez-vous à la liste des paramètres pour chaque format ci-dessous.

Paramètres de colonne pour les valeurs numériques :

Affichage	Définissez la manière dont la valeur doit être affichée : **Tel quel** - sous forme de texte standard ; **Barre** - sous forme de barre pleine remplie de couleur ; **Indicateurs** - sous forme de barre segmentée remplie de couleur ; **Sparkline** - mini graphique linéaire.
Min	Valeur minimale pour l'affichage des barres/indicateurs.
Max	Valeur maximale pour l'affichage des barres/indicateurs.
Largeur	Définissez l'épaisseur de la ligne du graphique à l'aide du curseur ou en saisissant manuellement une valeur comprise entre 0 et 10. Ce paramètre s'applique uniquement à l'affichage Sparkline.
Remplissage	Définissez le niveau de transparence de la couleur de remplissage à l'aide du curseur ou en saisissant manuellement une valeur comprise entre 0 et 10. Ce paramètre s'applique uniquement à l'affichage Sparkline.
Couleur	Sélectionnez la couleur de la ligne et du remplissage. Ce paramètre s'applique uniquement à l'affichage Sparkline.
Consultez la section *Configuration avancée* pour obtenir une description des champs liés à la période et à la sélection des données d'historique.	
Seuils	Indiquez les valeurs seuils à partir desquelles la couleur d'arrière-plan/de remplissage doit changer. La liste sera triée par ordre croissant lors de l'enregistrement. Pour les sparklines, les seuils s'appliquent uniquement à la dernière valeur de l'élément.
Décimales ***Configuration avancée***	Indiquez le nombre de décimales à afficher avec la valeur.

Données d'historique	Utilisez les données de l'historique ou des tendances : **Automatique** - sélection automatique ; **Historique** - utiliser les données d'historique ; **Tendances** - utiliser les données de tendance. Ce paramètre est disponible uniquement pour les valeurs numériques des éléments. Pour les valeurs textuelles, les données seront toujours extraites de l'historique.
Fonction d'agrégation	Indiquez la fonction d'agrégation à utiliser dans la *Période* sélectionnée : **min** - afficher la plus petite valeur ; **max** - afficher la plus grande valeur ; **avg** - afficher la valeur moyenne ; **count** - afficher le nombre de valeurs ; **sum** - afficher la somme des valeurs ; **first** - afficher la première valeur ; **last** - afficher la dernière valeur. Si **non utilisée** est sélectionné, la valeur la plus récente est affichée (sans agrégation). L'agrégation permet d'afficher une valeur agrégée pour l'intervalle spécifié (5 minutes, une heure, un jour), au lieu de la valeur la plus récente. Seules les données numériques peuvent être affichées pour *min*, *max*, *avg* et *sum*. Pour *count*, les données non numériques seront converties en données numériques.
Période	Indiquez la période à utiliser pour agréger les valeurs. Sélectionnez la source de données pour la période : **Tableau de bord** - utiliser le sélecteur de période du tableau de bord ; **Widget** - utiliser un widget compatible (défini dans le paramètre *Widget*) ; **Personnalisée** - utiliser une période personnalisée définie dans les paramètres *De* et *À* ; si elle est définie, une icône d'horloge sera affichée dans le coin supérieur droit du widget et indiquera l'heure définie au survol de la souris. Notez que, quelle que soit la configuration de la *Période* du widget, les widgets compatibles peuvent toujours l'utiliser comme source de données pour la période. Ce paramètre n'est pas disponible si la *Fonction d'agrégation* est définie sur « non utilisée ».
Widget	Saisissez ou sélectionnez un widget compatible comme source de données pour la période. Ce paramètre est disponible si *Période* est défini sur « Widget ».
De	Saisissez ou sélectionnez le début de la période. La syntaxe des périodes relatives (`now`, `now/d`, `now/w-1w`, etc.) est prise en charge. Ce paramètre est disponible si *Période* est défini sur « Personnalisée ».
À	Saisissez ou sélectionnez la fin de la période. La syntaxe des périodes relatives (`now`, `now/d`, `now/w-1w`, etc.) est prise en charge. Ce paramètre est disponible si *Période* est défini sur « Personnalisée ».
Agrégater les colonnes	Lorsque cette option est cochée, regroupez tous les éléments correspondant aux modèles d'éléments de la colonne en une seule colonne/ligne combinée. Les valeurs des éléments correspondants seront agrégées à l'aide de la *Fonction d'agrégation des colonnes* sélectionnée et affichées sous le *Nom de la colonne combinée*. Valeur par défaut : désactivée (non cochée).Ce paramètre s'affiche uniquement lorsque *Afficher la valeur en tant que* est défini sur **Numérique**.
Fonction d'agrégation des colonnes	Choisissez la manière dont les valeurs des modèles d'éléments regroupés sont combinées. Options disponibles : **min**, **max**, **avg**, **count**, **sum** (par défaut). Ce paramètre s'affiche uniquement lorsque *Agrégater les colonnes* est cochée.
Nom de la colonne combinée	Saisissez le nom d'affichage de la colonne agrégée. Ce champ est obligatoire lorsque *Agrégater les colonnes* est cochée et devient l'en-tête de colonne des valeurs combinées.

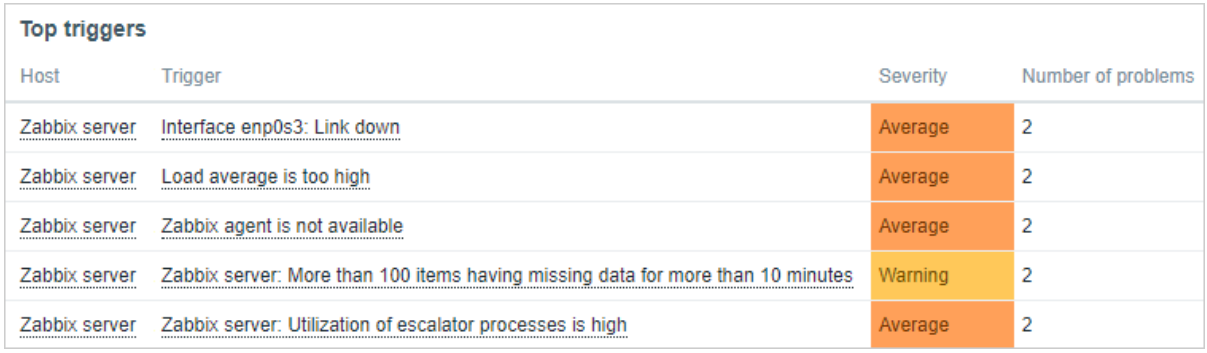
Paramètres de colonne pour les valeurs textuelles :

Mises en évidence	Indiquez les expressions régulières dont la correspondance doit entraîner la modification de la couleur d'arrière-plan/de remplissage.

Principaux déclencheurs

Vue d'ensemble

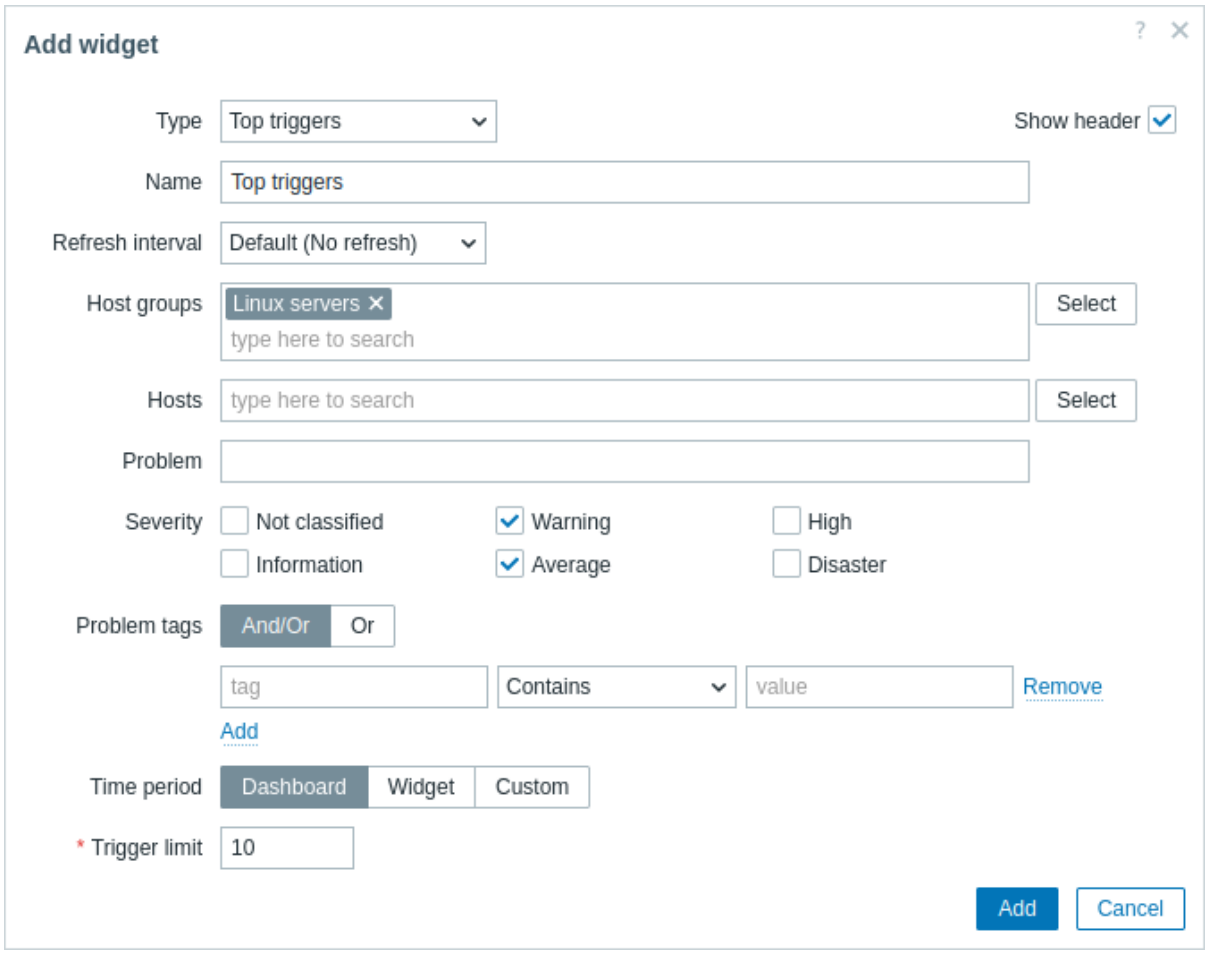
Le widget *Top triggers* affiche les déclencheurs ayant le plus grand nombre de problèmes.



Il affiche les mêmes données que *Rapports > Top 100 triggers* et peut afficher jusqu'à 1000 enregistrements.

Configuration

Pour configurer, sélectionnez *Top triggers* comme type de widget :



En plus des paramètres communs à tous les widgets, vous pouvez définir les options spécifiques suivantes :

Host groups	Sélectionnez les groupes d'hôtes dont les problèmes doivent être affichés dans le widget. Ce champ prend en charge l'autocomplétion ; ainsi, lorsque vous commencez à saisir le nom d'un groupe, une liste déroulante des groupes correspondants est proposée. La spécification d'un groupe d'hôtes parent sélectionne implicitement tous les groupes d'hôtes imbriqués. Les problèmes provenant de ces groupes d'hôtes seront affichés dans le widget ; si aucun groupe d'hôtes n'est saisi, les problèmes de tous les groupes d'hôtes seront affichés. Ce paramètre n'est pas disponible lors de la configuration du widget sur un tableau de bord de modèle.
Hosts	Sélectionnez les hôtes dont les problèmes doivent être affichés dans le widget. Ce champ prend en charge l'autocomplétion ; ainsi, lorsque vous commencez à saisir le nom d'un hôte, une liste déroulante des hôtes correspondants est proposée. Si aucun hôte n'est saisi, les problèmes de tous les hôtes seront affichés. Ce paramètre n'est pas disponible lors de la configuration du widget sur un tableau de bord de modèle.

Problem	Vous pouvez afficher uniquement les déclencheurs pour des problèmes particuliers. Pour cela, saisissez la chaîne à faire correspondre dans le nom du problème. Les macros ne sont pas développées.
Severity	Cochez les sévérités de déclencheur pour filtrer les déclencheurs à afficher dans le widget. Si aucune sévérité n'est cochée, tous les déclencheurs seront affichés.
Problem tags	Spécifiez les tags des problèmes à afficher dans le widget. Il est possible d'inclure comme d'exclure des tags et des valeurs de tag spécifiques. Plusieurs conditions peuvent être définies. La correspondance des noms de tag est toujours sensible à la casse. Plusieurs opérateurs sont disponibles pour chaque condition : **Exists** - inclure les noms de tag spécifiés ; **Equals** - inclure les noms et valeurs de tag spécifiés (sensible à la casse) ; **Contains** - inclure les noms de tag spécifiés lorsque les valeurs de tag contiennent la chaîne saisie (correspondance de sous-chaîne, insensible à la casse) ; **Does not exist** - exclure les noms de tag spécifiés ; **Does not equal** - exclure les noms et valeurs de tag spécifiés (sensible à la casse) ; **Does not contain** - exclure les noms de tag spécifiés lorsque les valeurs de tag contiennent la chaîne saisie (correspondance de sous-chaîne, insensible à la casse). Il existe deux types de calcul pour les conditions : **And/Or** - toutes les conditions doivent être remplies, les conditions ayant le même nom de tag seront regroupées par la condition *Or* ; **Or** - il suffit qu'une seule condition soit remplie.
Time period	Sélectionnez la source de données pour la période de temps : **Dashboard** - utiliser le sélecteur de période du tableau de bord ; **Widget** - utiliser un widget compatible (défini dans le paramètre *Widget*) ; **Custom** - utiliser une période de temps personnalisée définie dans les paramètres *From* et *To* ; si elle est définie, une icône d'horloge sera affichée dans le coin supérieur droit du widget, indiquant l'heure définie au survol de la souris. Notez que, quelle que soit la configuration de *Time period* du widget, les widgets compatibles peuvent toujours l'utiliser comme source de données pour la période de temps.
Widget	Saisissez ou sélectionnez un widget compatible comme source de données pour la période de temps. Ce paramètre est disponible si *Time period* est défini sur "Widget".
From	Saisissez ou sélectionnez le début de la période de temps. La syntaxe de temps relatif (`now`, `now/d`, `now/w-1w`, etc.) est prise en charge. Ce paramètre est disponible si *Time period* est défini sur "Custom".
To	Saisissez ou sélectionnez la fin de la période de temps. La syntaxe de temps relatif (`now`, `now/d`, `now/w-1w`, etc.) est prise en charge. Ce paramètre est disponible si *Time period* est défini sur "Custom".
Trigger limit	Définissez le nombre de déclencheurs à afficher. Plage de valeurs possible : 1-1000.

Vue d'ensemble des déclencheurs

Aperçu

Le widget *Aperçu des déclencheurs* affiche les états actuels des déclencheurs de l'hôte sous forme de tableau de blocs colorés. Il fournit un aperçu visuel rapide de l'état et de l'activité des déclencheurs sur différents hôtes.



Le widget utilise plusieurs éléments visuels pour présenter les détails des déclencheurs :

- Les couleurs des blocs pour les déclencheurs PROBLEM correspondent aux niveaux de gravité des problèmes.
- Les blocs clignotants indiquent des changements récents d'état des déclencheurs (au cours des 2 dernières minutes).
- Les flèches grises indiquent des déclencheurs avec dépendances (survolez-les pour voir les détails).
- Les icônes de case à cocher indiquent les problèmes reconnus ; tous les problèmes actifs ou résolus pour le déclencheur doivent être reconnus pour que cette icône apparaisse.

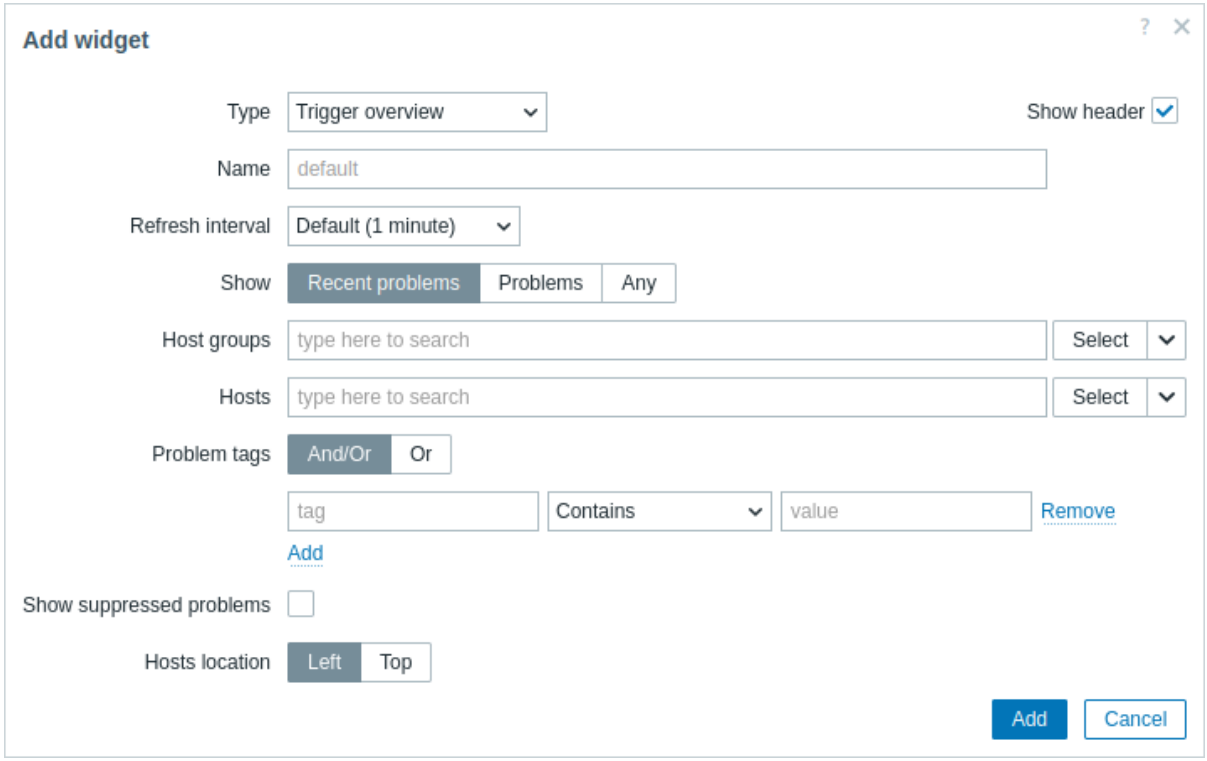
Note:
Les couleurs de gravité des déclencheurs peuvent être ajustées dans *Administration* > *Général*. Les gravités des déclencheurs peuvent être modifiées lors de la mise à jour des problèmes.

Un clic sur un bloc fournit des liens vers des informations liées aux déclencheurs (problèmes, configuration du déclencheur, etc.).

Par défaut, jusqu'à 50 enregistrements sont affichés (configurable dans *Administration* > *Général* > *GUI* sous *Nombre maximal de colonnes et de lignes dans les tableaux d'aperçu*). Si davantage de déclencheurs existent, un message vous invite à affiner les critères de filtrage. Cette limite est appliquée avant tout filtrage supplémentaire (par exemple, par tags).

Configuration

Pour configurer, sélectionnez *Aperçu des déclencheurs* comme type :



En plus des paramètres communs à tous les widgets, vous pouvez définir les options spécifiques suivantes :

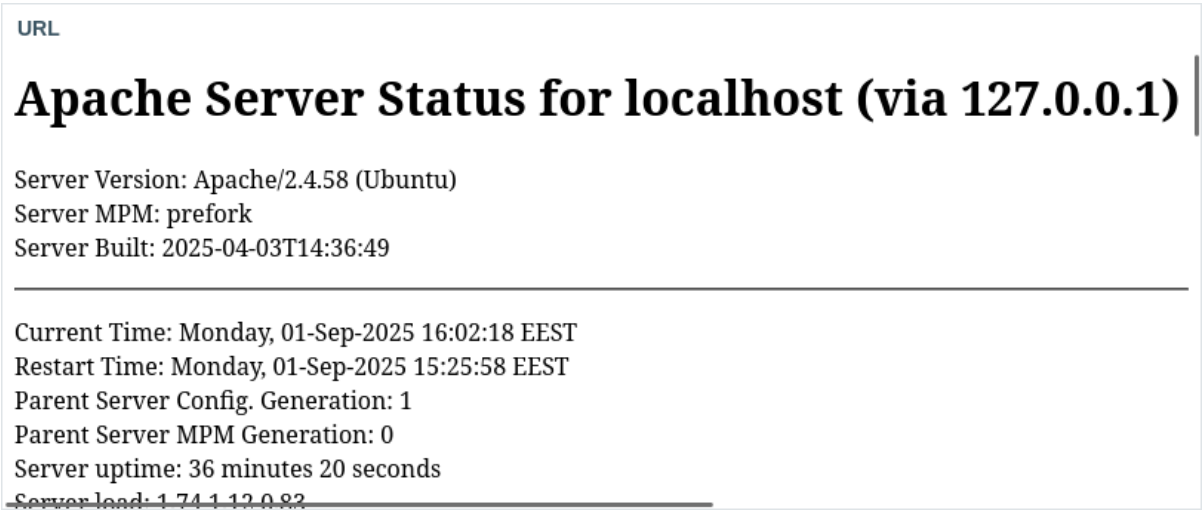
Show	Filtrer les déclencheurs par état du déclencheur : **Recent problems** - *(par défaut)* afficher les déclencheurs qui ont récemment été ou sont encore à l'état PROBLEM (résolus et non résolus) ; **Problems** - afficher les déclencheurs qui sont à l'état PROBLEM (non résolus) ; **Any** - afficher tous les déclencheurs.
Host groups	Sélectionnez des groupes d'hôtes. Vous pouvez également sélectionner un widget compatible comme source de données pour les groupes d'hôtes. Ce champ prend en charge l'autocomplétion ; ainsi, lorsque vous commencez à saisir le nom d'un groupe, une liste déroulante des groupes correspondants est proposée. Ce paramètre n'est pas disponible lors de la configuration du widget sur un tableau de bord de modèle.
Hosts	Sélectionnez des hôtes. Vous pouvez également sélectionner un widget compatible ou le tableau de bord comme source de données pour les hôtes. Ce champ prend en charge l'autocomplétion ; ainsi, lorsque vous commencez à saisir le nom d'un hôte, une liste déroulante des hôtes correspondants est proposée. Ce paramètre n'est pas disponible lors de la configuration du widget sur un tableau de bord de modèle.

Problem tags	Spécifiez des tags pour filtrer les déclencheurs affichés dans le widget. Il est possible d'inclure ou d'exclure des tags et des valeurs de tag spécifiques. Plusieurs conditions peuvent être définies. La correspondance des noms de tag est toujours sensible à la casse. **Remarque :** si le paramètre *Show* est défini sur « Any », tous les déclencheurs seront affichés même si des tags sont spécifiés. Cependant, bien que les changements récents d'état des déclencheurs (affichés sous forme de blocs clignotants) soient mis à jour pour tous les déclencheurs, les détails de l'état du déclencheur (couleur de gravité du problème et indication si le problème est acquitté) ne seront mis à jour que pour les déclencheurs correspondant aux tags spécifiés. Plusieurs opérateurs sont disponibles pour chaque condition : **Exists** - inclure les noms de tag spécifiés ; **Equals** - inclure les noms et valeurs de tag spécifiés (sensible à la casse) ; **Contains** - inclure les noms de tag spécifiés lorsque les valeurs de tag contiennent la chaîne saisie (correspondance de sous-chaîne, insensible à la casse) ; **Does not exist** - exclure les noms de tag spécifiés ; **Does not equal** - exclure les noms et valeurs de tag spécifiés (sensible à la casse) ; **Does not contain** - exclure les noms de tag spécifiés lorsque les valeurs de tag contiennent la chaîne saisie (correspondance de sous-chaîne, insensible à la casse). Il existe deux types de calcul pour les conditions : **And/Or** - toutes les conditions doivent être remplies, les conditions ayant le même nom de tag seront regroupées par la condition *Or* ; **Or** - il suffit qu'une seule condition soit remplie.
Show suppressed problems	Cochez la case pour afficher les problèmes qui seraient autrement supprimés (non affichés) en raison de la maintenance de l'hôte.
Layout	Sélectionnez l'option de disposition : **Horizontal** - les noms d'hôte seront affichés à gauche ; **Vertical** - les noms d'hôte seront affichés en haut.

URL

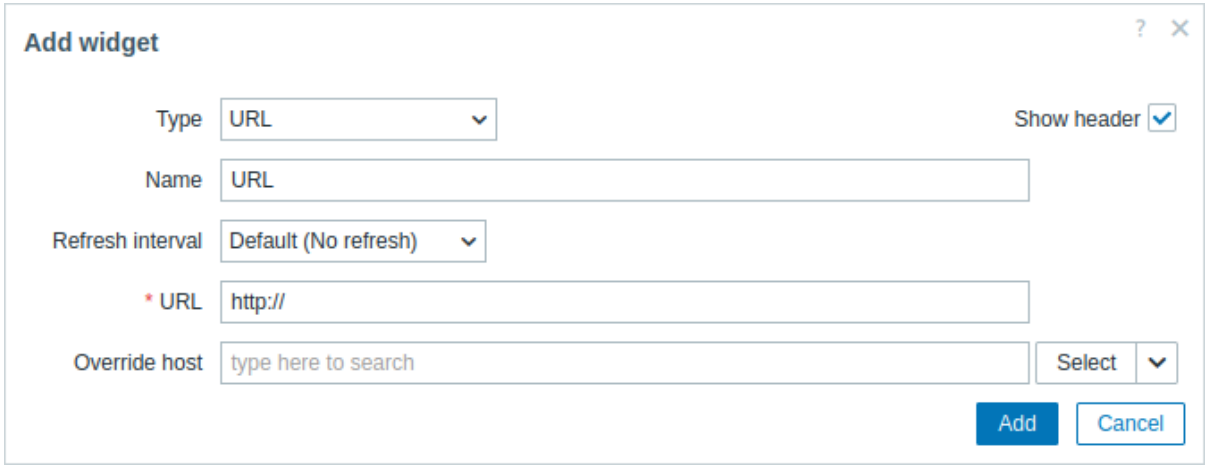
Aperçu

Le widget *URL* affiche le contenu récupéré à partir d'une URL.



Configuration

Pour configurer, sélectionnez *URL* comme type :



En plus des paramètres communs à tous les widgets, vous pouvez définir les options spécifiques suivantes :

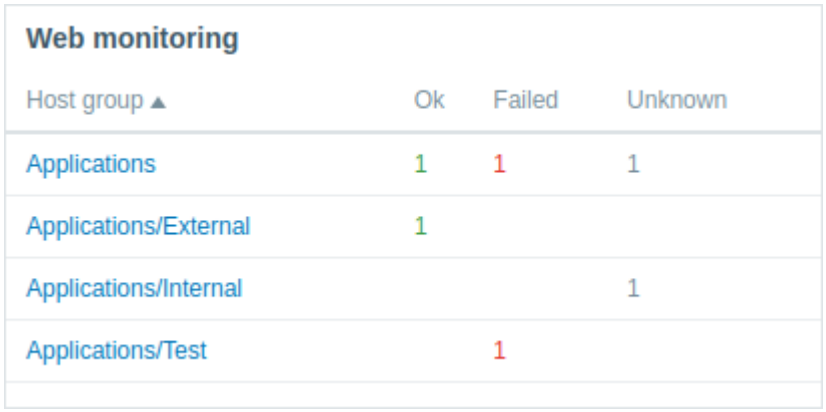
URL	Saisissez l'URL à afficher (jusqu'à 2048 caractères). Les URL externes doivent commencer par `http://` ou `https://`. Les URL internes prennent en charge les chemins relatifs (par exemple, `zabbix.php?action=report.status`). Les macros `{HOST.*}` sont prises en charge.
Override host	Sélectionnez un widget compatible ou le sélecteur d'hôte du tableau de bord comme source de données pour les hôtes. Ce paramètre n'est pas disponible lors de la configuration du widget sur un modèle de tableau de bord.

Attention:
Les navigateurs peuvent ne pas charger une page HTTP configurée dans le widget si l'interface Zabbix est accessible via HTTPS. De plus, vous devrez peut-être modifier l'en-tête de réponse HTTP X-Frame-Options pour afficher du contenu externe à l'interface Zabbix.

Surveillance web

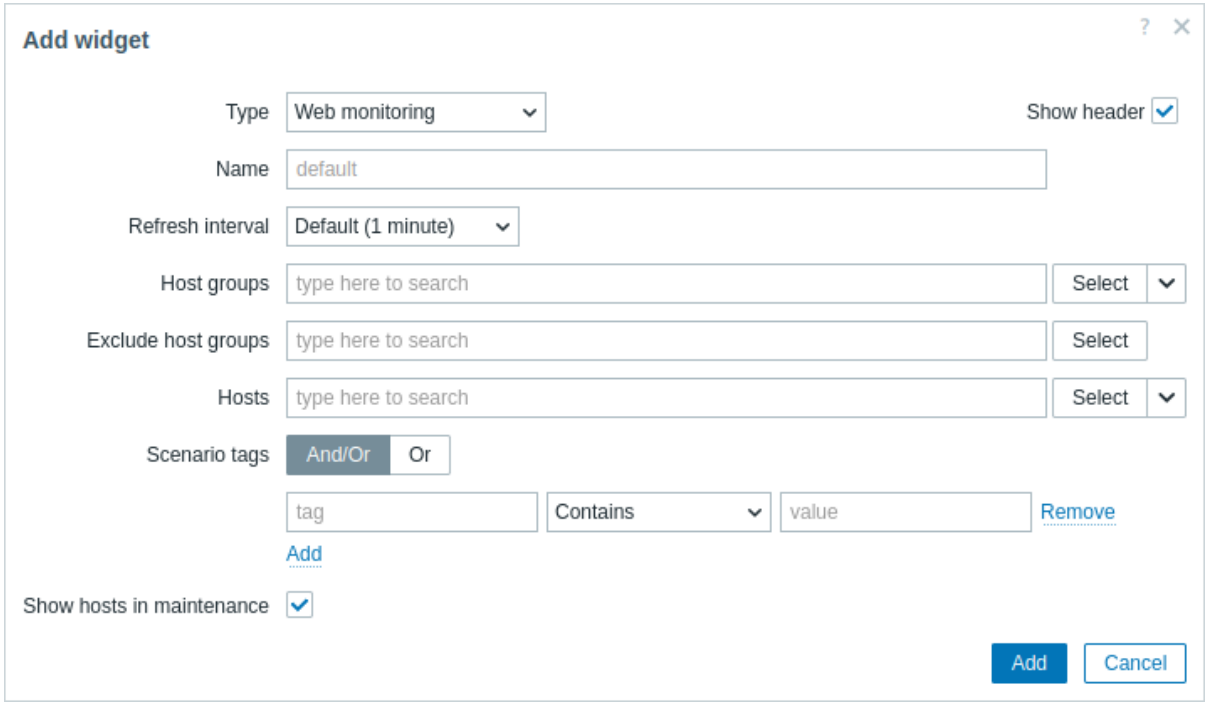
Aperçu

Le widget *Surveillance web* affiche l'état des scénarios web actifs.



Un clic sur le nom d'un groupe d'hôtes ouvre *Surveillance* > *Hôtes* > *Scénarios web*, où vous pouvez consulter les détails du scénario.

Configuration



En plus des paramètres communs à tous les widgets, vous pouvez définir les options spécifiques suivantes :

Groupes d'hôtes	Sélectionnez les groupes d'hôtes à afficher dans le widget. Vous pouvez également sélectionner un widget compatible comme source de données pour les groupes d'hôtes. Ce champ prend en charge l'auto-complétion ; lorsque vous commencez à saisir le nom d'un groupe, une liste déroulante des groupes correspondants est proposée. La spécification d'un groupe d'hôtes parent sélectionne implicitement tous les groupes d'hôtes imbriqués. Les données des hôtes de ces groupes d'hôtes seront affichées dans le widget ; si aucun groupe d'hôtes n'est saisi, tous les groupes d'hôtes seront affichés. Ce paramètre n'est pas disponible lors de la configuration du widget sur un tableau de bord de modèle.
Exclure les groupes d'hôtes	Sélectionnez les groupes d'hôtes à masquer dans le widget. Ce champ prend en charge l'auto-complétion ; lorsque vous commencez à saisir le nom d'un groupe, une liste déroulante des groupes correspondants est proposée. La spécification d'un groupe d'hôtes parent sélectionne implicitement tous les groupes d'hôtes imbriqués. Les données des hôtes de ces groupes d'hôtes ne seront pas affichées dans le widget. Par exemple, les hôtes 001, 002, 003 peuvent appartenir au groupe A et les hôtes 002, 003 également au groupe B. Si nous choisissons d'*afficher* le groupe A et d'*exclure* le groupe B en même temps, seules les données de l'hôte 001 seront affichées dans le tableau de bord. Ce paramètre n'est pas disponible lors de la configuration du widget sur un tableau de bord de modèle.
Hôtes	Sélectionnez les hôtes à afficher dans le widget. Vous pouvez également sélectionner un widget compatible ou le tableau de bord comme source de données pour les hôtes. Ce champ prend en charge l'auto-complétion ; lorsque vous commencez à saisir le nom d'un hôte, une liste déroulante des hôtes correspondants est proposée. Si aucun hôte n'est saisi, tous les hôtes seront affichés. Ce paramètre n'est pas disponible lors de la configuration du widget sur un tableau de bord de modèle.

Tags de scénario	Spécifiez des tags pour limiter le nombre de scénarios web affichés dans le widget. Il est possible d'inclure comme d'exclure des tags et des valeurs de tag spécifiques. Plusieurs conditions peuvent être définies. La correspondance des noms de tag est toujours sensible à la casse. Plusieurs opérateurs sont disponibles pour chaque condition : **Existe** - inclure les noms de tag spécifiés ; **Égal à** - inclure les noms de tag et les valeurs spécifiés (sensible à la casse) ; **Contient** - inclure les noms de tag spécifiés lorsque les valeurs de tag contiennent la chaîne saisie (correspondance de sous-chaîne, insensible à la casse) ; **N'existe pas** - exclure les noms de tag spécifiés ; **N'est pas égal à** - exclure les noms de tag et les valeurs spécifiés (sensible à la casse) ; **Ne contient pas** - exclure les noms de tag spécifiés lorsque les valeurs de tag contiennent la chaîne saisie (correspondance de sous-chaîne, insensible à la casse). Deux types de calcul sont disponibles pour les conditions : **Et/Ou** - toutes les conditions doivent être remplies, les conditions ayant le même nom de tag seront regroupées par la condition *Ou* ; **Ou** - il suffit qu'une seule condition soit remplie.
Afficher les hôtes en maintenance	Inclure dans les statistiques les hôtes qui sont en maintenance. Ce paramètre est libellé *Afficher les données en maintenance* lors de la configuration du widget sur un tableau de bord de modèle.

Surveillance

Vue d'ensemble

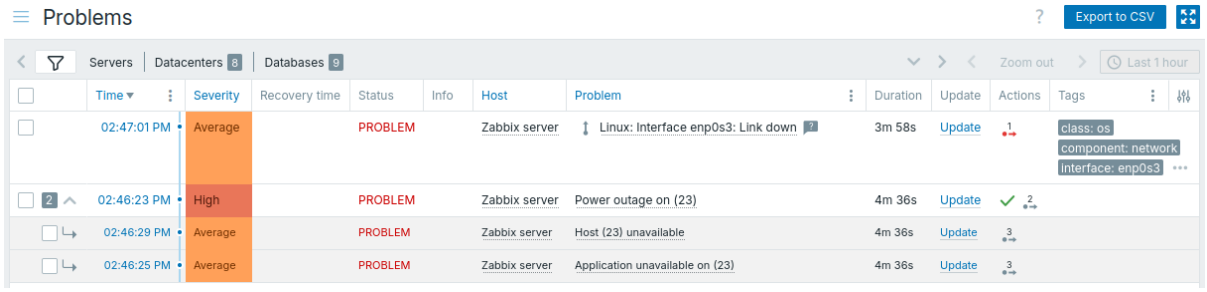
Le menu Monitoring est consacré à l'affichage des données. Toutes les informations que Zabbix est configuré pour collecter, visualiser et traiter seront affichées dans les différentes sections du menu *Monitoring*.

Problèmes

Aperçu

Dans *Monitoring* > *Problems*, vous pouvez voir les problèmes que vous avez actuellement. Les problèmes sont les déclencheurs qui se trouvent dans l'état "Problem".

Par défaut, tous les nouveaux problèmes sont classés comme problèmes de cause. Il est possible de reclasser manuellement certains problèmes en tant que problèmes de symptôme du problème de cause. Pour plus de détails, voir cause and symptom events.

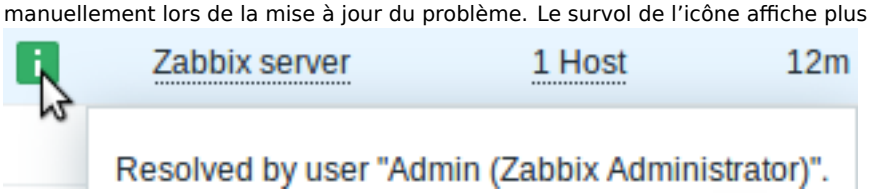
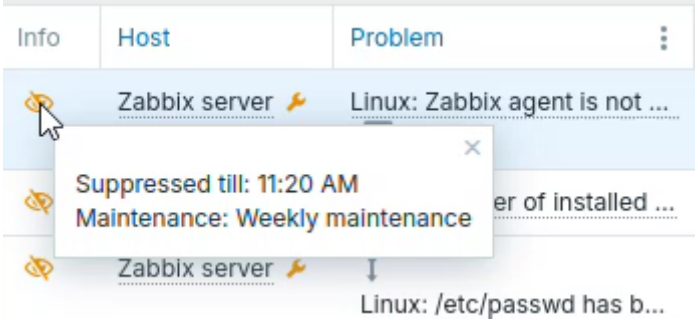


Les détails du problème sont affichés dans un tableau personnalisable. Le bouton *Customize table* dans l'en-tête du tableau vous permet d'afficher, de masquer ou de réorganiser les colonnes, ainsi que de rétablir la disposition par défaut. Les options suivantes sont également disponibles :

- *Compact view* - réduit la hauteur des lignes afin d'afficher davantage de problèmes à l'écran.
- *Highlight full row* - utilise la couleur de gravité du problème pour mettre en évidence les problèmes non résolus (non disponible dans les thèmes à contraste élevé).

Vous pouvez également redimensionner chaque colonne en faisant glisser sa bordure droite ; double-cliquez sur la bordure pour réinitialiser. Toutes les personnalisations sont appliquées immédiatement et sont toujours mémorisées pour l'utilisateur actuel dans l'onglet filter actif (par défaut ou enregistré).

Les colonnes suivantes sont disponibles :

Column	Description
Checkbox	Des cases à cocher pour la sélection des problèmes sont affichées. Les icônes à côté des cases à cocher ont la signification suivante :  - le nombre d'événements de symptôme pour le problème de cause ;  - développer pour afficher les événements de symptôme ;  - réduire pour masquer les événements de symptôme ;  - il s'agit d'un événement de symptôme.
Time	L'heure de début du problème est affichée. En cliquant sur le bouton *Options* dans l'en-tête de la colonne, vous accédez à l'option *Show timeline*, qui active ou désactive la chronologie visuelle et le regroupement des problèmes (non disponible lorsque *Compact view* est activé).
Severity	La gravité du problème est affichée. La gravité du problème est initialement basée sur la gravité du déclencheur de problème sous-jacent ; toutefois, après la survenue de l'événement, elle peut être mise à jour à l'aide de l'écran *Update problem* screen. La couleur de la gravité du problème est utilisée comme arrière-plan de cellule pendant la durée du problème.
Recovery time	L'heure de résolution du problème est affichée Cette colonne n'est pas disponible lorsque *Show* dans le filter est défini sur *Problems*.
Status	L'état du problème est affiché : **Problem** - problème non résolu **Resolved** - problème récemment résolu. Vous pouvez masquer les problèmes récemment résolus à l'aide du filtre. Les nouveaux problèmes et les problèmes récemment résolus clignotent pendant 2 minutes. Les problèmes résolus sont affichés pendant 5 minutes au total. Ces deux valeurs sont configurables dans *Administration* > *General* > *Trigger displaying options*. Cette colonne n'est pas disponible lorsque *Show* dans le filter est défini sur *Problems*.
Info	Une icône d'information verte s'affiche si un problème est fermé par corrélation globale ou manuellement lors de la mise à jour du problème. Le survol de l'icône affiche plus de détails :  Si un problème supprimé est affiché, l'icône suivante est affichée : . Lorsque l'événement est supprimé par la maintenance de l'hôte, l'icône est affichée en orange : . Si la maintenance et la suppression manuelle s'appliquent toutes deux, la suppression manuelle est prioritaire pour la couleur de l'icône. Le survol de l'icône affiche plus de détails : 
Host	L'hôte du problème est affiché. En cliquant sur le nom de l'hôte, vous ouvrez le host menu.

Column	Description
Problem	Le nom du problème est affiché. Le nom du problème est basé sur le nom du déclencheur de problème sous-jacent. Les macros dans le nom du déclencheur sont résolues au moment où le problème se produit et les valeurs résolues ne sont plus mises à jour. En cliquant sur le nom du problème, vous ouvrez le event menu. Le survol de l'icône  après le nom du problème affiche la description du déclencheur (pour les problèmes qui en disposent). En cliquant sur le bouton *Options* dans l'en-tête de la colonne, vous accédez à deux options : - *Show operational data* affiche les données opérationnelles après le nom du problème entre parenthèses (uniquement si le champ *Operational data* n'est pas vide dans la trigger configuration). - *Show trigger expression* affiche l'expression du déclencheur du problème sous le nom du problème (non disponible lorsque *Compact view* est activé).
Operational data	Des données dynamiques en temps réel pour le problème sont affichées. Le contenu dépend du champ *Operational data* dans la trigger configuration : si une chaîne avec des macros est définie, elle est affichée ici ; si le champ est laissé vide, les dernières valeurs de tous les éléments de l'expression du déclencheur sont affichées à la place.
Duration	La durée du problème est affichée. Voir aussi : Negative problem duration
Update	Cliquez sur le lien *Update* pour accéder à l'écran de problem update, où diverses actions peuvent être effectuées sur le problème, notamment l'ajout de commentaires et la prise en compte du problème.
Actions	L'historique des activités concernant le problème est affiché à l'aide d'icônes symboliques :  - le problème a été pris en compte. Cette icône est toujours affichée en premier.  - des commentaires ont été ajoutés. Le nombre de commentaires est également affiché.  - la gravité du problème a augmenté (par exemple, Information > Warning).  - la gravité du problème a diminué (par exemple, Warning > Information).  - la gravité du problème a été modifiée, puis est revenue au niveau d'origine (par exemple, Warning > Information > Warning).  - le problème a été supprimé.  - la suppression du problème a été levée.  - le problème a été supprimé en raison de la maintenance de l'hôte.  - la suppression du problème a été levée après la maintenance de l'hôte.  - des actions ont été effectuées. Le nombre d'actions est également affiché.  - des actions ont été effectuées, au moins une est en cours. Le nombre d'actions est également affiché.  - des actions ont été effectuées, au moins une a échoué. Le nombre d'actions est également affiché. Lorsque vous survolez les icônes avec la souris, des fenêtres contextuelles affichant des détails sur l'activité sont présentées. Voir viewing details pour en savoir plus sur les icônes utilisées dans la fenêtre contextuelle pour les actions effectuées.
Tags	Les tags du problème sont affichés. En outre, des tags provenant d'un système de ticketing externe peuvent également être affichés (voir l'option *Process tags* lors de la configuration des webhooks). Le survol ou le clic sur l'icône à trois points ••• après les tags affichés ouvre une fenêtre contextuelle avec tous les tags. En cliquant sur le bouton *Options* dans l'en-tête de la colonne, vous pouvez définir le nombre de tags affichés (1-3), choisir la manière dont les noms des tags sont affichés (complets, abrégés ou aucun), définir la priorité d'affichage des tags, ou dupliquer la colonne (les colonnes dupliquées peuvent être renommées ou supprimées).

Column	Description
Tag value	Les valeurs des tags du problème sont affichées. En cliquant sur le bouton ⋮ *Options* dans l'en-tête de la colonne, vous pouvez renommer la colonne, choisir quelles valeurs de tags afficher, ou dupliquer la colonne (les colonnes dupliquées peuvent être renommées ou supprimées).
Custom text	Du texte personnalisé, des valeurs de macros ou une combinaison des deux sont affichés. En cliquant sur le bouton ⋮ *Options* dans l'en-tête de la colonne, vous pouvez renommer la colonne, saisir le texte personnalisé ou les macros, ou dupliquer la colonne (les colonnes dupliquées peuvent être renommées ou supprimées). Supported macros: `{HOST.*}` (sauf `{HOST.METADATA}`), `{INVENTORY.*}`, `{ITEM.*}` (sauf `{ITEM.STATE.ERROR}`), `{TRIGGER.ID}`, `{EVENT.ID}`, et les user macros.

Durée de problème négative

Il est effectivement possible, dans certaines situations courantes, d'avoir une durée de problème négative, c'est-à-dire lorsque l'heure de résolution du problème est antérieure à l'heure de création du problème, par exemple :

- Si un hôte est surveillé par un proxy et qu'une erreur réseau se produit, entraînant l'absence de données reçues du proxy pendant un certain temps, le déclencheur nodata(/host/key) sera activé par le serveur. Lorsque la connexion est rétablie, le serveur recevra du proxy des données d'élément horodatées dans le passé. Le problème nodata(/host/key) sera alors résolu et aura une durée de problème négative ;
- Lorsque les données d'élément qui résolvent l'événement de problème sont envoyées par Zabbix sender et contiennent un horodatage antérieur à l'heure de création du problème, une durée de problème négative sera également affichée.

Note:
Une durée de problème négative n'affecte en aucune façon le calcul du SLA ni le rapport de disponibilité d'un déclencheur particulier ; elle ne réduit ni n'augmente le temps de problème.

Options de modification en masse

Les boutons sous la liste offrent plusieurs options de modification en masse :

- *Mise à jour en masse* - met à jour les problèmes sélectionnés en accédant à l'écran de mise à jour des problèmes

Pour utiliser cette option, cochez les cases correspondant aux problèmes concernés, puis cliquez sur le bouton *Mise à jour en masse*.

Boutons

Le bouton à droite offre l'option suivante :

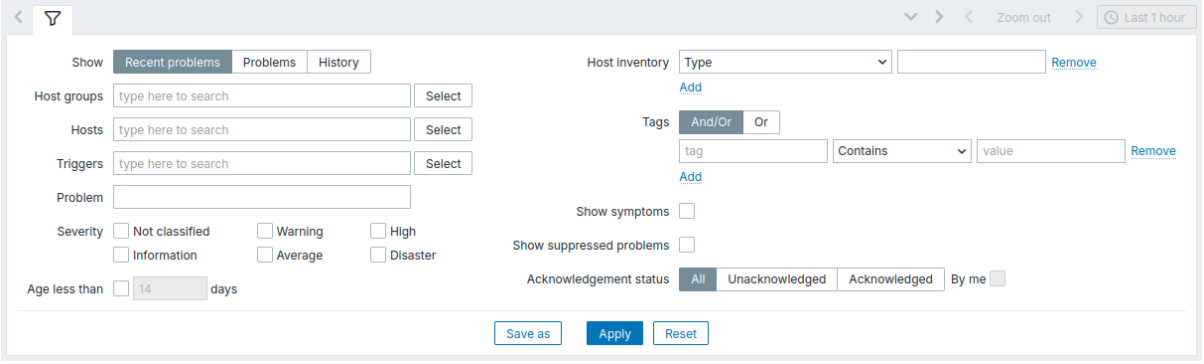
Export to CSV	Exporter le contenu de toutes les pages vers un fichier CSV.

Les boutons du mode d'affichage, communs à toutes les sections, sont décrits sur la page Monitoring.

Utilisation du filtre

Vous pouvez utiliser le filtre pour afficher uniquement les problèmes qui vous intéressent. Pour de meilleures performances de recherche, la recherche des données s'effectue avec des macros non résolues.

Le filtre se trouve au-dessus du tableau. Les paramètres de filtre favoris peuvent être enregistrés sous forme d'onglets, puis rapidement accessibles en cliquant sur les onglets au-dessus du filtre.

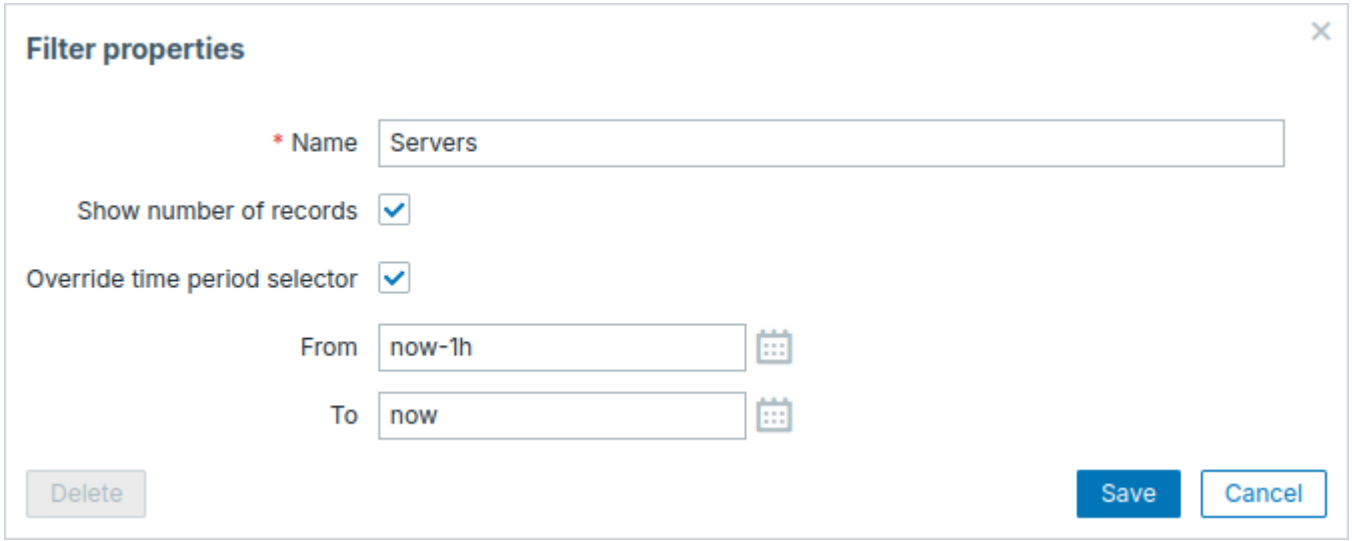


Parameter	Description
Show	Filtrer par statut du problème: **Recent problems** - les problèmes non résolus et récemment résolus sont affichés (par défaut) **Problems** - les problèmes non résolus sont affichés **History** - l'historique de tous les événements est affiché
Host groups	Filtrer par un ou plusieurs groupes d'hôtes. La spécification d'un groupe d'hôtes parent sélectionne implicitement tous les groupes d'hôtes imbriqués.
Hosts	Filtrer par un ou plusieurs hôtes.
Triggers	Filtrer par un ou plusieurs déclencheurs.
Problem	Filtrer par nom de problème.
Severity	Filtrer par gravité du déclencheur (problème).
Age less than	Filtrer selon l'ancienneté du problème.
Host inventory	Filtrer par type et valeur d'inventaire.
Tags	Filtrer par nom et valeur de balise d'événement. Il est possible d'inclure ainsi que d'exclure des balises et des valeurs de balise spécifiques. Plusieurs conditions peuvent être définies. La correspondance du nom de balise est toujours sensible à la casse. Plusieurs opérateurs sont disponibles pour chaque condition: **Exists** - inclure les noms de balise spécifiés **Equals** - inclure les noms et valeurs de balise spécifiés (sensible à la casse) **Contains** - inclure les noms de balise spécifiés lorsque les valeurs de balise contiennent la chaîne saisie (correspondance de sous-chaîne, insensible à la casse) **Does not exist** - exclure les noms de balise spécifiés **Does not equal** - exclure les noms et valeurs de balise spécifiés (sensible à la casse) **Does not contain** - exclure les noms de balise spécifiés lorsque les valeurs de balise contiennent la chaîne saisie (correspondance de sous-chaîne, insensible à la casse) Il existe deux types de calcul pour les conditions: **And/Or** - toutes les conditions doivent être remplies, les conditions ayant le même nom de balise seront regroupées par la condition Or **Or** - il suffit qu'une condition soit remplie Lors du filtrage, les balises spécifiées ici seront affichées en premier avec le problème, sauf si la liste *Tag display priority* (voir ci-dessous) en décide autrement.
Show symptoms	Cochez la case pour afficher sur sa propre ligne les problèmes classés comme symptômes.
Show suppressed problems	Cochez la case pour afficher les problèmes qui seraient autrement supprimés et masqués en raison de la maintenance de l'hôte ou supprimés manuellement par un utilisateur.
Acknowledgement status	Filtrer pour afficher tous les problèmes, uniquement les problèmes non reconnus, ou uniquement les problèmes reconnus. Cochez la case supplémentaire pour filtrer les problèmes que vous avez déjà reconnus.

Onglets pour les filtres favoris

Les ensembles de paramètres de filtre fréquemment utilisés peuvent être enregistrés dans des onglets.

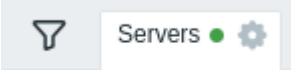
Pour enregistrer un nouvel ensemble de paramètres de filtre, ouvrez l'onglet principal, configurez les paramètres du filtre, puis cliquez sur le bouton *Save as*. Dans une nouvelle fenêtre contextuelle, définissez les *Filter properties*.



Parameter	Description
Name	Le nom du filtre à afficher dans la liste des onglets.
Show number of records	Cochez cette option si vous souhaitez que le nombre de problèmes soit affiché à côté du nom de l'onglet.
Override time period selector	Cochez cette option pour définir une période par défaut spécifique pour cet ensemble de filtres. Si cette option est activée, vous ne pourrez modifier la période de cet onglet qu'en mettant à jour les paramètres du filtre. Pour les onglets sans période personnalisée, la plage de temps peut être modifiée en cliquant sur le bouton de sélection de l'heure dans le coin supérieur droit (le nom du bouton dépend de l'intervalle de temps sélectionné: Cette semaine, 30 dernières minutes, Hier, etc.). Cette option est disponible uniquement pour les filtres dans *Monitoring* > *Problems*.
From/To	Début et fin de la période en syntaxe absolue (`Y-m-d H:i:s`) ou relative (`now-1d`). Disponible si *Set custom time period* est coché.

Une fois enregistré, le filtre est créé sous forme d'onglet de filtre nommé et est immédiatement activé.

Pour modifier les propriétés de filtre d'un filtre existant, cliquez sur le symbole d'engrenage à côté du nom de l'onglet actif.



Notes:

- Pour masquer la zone de filtre, cliquez sur le nom de l'onglet actuel. Cliquez à nouveau sur le nom de l'onglet actif pour rouvrir la zone de filtre.
- La navigation au clavier est prise en charge: utilisez les flèches pour passer d'un onglet à l'autre, puis appuyez sur *Enter* pour ouvrir.
- Les boutons gauche/droite au-dessus du filtre peuvent être utilisés pour passer d'un filtre enregistré à un autre. Le bouton pointant vers le bas ouvre également un menu déroulant contenant tous les filtres enregistrés, et vous pouvez cliquer sur celui dont vous avez besoin.
- Si les paramètres d'un filtre enregistré ont été modifiés (mais non enregistrés), un point vert s'affiche après le nom du filtre. Pour mettre à jour le filtre selon les nouveaux paramètres, cliquez sur le bouton *Update*, qui s'affiche à la place du bouton *Save as*.
- La disposition des colonnes du tableau est enregistrée pour chaque onglet de filtre. Pour personnaliser la disposition d'un filtre enregistré, ouvrez d'abord cet onglet de filtre, puis personnalisez les colonnes.
- Les onglets de filtre, les paramètres de filtre et la disposition des colonnes du tableau pour chaque onglet sont mémorisés dans le profil utilisateur et restaurés lorsque l'utilisateur rouvre la page.

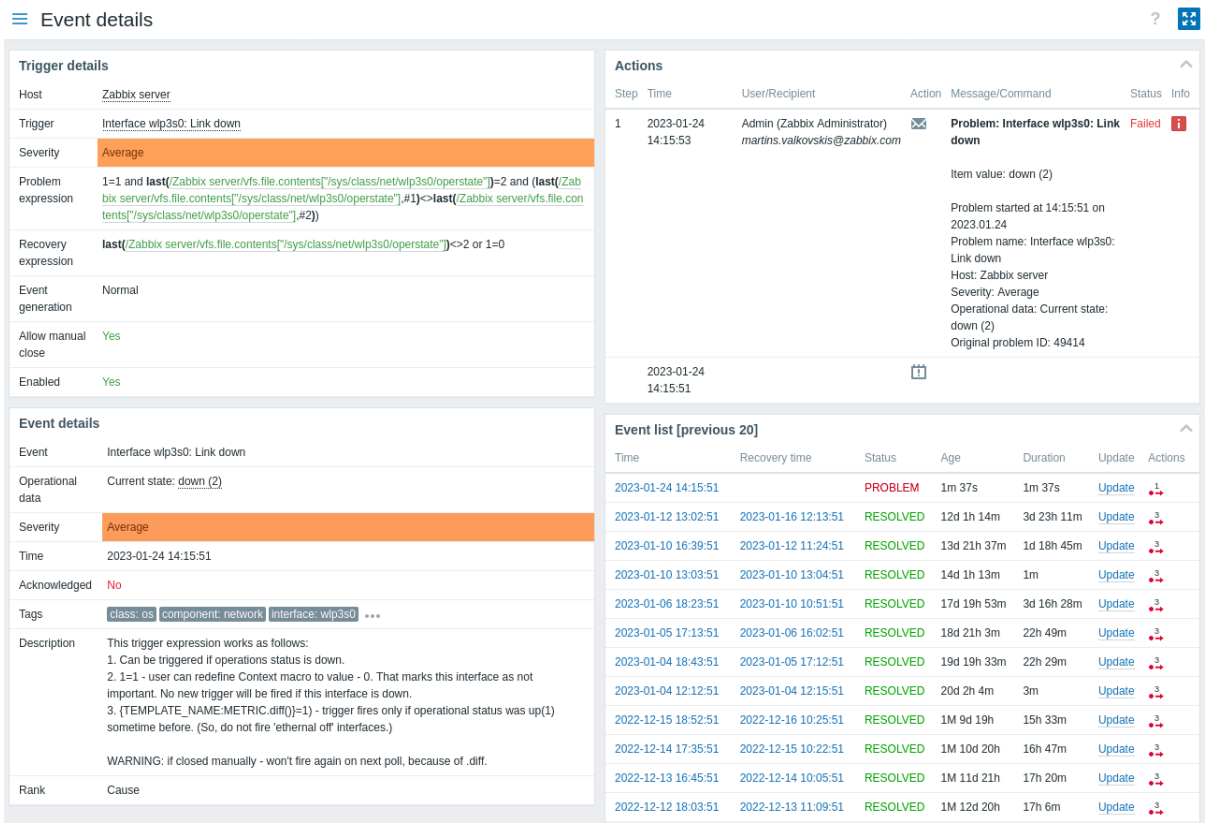
Note:
Pour partager des filtres, copiez et envoyez à d'autres une URL d'un filtre actif. Après avoir ouvert cette URL, les autres utilisateurs pourront enregistrer cet ensemble de paramètres comme filtre permanent dans leur compte Zabbix.
 Voir aussi: Paramètres de page.

Boutons de filtrage

Apply	Appliquer les critères de filtrage spécifiés (sans enregistrer).
Reset	Réinitialiser le filtre actuel et revenir aux paramètres enregistrés de l'onglet actuel. Dans l'onglet principal, cela effacera le filtre.
Save as	Enregistrer les paramètres de filtre actuels dans un nouvel onglet. Uniquement disponible sur l'onglet principal.
Update	Remplacer les paramètres de l'onglet par les paramètres actuellement spécifiés. Non disponible sur l'onglet principal.

Affichage des détails

Les heures de début du problème et de rétablissement dans *Monitoring* > *Problems* sont des liens. En cliquant dessus, vous ouvrez des détails supplémentaires sur l'événement.



Dans les détails de l'événement, vous pouvez voir si l'événement a été supprimé en raison d'une maintenance ou supprimé manuellement par un utilisateur.

Notez que la gravité du problème peut différer pour le déclencheur et l'événement de problème - si elle a été mise à jour pour l'événement de problème à l'aide de l'écran *Update problem* screen.

Dans la liste des actions, les icônes suivantes sont utilisées pour indiquer le type d'activité :

- - événement de problème généré.
- - message envoyé.
- - événement de problème acquitté.
- - événement de problème non acquitté.
- - un commentaire a été ajouté.
- - la gravité du problème a augmenté (par exemple, Information > Warning).
- - la gravité du problème a diminué (par exemple, Warning > Information).
- - la gravité du problème a été modifiée, puis est revenue au niveau d'origine (par exemple, Warning > Information > Warning).
- - une commande distante a été exécutée.
- - l'événement de problème a été rétabli.
- - le problème a été fermé manuellement.
- - le problème a été supprimé.
- - la suppression du problème a été levée.
- - le problème a été supprimé en raison de la maintenance de l'hôte.
- - la suppression du problème a été levée après la maintenance de l'hôte.
- - le problème a été converti en problème symptôme.
- - le problème a été converti en problème cause.

Aperçu

Par défaut, tous les nouveaux problèmes sont classés comme des problèmes de cause. Il est possible de reclasser manuellement certains problèmes comme des problèmes de symptôme du problème de cause.

Par exemple, une panne de courant peut être la cause racine réelle pour laquelle un hôte est injoignable ou un service est indisponible. Dans ce cas, les problèmes « hôte injoignable » et « service indisponible » doivent être classés comme des problèmes de symptôme de la « panne de courant » — le problème de cause.

La hiérarchie cause-symptôme ne prend en charge que deux niveaux. Un problème qui est déjà un symptôme ne peut pas se voir attribuer de problèmes de symptôme « subordonnés » ; tous les problèmes attribués comme symptômes à un problème de symptôme deviendront des symptômes du même problème de cause.

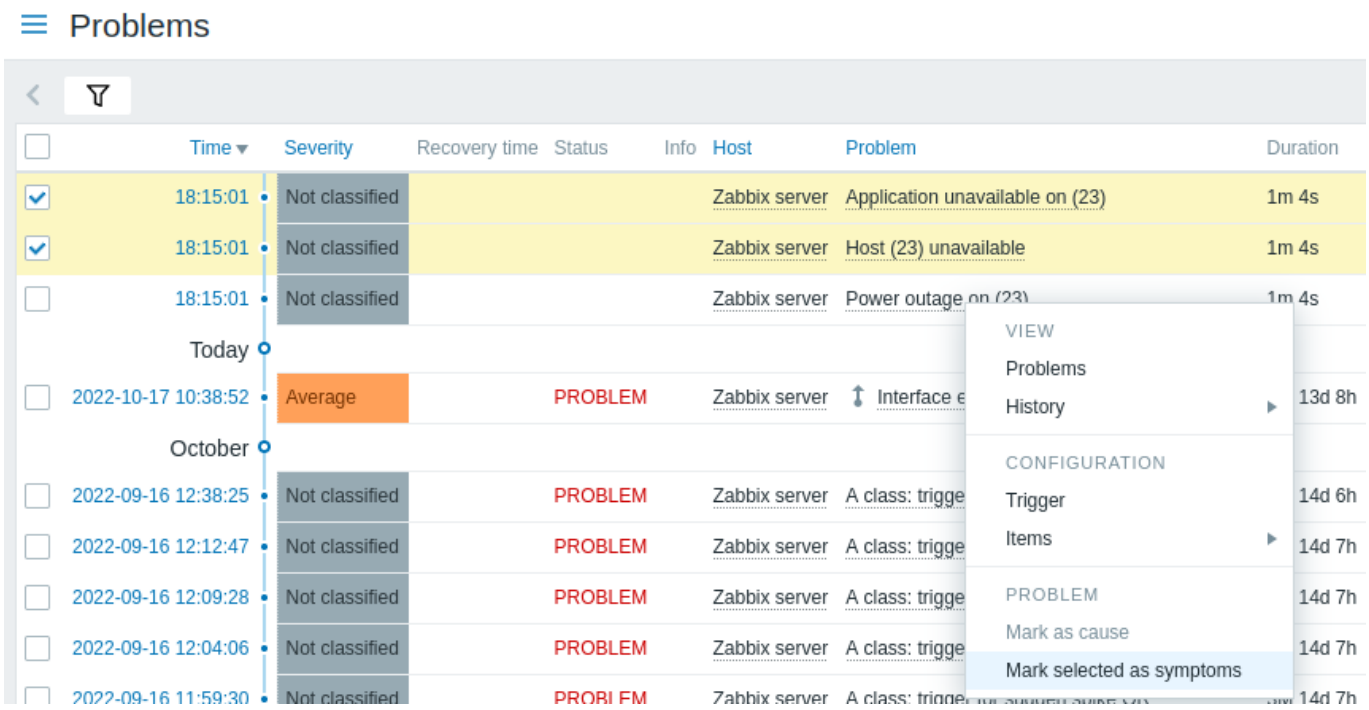
Seuls les problèmes de cause sont comptabilisés dans les totaux des problèmes sur les cartes, dans les widgets du tableau de bord tels que *Problèmes par gravité* ou *Hôtes avec problèmes*, etc. Cependant, le classement des problèmes n'affecte pas les services.

Un problème de symptôme ne peut être lié qu'à un seul problème de cause. Les problèmes de symptôme ne sont pas résolus automatiquement si le problème de cause est résolu ou fermé.

Configuration

Pour reclasser un problème en tant que problème symptôme, sélectionnez d'abord ce problème dans la liste des problèmes. Un ou plusieurs problèmes peuvent être sélectionnés.

Ensuite, allez au problème cause et, dans son menu contextuel, cliquez sur l'option *Marquer la sélection comme symptômes*.



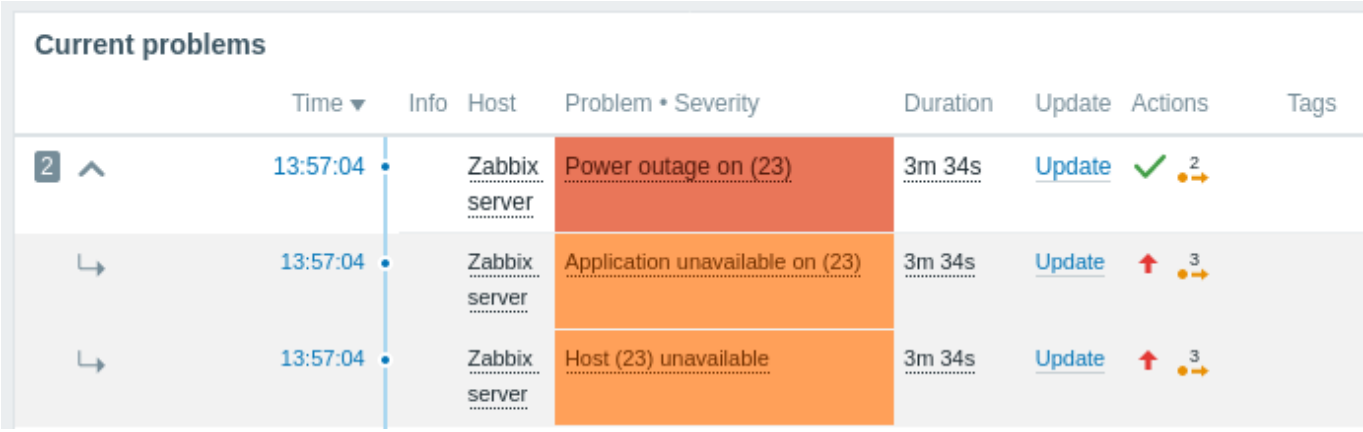
Après cela, les problèmes sélectionnés seront mis à jour par le serveur en tant que problèmes symptôme du problème cause.

Pendant la mise à jour du statut du problème, celui-ci est affiché de l'une des deux façons suivantes :

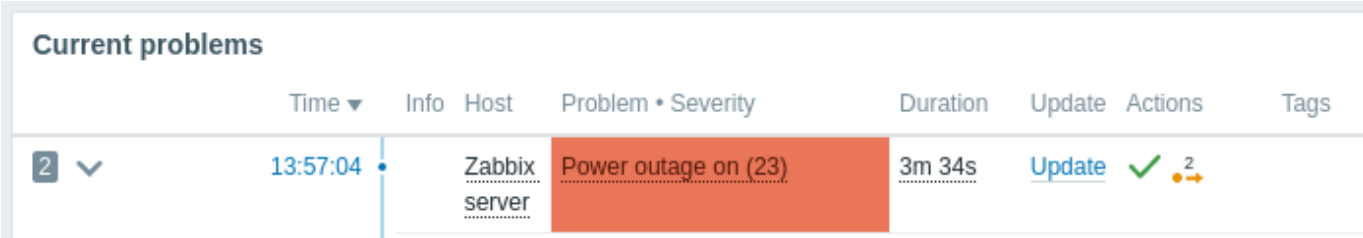
- Un statut clignotant "UPDATING" est affiché dans la colonne Status ;
- Une icône clignotante ou dans la colonne Info (cela s'applique si seuls *Problems* sont sélectionnés dans le filtre et que la colonne Status n'est donc pas affichée).

Affichage

Les problèmes de symptôme sont affichés sous le problème cause et marqués en conséquence dans *Monitoring* > *Problems* (ainsi que dans le widget du tableau de bord *Problems*) - avec une icône, une police plus petite et un arrière-plan différent.



Dans la vue réduite, seul le problème cause est visible ; l'existence de problèmes de symptôme est indiquée par le nombre au début de la ligne et par l'icône permettant de développer la vue.



Il est également possible d'afficher en plus les problèmes de symptôme en police normale et sur leur propre ligne. Pour cela, sélectionnez *Show symptoms* dans les paramètres du filtre ou dans la configuration du widget.

Revenir à un problème cause

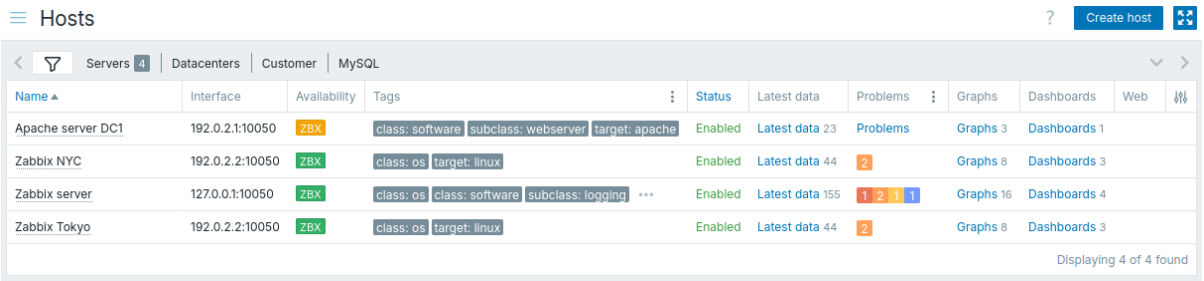
Un problème symptôme peut être rétabli en tant que problème cause. Pour ce faire, soit :

- cliquez sur l'option *Marquer comme cause* dans le menu contextuel du problème symptôme ;
- cochez l'option *Convertir en cause* dans l'écran de mise à jour du problème et cliquez sur *Mettre à jour* (cette option fonctionne également si plusieurs problèmes sont sélectionnés).

Hôtes

Aperçu

La section *Monitoring* > *Hôtes* affiche une liste complète des hôtes surveillés avec des informations détaillées sur l'interface de l'hôte, la disponibilité, les tags, les problèmes actuels, l'état (activé/désactivé) et des liens permettant de naviguer facilement vers les dernières données de l'hôte, l'historique des problèmes, les graphiques, les tableaux de bord et les scénarios web.



Les détails de l'hôte sont affichés dans un tableau personnalisable. Le bouton *Personnaliser le tableau* dans l'en-tête du tableau vous permet d'afficher, de masquer ou de réorganiser les colonnes, et de rétablir la disposition par défaut. Vous pouvez également redimensionner chaque colonne en faisant glisser sa bordure droite ; double-cliquez sur la bordure pour la réinitialiser. Toutes les personnalisations sont appliquées immédiatement et sont toujours mémorisées pour l'utilisateur actuel dans l'onglet filtre actif (par défaut ou enregistré).

Les colonnes suivantes sont disponibles :

Column	Description
Name	Le nom visible de l'hôte. Cliquer sur le nom ouvre le menu de l'hôte. Une icône de clé orange après le nom indique que cet hôte est en maintenance. Cliquez sur l'en-tête de la colonne pour trier les hôtes par nom par ordre croissant ou décroissant.
Interface	L'interface principale de l'hôte est affichée.
Availability	La disponibilité de l'hôte par interface configurée est affichée. Les icônes de disponibilité représentent l'état actuel de l'interface de l'hôte sur le serveur Zabbix. Par conséquent, si vous désactivez un hôte dans l'interface, sa disponibilité sera mise à jour après que le serveur Zabbix aura synchronisé les modifications de configuration. De même, si vous activez un hôte, sa disponibilité sera mise à jour après que le serveur Zabbix aura synchronisé les modifications de configuration et interrogé l'hôte. Les icônes de disponibilité représentent uniquement les types d'interface configurés (Agent, SNMP, IPMI, JMX). Le survol de l'icône affiche une fenêtre contextuelle avec la liste de toutes les interfaces du même type, avec les détails, l'état et les erreurs. Pour l'interface Agent, la fenêtre contextuelle affiche les interfaces (passives) et les vérifications actives. Si un hôte n'a que des vérifications actives, l'icône d'interface Agent est affichée même si aucune interface Agent n'est configurée pour l'hôte. La colonne est vide pour les hôtes sans interface. L'état d'une seule interface d'hôte est déterminé par la connexion entre un élément activé utilisant l'interface et l'hôte. L'état peut être : **Available** - la connexion à l'hôte a réussi ; **Not available** - la connexion à l'hôte a échoué (expiration du délai, problèmes de pare-feu, etc.) ; **Unknown** - la connexion à l'hôte n'a pas été tentée ou le résultat est inconnu. Pour plus de détails sur la manière dont le serveur Zabbix détermine l'état de l'interface, voir Unknown interface status et Unreachable/unavailable host interface settings. L'état de toutes les interfaces d'un hôte d'un même type (Agent, SNMP, IPMI, JMX) est déterminé par les interfaces utilisées par au moins un élément activé. L'état est indiqué par la couleur de l'icône : **Green** - toutes les interfaces sont disponibles ; **Yellow** - au moins une interface n'est pas disponible, et au moins une est disponible ou inconnue ; **Red** - toutes les interfaces ne sont pas disponibles ; **Gray** - au moins une interface est inconnue, mais aucune n'est indisponible. **Disponibilité des vérifications actives.** Si au moins une vérification active est activée sur l'hôte, les vérifications actives affectent également la disponibilité totale de l'interface Agent comme décrit ci-dessus. Pour déterminer la disponibilité des vérifications actives, des messages de heartbeat sont envoyés dans le thread de vérification active de l'agent. La fréquence des messages de heartbeat est contrôlée par le paramètre `HeartbeatFrequency` dans la configuration de Zabbix agent ou agent 2 (60 secondes par défaut, plage 0-3600). Les vérifications actives sont considérées comme indisponibles lorsque le heartbeat de vérification active date de plus de 2 x HeartbeatFrequency secondes. **Remarque :** les agents Zabbix antérieurs à la version 6.2.x n'envoient pas de heartbeats de vérification active, de sorte que la disponibilité de leurs hôtes reste inconnue.
Tags	Les tags de l'hôte sont affichés. Le survol ou le clic sur l'icône à trois points après les tags affichés ouvre une fenêtre contextuelle contenant tous les tags. Cliquer sur le bouton *Options* dans l'en-tête de la colonne vous permet de définir le nombre de tags affichés (1-3), de choisir la manière dont les noms de tags sont affichés (complets, abrégés ou aucun), de définir la priorité d'affichage des tags, ou de dupliquer la colonne (les colonnes dupliquées peuvent être renommées ou supprimées).
Tag value	Les valeurs des tags de l'hôte sont affichées. Cliquer sur le bouton *Options* dans l'en-tête de la colonne vous permet de renommer la colonne, de choisir quelles valeurs de tags afficher, ou de dupliquer la colonne (les colonnes dupliquées peuvent être renommées ou supprimées).

Column	Description
Status	État de l'hôte - *Enabled* ou *Disabled*. Cliquez sur l'en-tête de la colonne pour trier les hôtes par état par ordre croissant ou décroissant.
Latest data	Cliquer sur le lien ouvrira la section *Monitoring* > *Latest data* avec toutes les dernières données collectées depuis l'hôte. Le nombre d'éléments avec des dernières données est affiché en gris.
Problems	Le nombre de problèmes ouverts pour un hôte est affiché, trié par gravité. La couleur de chaque icône de problème indique la gravité, tandis que le nombre indique le nombre de problèmes pour cette gravité. Cliquer sur une icône ouvre *Monitoring* > *Problems* pour cet hôte ; si un hôte n'a aucun problème, un lien *Problems* est affiché à la place. Cliquer sur le bouton ⋮ *Options* dans l'en-tête de la colonne vous donne accès à l'option *Show suppressed problems*, qui inclut les problèmes qui seraient autrement supprimés (masqués) en raison de la maintenance de l'hôte ou d'une suppression de problème individuelle.
Graphs	Cliquer sur le lien affichera les graphiques configurés pour l'hôte. Le nombre de graphiques est affiché en gris. Si un hôte n'a aucun graphique, aucun lien n'est affiché.
Dashboards	Cliquer sur le lien affichera les tableaux de bord configurés pour l'hôte. Le nombre de tableaux de bord est affiché en gris. Si un hôte n'a aucun tableau de bord, aucun lien n'est affiché.
Web	Cliquer sur le lien affichera les scénarios web configurés pour l'hôte. Le nombre de scénarios web est affiché en gris. Si un hôte n'a aucun scénario web, aucun lien n'est affiché.
Custom text	Du texte personnalisé, des valeurs de macros ou une combinaison des deux est affiché. Cliquer sur le bouton ⋮ *Options* dans l'en-tête de la colonne vous permet de renommer la colonne, de saisir le texte personnalisé ou les macros, ou de dupliquer la colonne (les colonnes dupliquées peuvent être renommées ou supprimées). Macros prises en charge : `{HOST.*}` (sauf `{HOST.METADATA}`), `{INVENTORY.*}` et les macros utilisateur.

Boutons

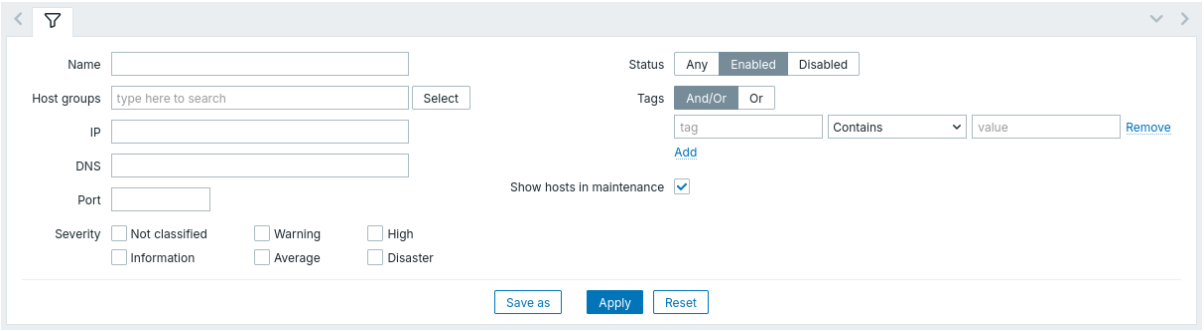
Créer un hôte vous permet de créer un nouvel hôte. Ce bouton est disponible uniquement pour les utilisateurs Admin et Super Admin.

Les boutons du mode d'affichage, communs à toutes les sections, sont décrits sur la page Surveillance.

Utilisation du filtre

Vous pouvez utiliser le filtre pour afficher uniquement les hôtes qui vous intéressent. Pour de meilleures performances de recherche, les données sont recherchées avec les macros non résolues.

Le filtre se trouve au-dessus du tableau. Il est possible de filtrer les hôtes par nom, groupe d'hôtes, IP ou DNS, port d'interface, tags, gravité du problème, statut (activé/désactivé/tous) ; vous pouvez également choisir d'afficher les hôtes actuellement en maintenance.



Parameter	Description
Name	Filtrer par nom d'hôte visible.
Host groups	Filtrer par un ou plusieurs groupes d'hôtes. La spécification d'un groupe d'hôtes parent sélectionne implicitement tous les groupes d'hôtes imbriqués.
IP	Filtrer par adresse IP.
DNS	Filtrer par nom DNS.

Parameter	Description
Port	Filtrer par numéro de port.
Severity	Filtrer par gravité du problème. Par défaut, les problèmes de toutes les gravités sont affichés. Les problèmes sont affichés s'ils ne sont pas supprimés.
Status	Filtrer par statut de l'hôte.
Tags	Filtrer par nom et valeur de tag d'hôte. Les hôtes peuvent être filtrés à l'aide des tags au niveau de l'hôte ainsi que des tags de tous les modèles liés, y compris les modèles imbriqués. Il est possible d'inclure ou d'exclure des tags et des valeurs de tag spécifiques. Plusieurs conditions peuvent être définies. La correspondance du nom du tag est toujours sensible à la casse. Plusieurs opérateurs sont disponibles pour chaque condition : **Exists** - inclure les noms de tag spécifiés **Equals** - inclure les noms et valeurs de tag spécifiés (sensible à la casse) **Contains** - inclure les noms de tag spécifiés lorsque les valeurs de tag contiennent la chaîne saisie (correspondance de sous-chaîne, insensible à la casse) **Does not exist** - exclure les noms de tag spécifiés **Does not equal** - exclure les noms et valeurs de tag spécifiés (sensible à la casse) **Does not contain** - exclure les noms de tag spécifiés lorsque les valeurs de tag contiennent la chaîne saisie (correspondance de sous-chaîne, insensible à la casse) Il existe deux types de calcul pour les conditions : **And/Or** - toutes les conditions doivent être remplies, les conditions ayant le même nom de tag seront regroupées par la condition Or **Or** - une seule condition suffit
Show hosts in maintenance	Cochez la case pour afficher les hôtes en maintenance (affiché par défaut).

Enregistrement du filtre

Les paramètres de filtre favoris peuvent être enregistrés sous forme d'onglets, puis accessibles rapidement en cliquant sur l'onglet correspondant au-dessus du filtre.

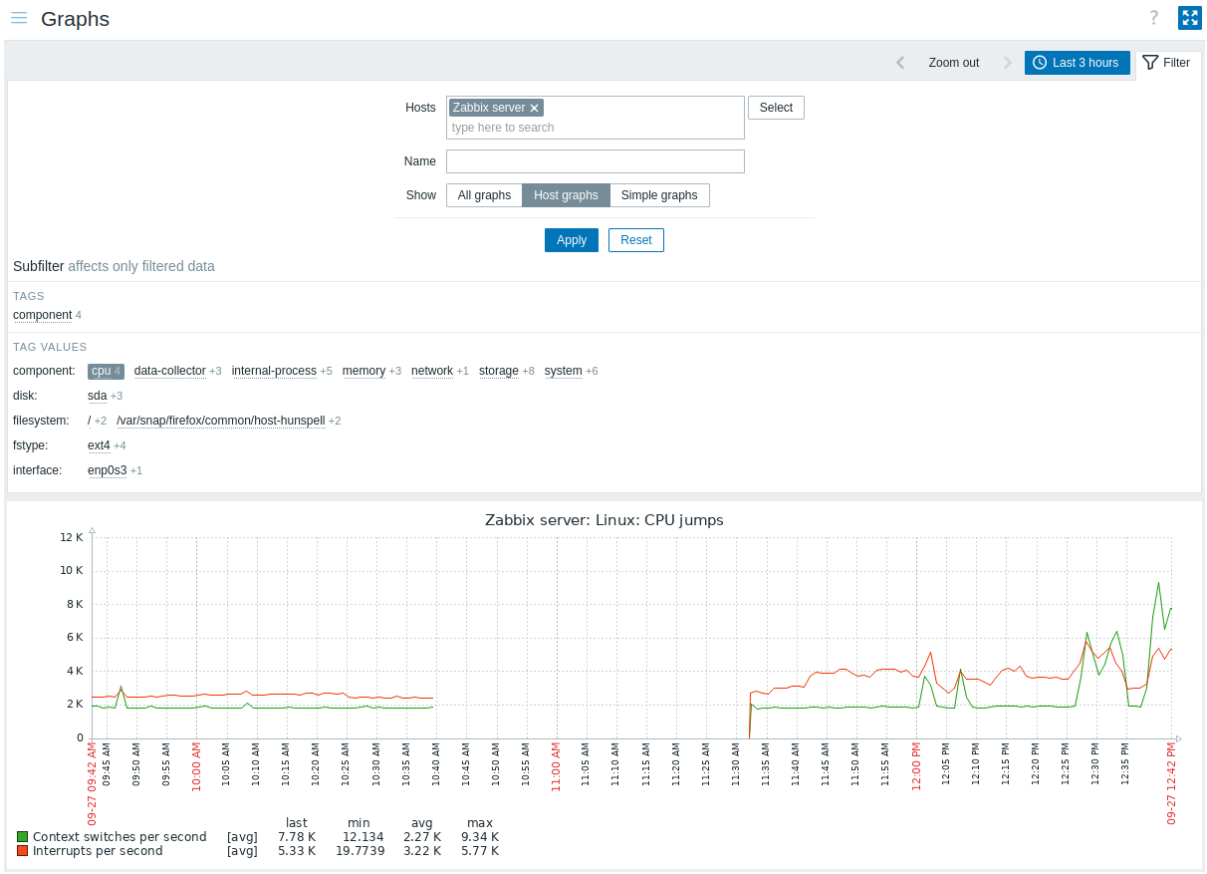
Voir plus de détails sur l'enregistrement des filtres.

Graphiques

Vue d'ensemble

Les graphiques de l'hôte sont accessibles depuis *Monitoring* > *Hôtes* en cliquant sur Graphiques pour l'hôte concerné.

Tout graphique personnalisé configuré pour l'hôte peut être affiché, ainsi que tout graphique simple.



Les graphiques sont triés par :

- nom du graphique (graphiques personnalisés)
- nom de l'élément (graphiques simples)

Les graphiques des hôtes désactivés sont également accessibles.

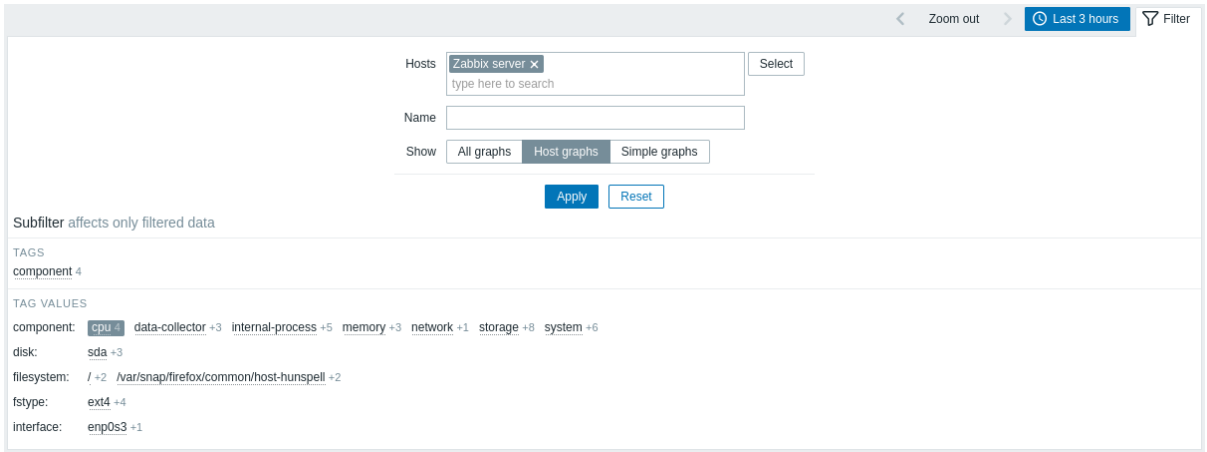
Sélecteur de période

Notez le sélecteur de période au-dessus du graphique. Il permet de sélectionner les périodes souvent requises en un clic de souris.

Voir aussi : *Sélecteur de période*

Utilisation du filtre

Pour afficher un graphique spécifique, sélectionnez-le dans le filtre. Le filtre vous permet de spécifier le hôte, le nom du graphique et l'option *Afficher* (tous les graphiques/graphiques de hôte/graphiques simples).



Si aucun hôte n'est sélectionné dans le filtre, aucun graphique ne s'affiche.

Utilisation du sous-filtre

Le sous-filtre permet de modifier davantage le filtrage appliqué par le filtre principal.

Il contient des liens cliquables permettant d'accéder rapidement aux graphiques associés. Les graphiques sont associés par une entité commune : le nom ou la valeur d'une étiquette. Lorsqu'un nom ou une valeur d'étiquette est sélectionné, il est mis en

évidence sur un arrière-plan gris et les graphiques sont immédiatement filtrés (il n'est pas nécessaire de cliquer sur *Appliquer* dans le filtre principal). Cliquer sur un autre nom ou une autre valeur d'étiquette l'ajoute aux résultats filtrés. Cliquer à nouveau sur le nom ou la valeur d'étiquette supprime le filtrage correspondant.

Les sous-filtres sont générés à partir des données filtrées, limitées à 1000 enregistrements. Si vous souhaitez afficher davantage d'enregistrements dans le sous-filtre, vous devez augmenter la valeur du paramètre *Limite pour les résultats de recherche et de filtrage* (dans *Administration* > *Général* > *Interface*).

Contrairement au filtre principal, le sous-filtre est mis à jour avec chaque demande d'actualisation du tableau afin d'obtenir en permanence des informations à jour sur les options de filtrage disponibles et leur nombre.

Le nombre d'entités affichées horizontalement est limité à 100. S'il y en a davantage, une icône représentant trois points est affichée à la fin ; elle n'est pas cliquable. Les listes verticales (telles que les étiquettes avec leurs valeurs) sont limitées à 20 entrées. S'il y en a davantage, une icône représentant trois points est affichée ; elle n'est pas cliquable.

Un nombre affiché à côté de chaque entité cliquable indique le nombre de graphiques qu'elle représente dans les résultats du filtre principal.

Lorsqu'une entité est sélectionnée, les nombres associés aux autres entités disponibles sont affichés avec un signe plus indiquant le nombre de graphiques pouvant être ajoutés à la sélection actuelle.

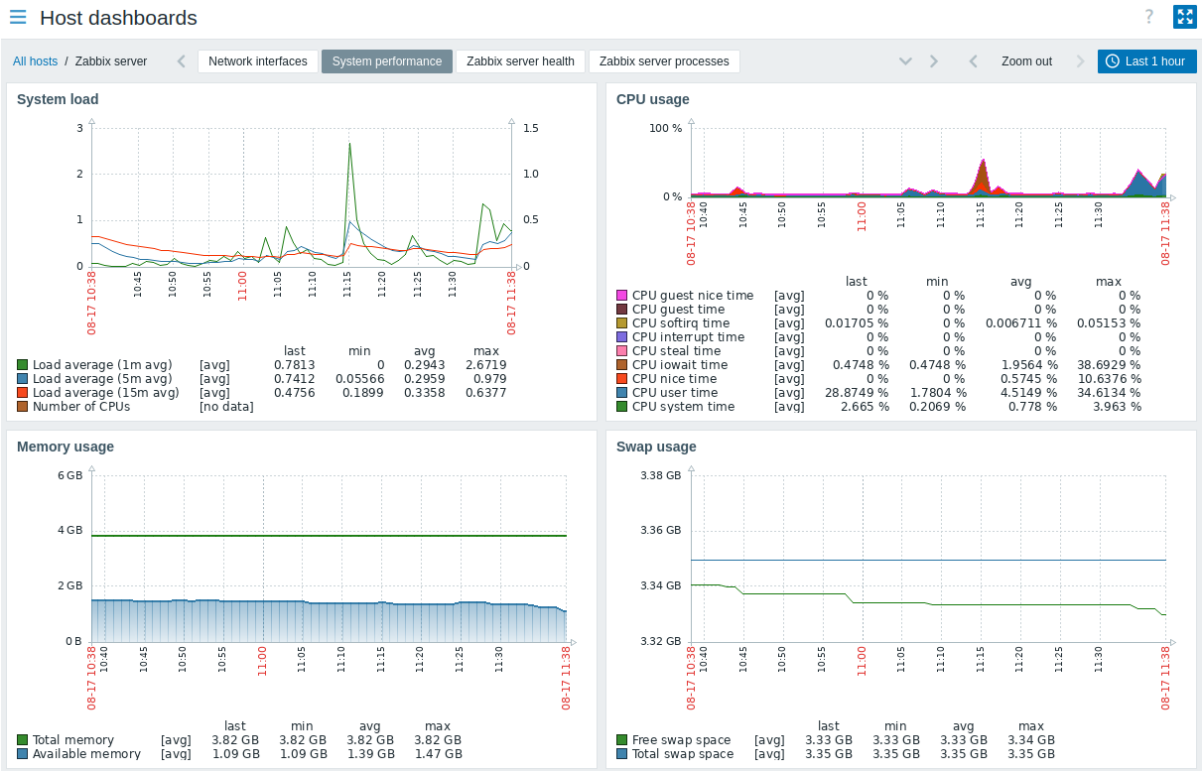
Boutons

Les boutons du mode d'affichage, communs à toutes les sections, sont décrits sur la page Monitoring.

Tableaux de bord de l'hôte

Présentation

Les tableaux de bord des hôtes ressemblent aux tableaux de bord globaux ; cependant, les tableaux de bord des hôtes n'ont pas de propriétaire et affichent uniquement les données de l'hôte sélectionné.



Lorsque vous consultez les tableaux de bord des hôtes, vous pouvez passer d'un tableau de bord configuré à l'autre en cliquant sur :

- les onglets des tableaux de bord ;
- les boutons fléchés  situés sous l'en-tête ;
- le bouton fléché  situé sous l'en-tête, qui affiche la liste complète des tableaux de bord des hôtes disponibles.

Pour accéder à la section *Supervision* > *Hôtes*, cliquez sur le lien de navigation *Tous les hôtes* situé sous l'en-tête, dans le coin supérieur gauche.

Configuration

Les tableaux de bord d'hôte sont configurés au niveau du template. Une fois qu'un template est lié à un hôte, des tableaux de bord d'hôte sont générés pour cet hôte. Notez que les tableaux de bord d'hôte *ne peuvent pas* être configurés dans la section *Dashboards*, qui est réservée aux tableaux de bord globaux.

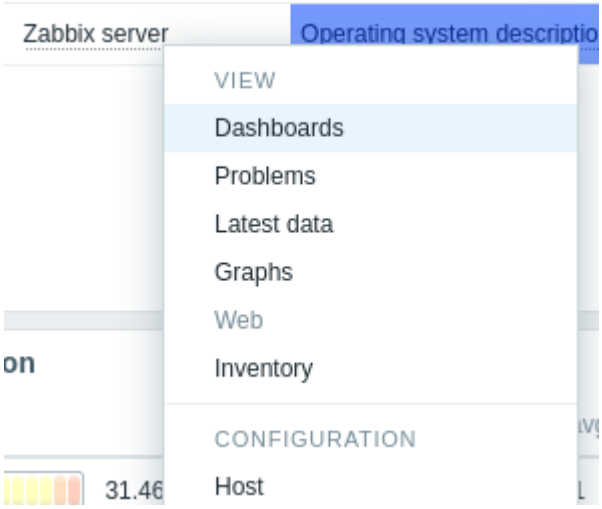
Les widgets des tableaux de bord d'hôte ne peuvent également être configurés qu'au niveau du template, à l'exception de la modification de l'intervalle d'actualisation. De plus, les widgets des tableaux de bord d'hôte ne peuvent être copiés vers d'autres tableaux de bord d'hôte qu'au sein du même template. Notez que les widgets des tableaux de bord globaux *ne peuvent pas* être copiés vers des tableaux de bord d'hôte.

Note:
Les tableaux de bord d'hôte étaient auparavant des écrans d'hôte avant Zabbix 5.2. Lors de l'importation d'un ancien template contenant des écrans, l'importation des écrans sera ignorée.

Accès

Les tableaux de bord d'hôte peuvent être consultés :

- après avoir recherché un nom d'hôte dans la recherche globale (cliquez sur le lien *Tableaux de bord* fourni dans les résultats de recherche) ;
- après avoir cliqué sur un nom d'hôte dans *Inventaire* > *Hôtes* (cliquez sur le lien *Tableaux de bord* fourni dans la vue d'ensemble de l'hôte) ;
- depuis le menu de l'hôte en cliquant sur *Tableaux de bord*.

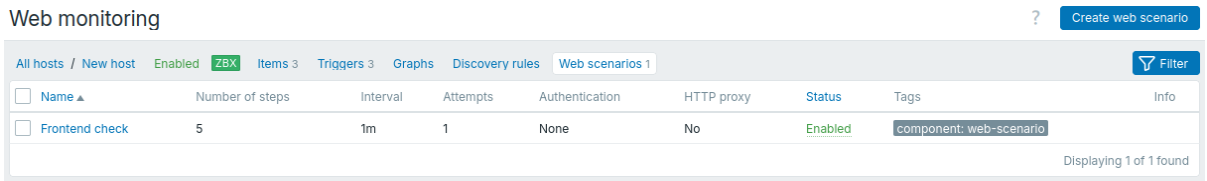


Notez que les tableaux de bord d'hôte *ne peuvent pas* être consultés directement dans la section *Tableaux de bord*, qui est réservée aux tableaux de bord globaux.

Scénarios web

Aperçu

Les informations sur l'scénario web de l'hôte peuvent être consultées depuis *Monitoring* > *Hôtes* en cliquant sur Web pour l'hôte concerné.

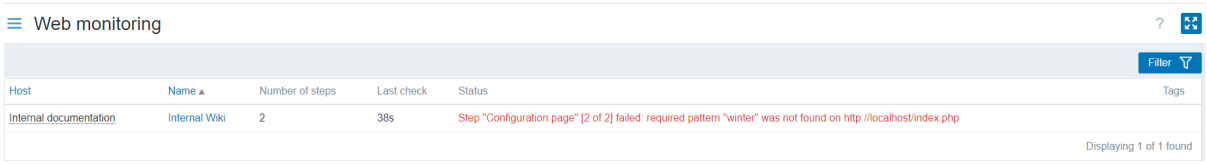


En cliquant sur le nom de l'hôte, vous ouvrez le menu de l'hôte. Les données des hôtes désactivés sont également accessibles. Le nom d'un hôte désactivé est affiché en rouge.

L'état d'un scénario web peut être :

- OK (vert) - toutes les étapes ont réussi et Zabbix a collecté des données pour le scénario.
- Erreur (rouge) - une ou plusieurs étapes ont échoué (voir le message d'erreur pour obtenir des indications sur les paramètres à corriger).

- Inconnu (gris) - aucun état d'étape n'est encore disponible ou aucune donnée n'a été collectée.



Le nombre maximal de scénarios affichés par page dépend du paramètre de profil utilisateur *Rows per page* setting.

Par défaut, seules les valeurs comprises dans les dernières 24 heures sont affichées. Cette limite a été introduite afin d'améliorer les temps de chargement initiaux des grandes pages de dernières données. Vous pouvez étendre cette période en modifiant la valeur du paramètre *Max history display period* dans la section de menu *Administration > General > GUI*.

Le nom du scénario est un lien vers des statistiques plus détaillées à son sujet :



Utilisation du filtre

La page affiche une liste de tous les scénarios web de l'hôte sélectionné. Pour afficher les scénarios web d'un autre hôte ou groupe d'hôtes sans revenir à la page *Monitoring > Hosts*, sélectionnez cet hôte ou ce groupe dans le filtre. Vous pouvez également filtrer les scénarios en fonction des tags.

Boutons

Les boutons du mode d'affichage, communs à toutes les sections, sont décrits sur la page Monitoring.

Dernières données

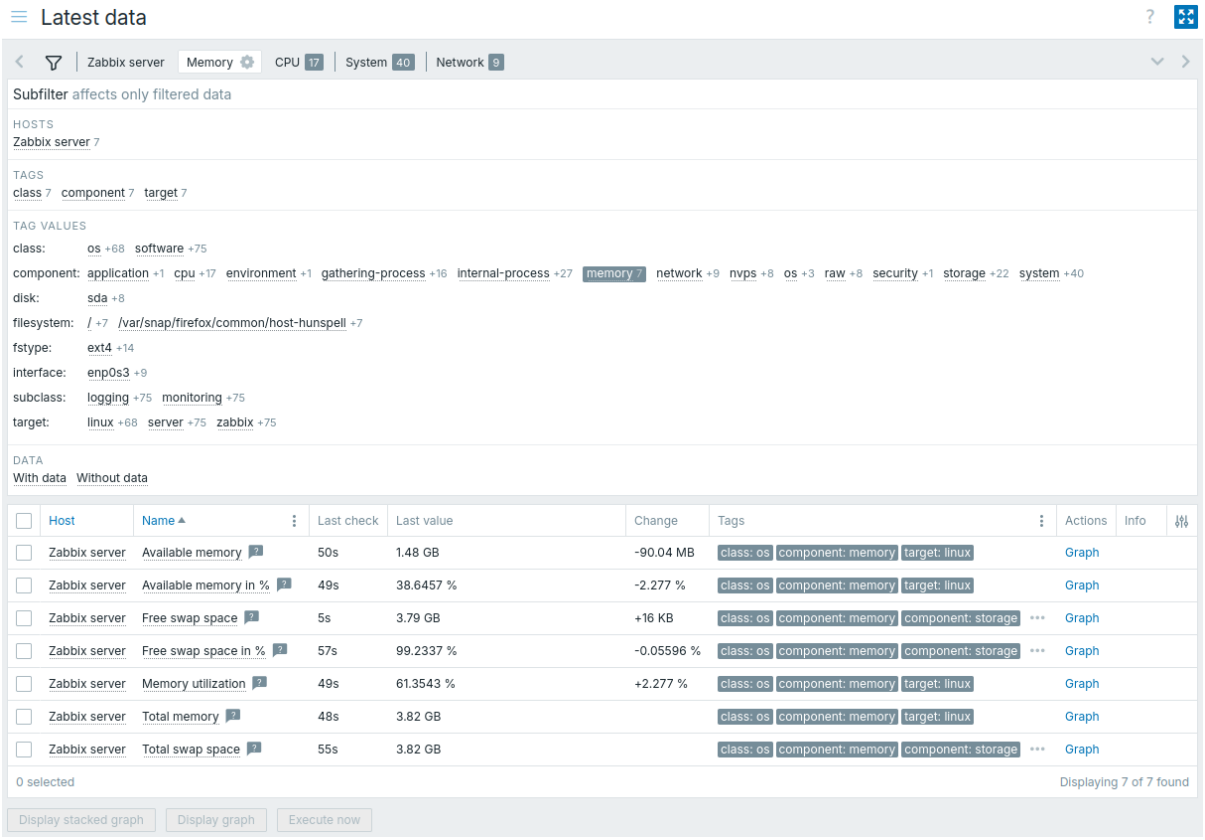
Vue d'ensemble

La section *Monitoring > Latest data* affiche les dernières valeurs collectées par les éléments.

Cette section contient les éléments suivants:

- Filtre
- Sous-filtre
- Liste des éléments

Note:
Le sous-filtre et la liste des éléments ne s'affichent que si le filtre est défini et qu'il existe des résultats à afficher.



Les détails de la dernière valeur sont affichés dans un tableau personnalisable. Le bouton *Personnaliser le tableau* dans l'en-tête du tableau vous permet d'afficher, de masquer ou de réorganiser les colonnes, ainsi que de rétablir la disposition par défaut. Vous pouvez également redimensionner chaque colonne en faisant glisser sa bordure droite; double-cliquez sur la bordure pour la réinitialiser. Toutes les personnalisations sont appliquées immédiatement et sont toujours mémorisées pour l'utilisateur actuel dans l'onglet filtre actif (par défaut ou enregistré).

Les colonnes suivantes sont disponibles:

Column	Description
Host	Nom de l'hôte auquel l'élément appartient. Cliquer sur le nom affiche le menu contextuel de l'hôte. Si un hôte est en maintenance, une icône de clé orange s'affiche après le nom de l'hôte. Si un hôte est désactivé, le nom de l'hôte s'affiche en rouge. Notez que les données des hôtes désactivés (y compris les graphiques et les listes de valeurs des éléments) sont accessibles dans la section *Latest data*.
Name	Nom de l'élément. Cliquer sur le nom affiche le menu de l'élément. Une icône de point d'interrogation s'affiche à côté du nom de l'élément pour tous les éléments qui ont une description. Survolez l'icône pour afficher une info-bulle contenant la description de l'élément. Cliquer sur le bouton *Options* dans l'en-tête de la colonne vous donne accès à l'option *Afficher la clé de l'élément* qui affiche la clé de l'élément sous le nom de chaque élément.
Interval	Intervalle de mise à jour de l'élément.
History	Période de conservation de l'historique de l'élément.
Trends	Période de conservation des tendances de l'élément.
Type	Type de l'élément.
Last check	Temps écoulé depuis la dernière vérification de l'élément.

Column	Description
Last value	Valeur la plus récente de l'élément. Les valeurs sont affichées avec la conversion d'unités et le mappage des valeurs appliqués. Survolez la valeur pour afficher les données brutes (tronquées à 8192 caractères). Les valeurs binaires sont affichées sous forme de texte indicatif (*binary value*) au lieu de la valeur réelle. Par défaut, seules les valeurs reçues au cours des dernières 24 heures sont affichées. Cette limite améliore les temps de chargement initiaux pour les grandes pages de données récentes; pour l'étendre, mettez à jour la valeur du paramètre *Max history display period* dans *Administration* > *General* > *GUI*.
Change	Différence entre la valeur précédente et la valeur la plus récente. Pour les éléments dont la fréquence de mise à jour est d'un jour ou plus, le montant du changement ne sera jamais affiché (avec le paramètre par défaut). Dans ce cas, la dernière valeur ne sera pas affichée du tout si elle a été reçue il y a plus de 24 heures.
Tags	Tags associés à l'élément. Cliquer sur un tag l'active dans le sous-filtre, filtrant ainsi la liste des éléments par ce tag (et par tout autre tag sélectionné dans le sous-filtre). Une fois cliqués, ces tags ne sont plus cliquables; pour ajuster le filtrage par tags, il faut utiliser le sous-filtre. Cliquer sur le bouton *Options* dans l'en-tête de la colonne vous permet de définir le nombre de tags affichés (1-3), de choisir la manière dont les noms des tags sont affichés (complets, abrégés ou aucun), de définir la priorité d'affichage des tags, ou de dupliquer la colonne (les colonnes dupliquées peuvent être renommées ou supprimées).
Tag value	Les valeurs des tags de l'élément sont affichées. Cliquer sur le bouton *Options* dans l'en-tête de la colonne vous permet de renommer la colonne, de choisir quelles valeurs de tags afficher, ou de dupliquer la colonne (les colonnes dupliquées peuvent être renommées ou supprimées).
Actions	Lien vers le graphique simple/historique des valeurs de l'élément.
Info	Informations supplémentaires sur l'élément. Si un élément présente des erreurs (par exemple, s'il est devenu non pris en charge), une icône d'information s'affiche. Survolez l'icône pour obtenir des détails.
Custom text	Un texte personnalisé, des valeurs de macros ou une combinaison des deux est affiché. Cliquer sur le bouton *Options* dans l'en-tête de la colonne vous permet de renommer la colonne, de saisir le texte personnalisé ou les macros, ou de dupliquer la colonne (les colonnes dupliquées peuvent être renommées ou supprimées). Macros prises en charge: `{HOST.*}` (sauf `{HOST.METADATA}`), `{INVENTORY.*}`, `{ITEM.*}` (sauf `{ITEM.STATE.ERROR}`), et macros utilisateur.

Boutons

Les boutons du mode d'affichage, communs à toutes les sections, sont décrits sur la page Monitoring.

Actions de masse

Les boutons sous la liste proposent des actions de masse avec un ou plusieurs éléments sélectionnés :

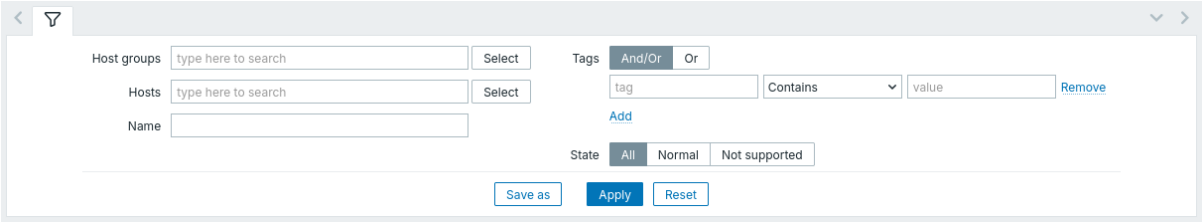
- *Afficher le graphique empilé* - afficher un graphique ad hoc empilé.
- *Afficher le graphique* - afficher un simple graphique ad hoc.
- *Exécuter maintenant* - exécuter immédiatement une vérification pour les nouvelles valeurs d'élément. Pris en charge uniquement pour les vérifications **passives** (voir plus de détails). Cette option est disponible uniquement pour les hôtes avec un accès en lecture-écriture. L'accès à cette option pour les hôtes avec des autorisations en lecture seule dépend de l'option de rôle utilisateur appelée *Invoked "Execute now" on read-only hosts*.

Pour utiliser ces options, cochez les cases devant les éléments concernés, puis cliquez sur le bouton requis.

Utilisation du filtre

Vous pouvez utiliser le filtre pour afficher uniquement les éléments qui vous intéressent. Pour améliorer les performances de recherche, les données sont recherchées avec les macros non résolues.

L'icône du filtre  se trouve au-dessus de la liste des éléments et du sous-filtre. Cliquez dessus pour développer le filtre.



Le filtre vous permet de restreindre la liste des éléments par groupe d'hôtes, hôte, nom d'élément, étiquettes et état. La spécification d'un groupe d'hôtes parent dans le filtre sélectionne implicitement tous les groupes d'hôtes imbriqués. Consultez *Monitoring* > *Problems* pour plus d'informations sur le filtrage par étiquettes.

L'option de filtre *Afficher les détails* permet de développer les informations affichées pour les éléments. Sélectionnez-la pour afficher des détails tels que l'intervalle d'actualisation de l'élément, les paramètres de l'historique et des tendances, le type d'élément et les erreurs de l'élément (correct/ non pris en charge).

Enregistrement du filtre

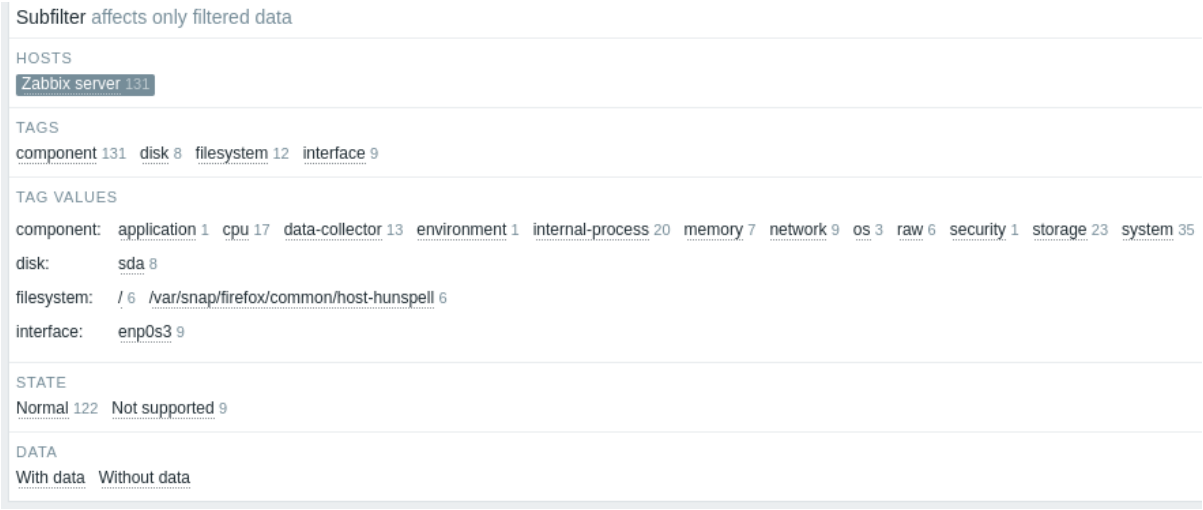
Les paramètres du filtre favori peuvent être enregistrés sous forme d'onglets, puis rapidement accessibles en cliquant sur l'onglet correspondant au-dessus du filtre.

Voir plus de détails sur l'enregistrement des filtres.

Utilisation du sous-filtre

Le sous-filtre permet de modifier davantage le filtrage défini dans le filtre principal.

Il contient des liens cliquables permettant d'accéder rapidement aux éléments associés. Les éléments sont associés par une entité commune : hôte, nom ou valeur de balise, état de l'élément ou état des données. Lorsqu'une entité est sélectionnée, elle est mise en évidence par un arrière-plan gris et les éléments sont immédiatement filtrés (il n'est pas nécessaire de cliquer sur *Appliquer* dans le filtre principal). La sélection d'une autre entité l'ajoute aux résultats filtrés. La sélection répétée de l'entité supprime le filtrage.



Les sous-filtres sont générés à partir des données filtrées, limitées à 1000 enregistrements. S'il existe 20 hôtes comportant chacun 100 éléments (soit 2000 enregistrements au total), seuls la moitié des hôtes seront visibles dans le sous-filtre. Pour afficher davantage d'enregistrements dans le sous-filtre, vous devez augmenter la valeur du paramètre *Limite des résultats de recherche et de filtrage* (dans *Administration* > *Général* > *Interface graphique*).

Contrairement au filtre principal, le sous-filtre est mis à jour à chaque demande d'actualisation du tableau afin de toujours afficher les informations à jour sur les options de filtrage disponibles et leur nombre.

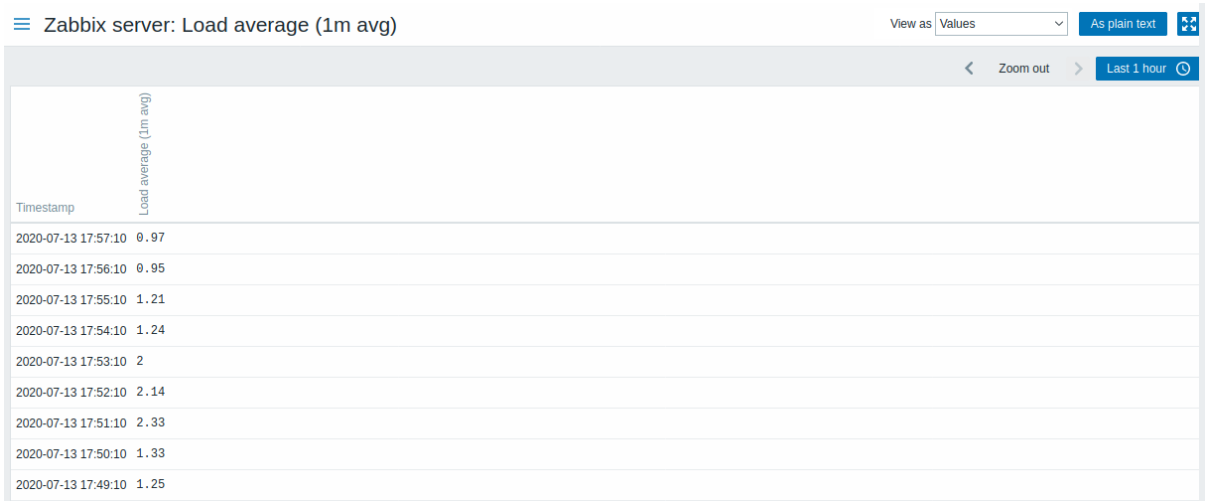
Pour chaque groupe d'entités (hôtes, balises, valeurs de balise, etc.), jusqu'à 10 lignes d'entités sont affichées. S'il existe davantage d'entités, cette liste peut être développée pour afficher un maximum de 1000 entrées (valeur de SUBFILTER_VALUES_PER_GROUP dans les définitions de l'interface) en cliquant sur l'icône à trois points ••• à la fin de la liste. Pour les *valeurs de balise*, la liste peut être développée pour afficher un maximum de 200 noms de balises avec leurs valeurs correspondantes. Notez qu'une fois entièrement développée, la liste ne peut pas être réduite.

Un nombre situé à côté de chaque entité cliquable indique le nombre d'éléments qui lui sont associés (en fonction des résultats du filtre principal). Lorsqu'une entité est sélectionnée, les nombres associés aux autres entités disponibles s'affichent avec un signe plus indiquant combien d'éléments peuvent être ajoutés à la sélection actuelle. Les entités sans élément ne sont pas affichées, sauf si elles ont été sélectionnées auparavant dans le sous-filtre.

Graphiques et historique

La colonne *Graph/History* dans la liste des éléments propose les liens suivants :

- **History** - pour tous les éléments textuels, menant à des listes (*Values/500 latest values*) affichant l'historique des valeurs précédentes de l'élément.
- **Graph** - pour tous les éléments numériques, menant à un graphique simple. Notez que lorsque le graphique est affiché, une liste déroulante en haut à droite offre également la possibilité de basculer vers *Values/500 latest values*.



Les valeurs affichées dans cette liste sont brutes, c'est-à-dire qu'aucun post-traitement n'est appliqué.

Note:
Le nombre total de valeurs affichées est défini par la valeur du paramètre *Limit for search and filter results*, configuré dans *Administration > General > GUI*.

Cartes

Vue d'ensemble

Dans la section *Monitoring > Maps*, vous pouvez configurer, gérer et afficher des cartes réseau.

Lorsque vous ouvrez cette section, vous verrez soit la dernière carte à laquelle vous avez accédé, soit la liste de toutes les cartes auxquelles vous avez accès.

Toutes les cartes peuvent être publiques ou privées. Les cartes publiques sont disponibles pour tous les utilisateurs, tandis que les cartes privées ne sont accessibles qu'à leur propriétaire et aux utilisateurs avec lesquels la carte est partagée.

Liste des cartes



Données affichées :

Column	Description
Name	Nom de la carte. Cliquez sur le nom pour afficher la carte.
Width	La largeur de la carte est affichée.
Height	La hauteur de la carte est affichée.
Actions	Deux actions sont disponibles : **Properties** - définir les propriétés générales de la carte **Edit** - accéder à la grille pour ajouter des éléments de carte

Pour configurer une nouvelle carte, cliquez sur le bouton *Create map* dans le coin supérieur droit. Pour importer une carte depuis un fichier YAML, XML ou JSON, cliquez sur le bouton *Import* dans le coin supérieur droit. L'utilisateur qui importe la carte en deviendra le propriétaire.

Deux boutons sous la liste offrent plusieurs options de modification en masse :

- *Export* - exporter les cartes vers un fichier YAML, XML ou JSON
- *Delete* - supprimer les cartes

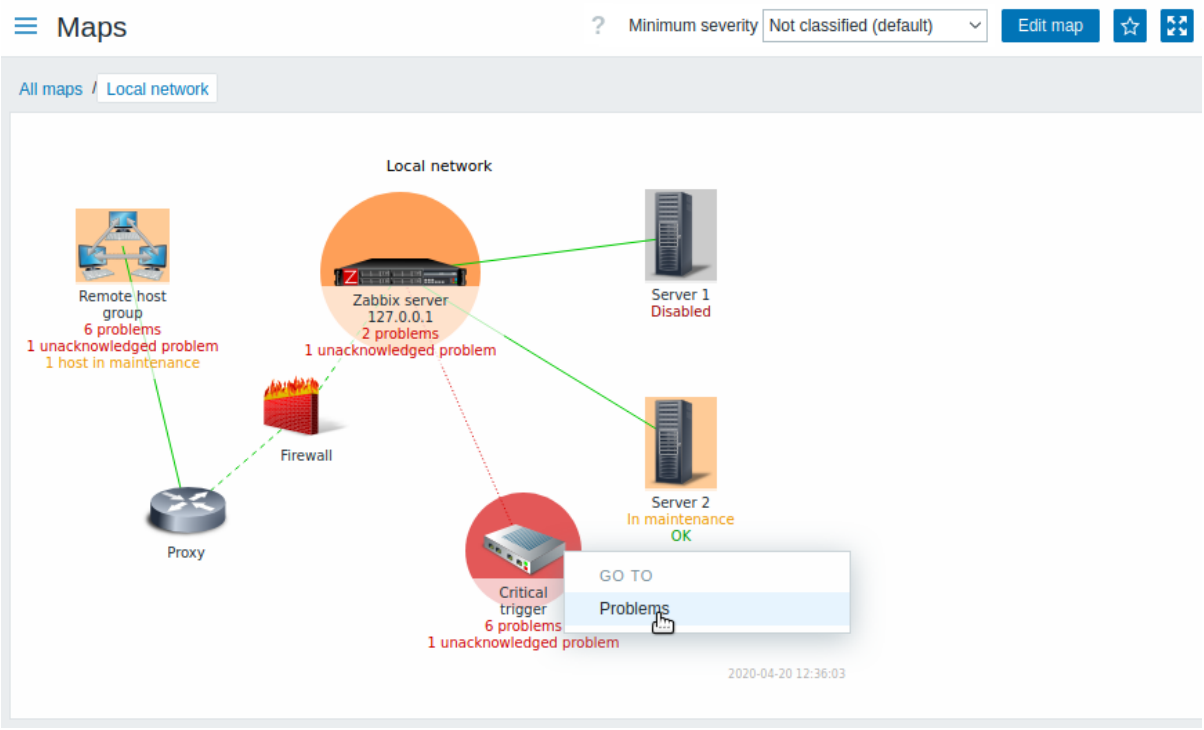
Pour utiliser ces options, cochez les cases devant les cartes concernées, puis cliquez sur le bouton requis.

Utilisation du filtre

Vous pouvez utiliser le filtre pour afficher uniquement les cartes qui vous intéressent. Pour de meilleures performances de recherche, les données sont recherchées avec des macros non résolues.

Affichage des cartes

Pour afficher une carte, cliquez sur son nom dans la liste de toutes les cartes.



Vous pouvez utiliser la liste déroulante dans la barre de titre de la carte pour sélectionner le niveau de gravité le plus bas des déclencheurs de problème à afficher. La gravité marquée comme *par défaut* est le niveau défini dans la configuration de la carte. Si la carte contient une sous-carte, la navigation vers la sous-carte conservera la gravité de la carte de niveau supérieur (sauf si elle est *Non classé* ; dans ce cas, elle ne sera pas transmise à la sous-carte).

Mise en évidence des icônes

Si un élément de carte est dans un état de problème, il est mis en évidence par un cercle rond. La couleur de remplissage du cercle correspond à la couleur de gravité du problème. Seuls les problèmes dont le niveau de gravité est égal ou supérieur au niveau de gravité sélectionné seront affichés avec l'élément. Si tous les problèmes sont reconnus, une bordure verte épaisse autour du cercle est affichée.

De plus :

- un hôte en maintenance est mis en évidence par un carré orange rempli. Notez que la mise en évidence de la maintenance est prioritaire sur celle de la gravité du problème si l'élément de carte est un hôte.
- un hôte désactivé (non surveillé) est mis en évidence par un carré gris rempli.

La mise en évidence est affichée si la case à cocher *Mise en évidence des icônes* est cochée dans la configuration de la carte.

Indicateurs de changements récents

Les triangles rouges orientés vers l'intérieur autour d'un élément indiquent un changement récent de l'état d'un déclencheur, survenu au cours des 30 dernières minutes. Ces triangles s'affichent si la case à cocher *Marquer les éléments lors d'un changement d'état du déclencheur* est cochée dans la configuration de la carte.

Liens

Cliquer sur un élément de carte ouvre un menu avec certains liens disponibles. Cliquer sur le nom de l'hôte affiche le menu de l'hôte.

Boutons

Les boutons à droite offrent les options suivantes :

Edit map	Accéder à la modification du contenu de la carte.
☆	Ajouter la carte au widget des favoris dans Tableaux de bord.
★	La carte se trouve dans le widget des favoris dans Tableaux de bord. Cliquez pour retirer la carte du widget des favoris.

Les boutons du mode d'affichage, communs à toutes les sections, sont décrits sur la page Surveillance.

Résumé lisible dans les cartes

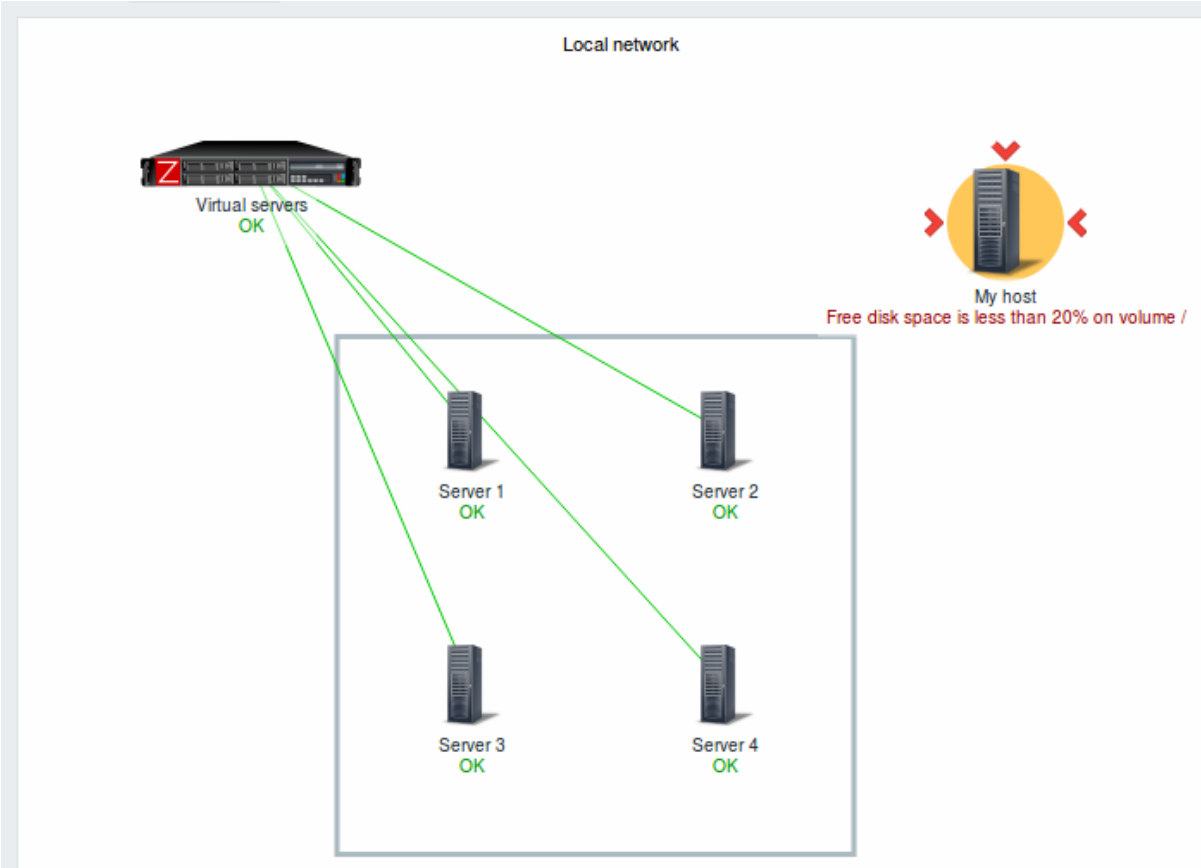
Une propriété « aria-label » masquée est disponible, permettant de lire les informations de la carte avec un lecteur d'écran. La description générale de la carte et la description de chaque élément sont disponibles au format suivant :

- pour la description de la carte : `<Nom de la carte>, <* sur * éléments dans un état de problème>, <* problèmes au total>.`
- pour décrire un élément avec un problème : `<Type d'élément>, État <état de l'élément>, <Nom de l'élément>, <Description du problème>.`
- pour décrire un élément avec plusieurs problèmes : `<Type d'élément>, État <état de l'élément>, <Nom de l'élément>, <* problèmes>.`
- pour décrire un élément sans problème : `<Type d'élément>, État <état de l'élément>, <Nom de l'élément>.`

Par exemple, la description suivante est disponible :

`'Réseau local, 1 sur 6 éléments dans un état de problème, 1 problème au total. Hôte, État problème, Mon hô`

pour la carte suivante :



Référencer une carte du réseau

Les cartes réseau peuvent être référencées par les paramètres GET `sysmapid` et `mapname`. Par exemple,

`http://zabbix/zabbix/zabbix.php?action=map.view&mapname=Local%20network`

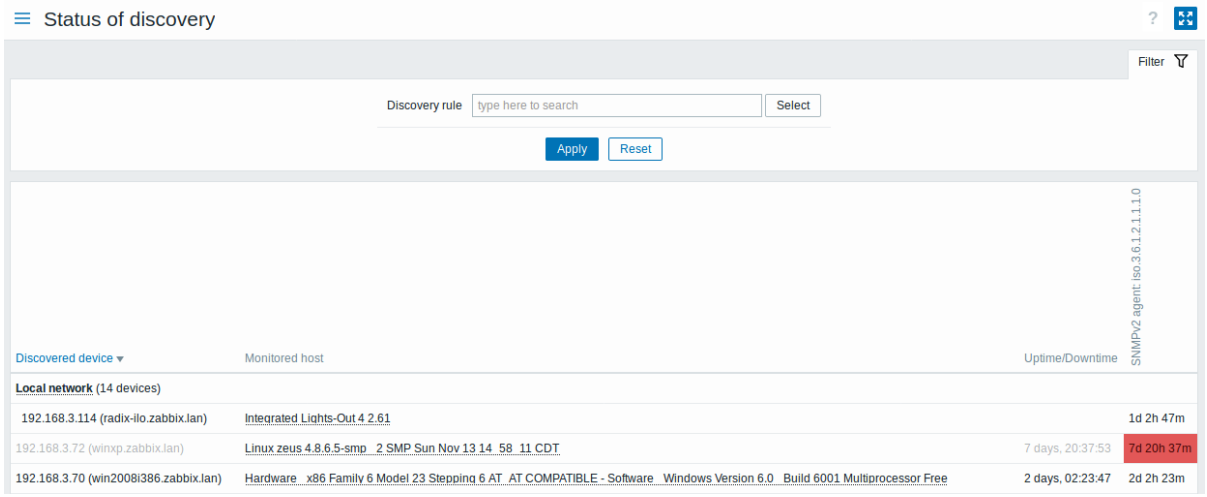
ouvrira la carte portant ce nom (Local network).

Si `sysmapid` (ID de carte) et `mapname` (nom de carte) sont spécifiés, `mapname` a une priorité plus élevée.

Découverte

Vue d'ensemble

Dans la section *Supervision* > *Découverte*, les résultats de la découverte réseau sont affichés. Les périphériques découverts sont triés selon la règle de découverte.



Données affichées :

Colonne	Description
Périphérique découvert	Les périphériques découverts sont répertoriés et regroupés selon la règle de découverte. Cliquer sur la règle de découverte ouvre le menu de la règle contenant le lien vers le formulaire de configuration de la règle de découverte. Si un périphérique possède plusieurs adresses IP (selon le *Critère d'unicité du périphérique* de la règle de découverte), elles sont répertoriées ensemble : l'adresse IP principale du périphérique en premier, suivie de ses autres adresses IP (avec indentation). L'adresse IP principale est l'adresse IP la plus basse associée à un hôte répertorié dans la colonne *Hôte supervisé* ou, s'il n'existe aucune adresse de ce type, l'adresse IP la plus basse du périphérique. Les périphériques peuvent être triés selon leur adresse IP principale, par ordre croissant ou décroissant ; les autres adresses IP d'un périphérique sont toujours répertoriées sous son adresse IP principale, par ordre croissant.
Hôte supervisé	Si un périphérique est déjà supervisé, le nom de l'hôte est répertorié dans cette colonne. Cliquer sur le nom de l'hôte ouvre le menu de l'hôte.
Temps de fonctionnement/Temps d'arrêt	La durée depuis la découverte du périphérique ou depuis sa perte après la découverte précédente est affichée dans cette colonne. Pour un périphérique possédant plusieurs adresses IP, la durée est affichée uniquement pour son adresse IP principale.
Vérification de découverte	L'état du service individuel (vérification de découverte) pour chaque périphérique découvert est affiché. Une cellule rouge indique que le service est indisponible. Le temps de fonctionnement ou d'arrêt du service est inclus dans la cellule. Cette colonne est affichée uniquement si le service a été trouvé sur au moins un périphérique découvert.

Note:
Le nombre de règles de découverte et le nombre total de périphériques découverts affichés dans cette section sont limités par le paramètre *Limite pour les résultats de recherche et de filtrage* dans *Administration > Général > Interface*. Les règles ou périphériques dépassant cette limite ne sont pas affichés, sans qu'aucune indication ne le signale.

Boutons

Les boutons du mode d'affichage, communs à toutes les sections, sont décrits sur la page Monitoring.

Utilisation du filtre

Vous pouvez utiliser le filtre pour afficher uniquement les règles de découverte qui vous intéressent. Pour de meilleures performances de recherche, les données sont recherchées avec les macros non résolues.

Lorsque rien n'est sélectionné dans le filtre, toutes les règles de découverte activées sont affichées. Pour sélectionner une règle de découverte spécifique à afficher, commencez à saisir son nom dans le filtre. Toutes les règles de découverte activées correspondantes seront □□□□ées pour la sélection. Plus d'une règle de découverte peut être sélectionnée.

APM

Vue d'ensemble

Le menu *APM* est destiné aux fonctionnalités de surveillance des performances des applications (APM) de Zabbix. Il affiche les données de télémétrie que vos applications et services envoient à l'aide d'OpenTelemetry, une norme ouverte pour la collecte de données de télémétrie. Ces données vous permettent de voir comment les requêtes circulent dans vos applications et services, de mesurer leurs performances et de trouver la cause des opérations lentes ou ayant échoué.

Le menu contient les sections suivantes, qui correspondent aux types de signaux OpenTelemetry :

- Traces
- Métriques
- Journaux

Contrairement à *Surveillance* > *Dernières données*, qui affiche les valeurs collectées par les éléments Zabbix, les sections *APM* affichent directement les données de télémétrie provenant de la source de données. Vous n'avez donc pas besoin de créer d'éléments. Pour afficher les données dans les sections *APM* :

1. Spécifiez une source de données de télémétrie dans *Administration* > *Source de données*.
2. Assurez-vous que les données de télémétrie sont écrites dans la source de données par un proxy Zabbix avec APM activé, ou par un collecteur OpenTelemetry. Notez que le serveur Zabbix ne peut pas recevoir de données de télémétrie ; seul le proxy Zabbix le peut.

Pour utiliser les données de télémétrie dans des déclencheurs ou des graphiques, utilisez l'élément requête de télémétrie.

Traces

Vue d'ensemble

La section *APM* > *Traces* affiche les traces collectées à partir de vos applications et services.

Une **trace** représente le chemin qu'une requête suit à travers vos applications et services. Par exemple, lorsqu'un utilisateur clique sur le bouton *Passer la commande* dans une boutique en ligne, la requête peut passer du service d'interface aux services de validation de commande et de paiement, puis à la base de données.

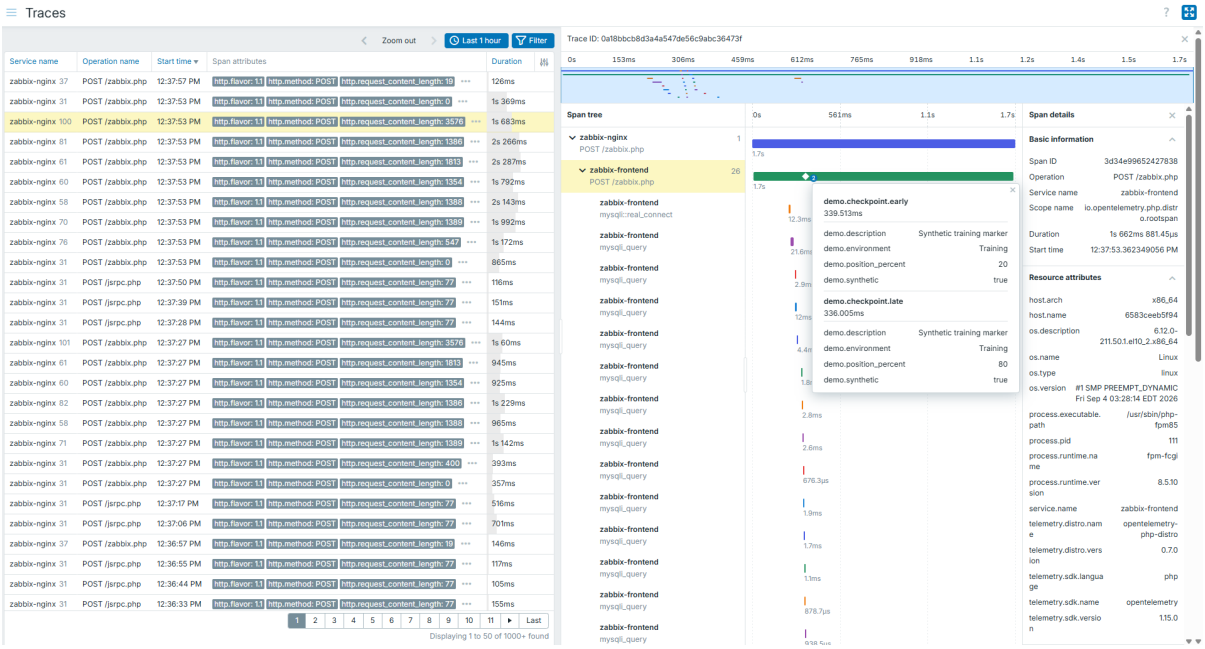
Une trace se compose de **spans**. Chaque span représente une opération unique, telle qu'une requête HTTP envoyée par le service d'interface au service de validation de commande, une requête de base de données qui enregistre la commande ou un appel de fonction qui calcule les frais d'expédition.

Les spans forment une hiérarchie. Le **span racine** représente la première opération de la requête. Chaque autre span est un **span enfant** du span pendant lequel son opération a été démarrée. Par exemple, le service d'interface qui reçoit la requête *Passer la commande* constitue le span racine, et le service de validation de commande qui traite la commande est son span enfant.

Chaque span est produit par un **service** (par exemple, `zabbix-frontend`; nommé par l'attribut de ressource OpenTelemetry `service.name`) et par un scope d'instrumentation, c'est-à-dire la partie du code qui l'a créé (par exemple, une bibliothèque qui trace les requêtes HTTP).

Pour fournir un contexte sur chaque opération, les spans possèdent également des **attributs** clé-valeur : les attributs de span décrivent l'opération elle-même (par exemple, `http.method: POST`), tandis que les attributs de ressource décrivent le service et l'environnement dans lequel il s'exécute (par exemple, `host.name: eabf8c2a5ddc`).

Pour plus d'informations sur les traces, consultez la documentation OpenTelemetry.



La section *APM* > *Traces* contient les éléments suivants :

- Bouton Mode kiosque - affiche uniquement le contenu de la page, sans le menu ni l'en-tête.
- Sélecteur de période - sélectionne la période pour laquelle afficher les traces.
- Filtre - affiche uniquement les traces qui vous intéressent.
- Liste des traces - permet de consulter les traces, décrites par leurs spans racines.
- Panneau de trace - permet de consulter tous les spans de la trace sélectionnée sur une chronologie.

Liste des traces

Chaque ligne de la liste des traces représente une trace ; les colonnes affichent les données de son span racine.

Service name	Operation name	Start time ▾	Span attributes	Duration
zabbix-nginx 42	POST /zabbix.php	03:40:28 PM	http.flavor: 1.1 http.method: POST http.request_content_length: 796	2s 983ms
zabbix-nginx 45	POST /zabbix.php	03:40:28 PM	http.flavor: 1.1 http.method: POST http.request_content_length: 827	2s 694ms
zabbix-nginx 41	POST /zabbix.php	03:40:28 PM	http.flavor: 1.1 http.method: POST http.request_content_length: 789	2s 595ms
zabbix-nginx 49	POST /zabbix.php	03:40:28 PM	http.flavor: 1.1 http.method: POST http.request_content_length: 1375	2s 964ms
zabbix-nginx 46	POST /zabbix.php	03:40:28 PM	http.flavor: 1.1 http.method: POST http.request_content_length: 1359	3s 75ms
zabbix-nginx 46	POST /zabbix.php	03:40:28 PM	http.flavor: 1.1 http.method: POST http.request_content_length: 1368	2s 983ms
zabbix-nginx 56	POST /zabbix.php	03:40:28 PM	http.flavor: 1.1 http.method: POST http.request_content_length: 1821	3s 160ms
zabbix-nginx 41	POST /zabbix.php	03:40:28 PM	http.flavor: 1.1 http.method: POST http.request_content_length: 790	2s 777ms
zabbix-nginx 46	POST /zabbix.php	03:40:28 PM	http.flavor: 1.1 http.method: POST http.request_content_length: 803	2s 992ms
zabbix-nginx 42	POST /zabbix.php	03:40:28 PM	http.flavor: 1.1 http.method: POST http.request_content_length: 793	2s 900ms

Les traces sont affichées dans un tableau personnalisable. Le bouton *Personnaliser le tableau* dans l'en-tête du tableau permet d'afficher, de masquer ou de réorganiser les colonnes, ainsi que de rétablir la disposition par défaut. Vous pouvez également redimensionner chaque colonne en faisant glisser sa bordure droite ; double-cliquez sur la bordure pour rétablir sa taille. Toutes les personnalisations sont appliquées immédiatement et mémorisées pour l'utilisateur actuel.

La liste des traces peut contenir les colonnes suivantes :

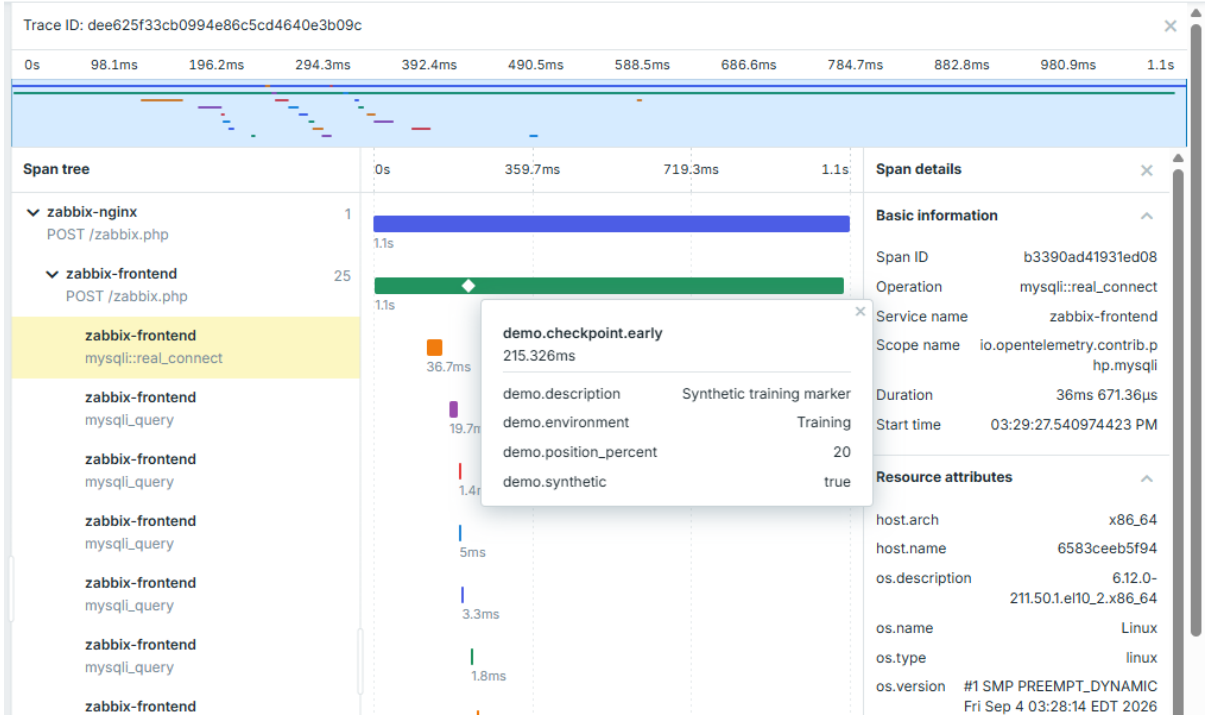
Note:
Chaque colonne correspond à une colonne de la table `otel_traces` de la source de données de télémétrie ; pour obtenir la liste des colonnes, consultez Colonnes de données de télémétrie prises en charge. La plupart des colonnes portent le même nom que dans la source de données (par exemple, *Trace ID* correspond à la colonne `TraceId`). Lorsque les noms diffèrent, la colonne de la source de données est indiquée dans le tableau ci-dessous.

Colonne	Description
Trace ID	ID de la trace, partagé par tous les spans appartenant à la trace.
Span ID	ID du span racine.

Colonne	Description
Trace state	Données supplémentaires, le cas échéant, que les outils de traçage associent à la trace (voir W3C Trace Context).
Service name	Service ayant produit le span racine, suivi du nombre de spans dans la trace. Si certains spans possèdent le code d'état *Error* (voir ci-dessous), une icône rouge indique leur nombre.
Operation name	Opération représentée par le span racine (par exemple, `POST /zabbix.php`). Colonne de la source de données : `SpanName`.
Span kind	Rôle du span racine : **Unspecified** - non spécifié. **Internal** - opération effectuée au sein d'une application. **Server** - traitement d'une requête entrante. **Client** - requête sortante. **Producer** - création d'une opération asynchrone (par exemple, l'envoi d'un message). **Consumer** - traitement d'une opération asynchrone.
Scope name	Scope d'instrumentation ayant produit le span racine.
Scope version	Version du scope d'instrumentation.
Status code	État de l'opération représentée par le span racine : **Unset** - aucun état explicite n'a été attribué au span racine. **Ok** - l'opération s'est terminée avec succès. **Error** - l'opération s'est terminée avec une erreur.
Status message	Informations supplémentaires, le cas échéant, sur l'état.
Start time	Date et heure auxquelles le span racine a commencé. Colonne de la source de données : `Timestamp`.
Span attributes	Attributs du span racine. Cliquez sur le bouton à trois points ••• pour afficher tous les attributs.
Resource attributes	Attributs de ressource du span racine. Cliquez sur le bouton à trois points ••• pour afficher tous les attributs.
Duration	Durée de la trace ; la barre grise l'affiche par rapport aux autres traces.

Panneau de trace

Cliquez sur une trace dans la liste des traces pour ouvrir le panneau de trace, qui affiche tous les spans de la trace sur une chronologie : les opérations effectuées, leur ordre d'exécution et la durée de chacune.

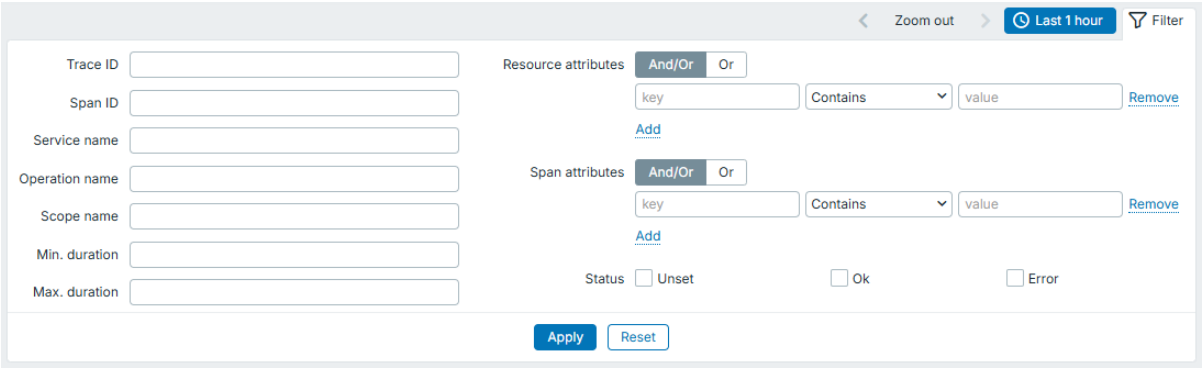


Le panneau de trace peut contenir les éléments suivants :

Élément	Description
ID de trace	ID de la trace sélectionnée.
Vue d'ensemble de la chronologie de la trace	La trace complète, avec les spans répartis sur un maximum de 8 lignes (plusieurs spans peuvent partager une ligne). Faites glisser les bordures de la plage sélectionnée, ou la plage elle-même, pour zoomer sur la chronologie des spans correspondant à cette partie de la trace.
Arborescence des spans	Les spans dans leur hiérarchie ; chacun affiche son service (en gras), le nom de l'opération et le nombre de spans enfants directs.
Chronologie des spans	Les spans sous forme de barres : leur position et leur longueur indiquent le moment où chaque span a commencé et sa durée. Les marqueurs indiquent les événements (par exemple, les exceptions) ; survolez un marqueur ou cliquez dessus pour afficher les détails.
Détails du span	Cliquez sur un span dans l'arborescence des spans pour afficher ses détails et ses attributs.

Utilisation du filtre

Vous pouvez utiliser le filtre pour afficher uniquement les traces qui vous intéressent.



Pour filtrer par attributs, saisissez la clé de l'attribut, sélectionnez l'opérateur, puis saisissez la valeur de l'attribut.

Les opérateurs suivants sont pris en charge :

- **Contient** - *(par défaut)* la valeur de l'attribut contient la chaîne saisie.
- **Est égal à** - la valeur de l'attribut est égale à la chaîne saisie.
- **Ne contient pas** - la valeur de l'attribut ne contient pas la chaîne saisie.
- **N'est pas égal à** - la valeur de l'attribut n'est pas égale à la chaîne saisie.
- **Existe** - l'attribut avec la clé saisie existe.
- **N'existe pas** - l'attribut avec la clé saisie n'existe pas.

Si vous ajoutez plusieurs conditions, sélectionnez la manière de les évaluer :

- **Et/Ou** - *(par défaut)* toutes les conditions doivent être remplies ; les conditions ayant la même clé d'attribut sont combinées avec *Ou*.
- **Ou** - au moins une condition doit être remplie.

Métriques

Vue d'ensemble

La section *APM* > *Métriques* affiche les points de données de métriques collectés à partir de vos applications et services.

Une **métrique** est une mesure (par exemple, la durée d'une requête, le nombre de commandes traitées ou l'utilisation de la mémoire). Les applications envoient leurs valeurs sous forme de **points de données** qui contiennent la valeur (ou les valeurs agrégées) mesurée à un moment précis. Une métrique est similaire à un élément Zabbix, et un point de données est similaire à une valeur de cet élément.

Un point de données peut être de l'un des types suivants :

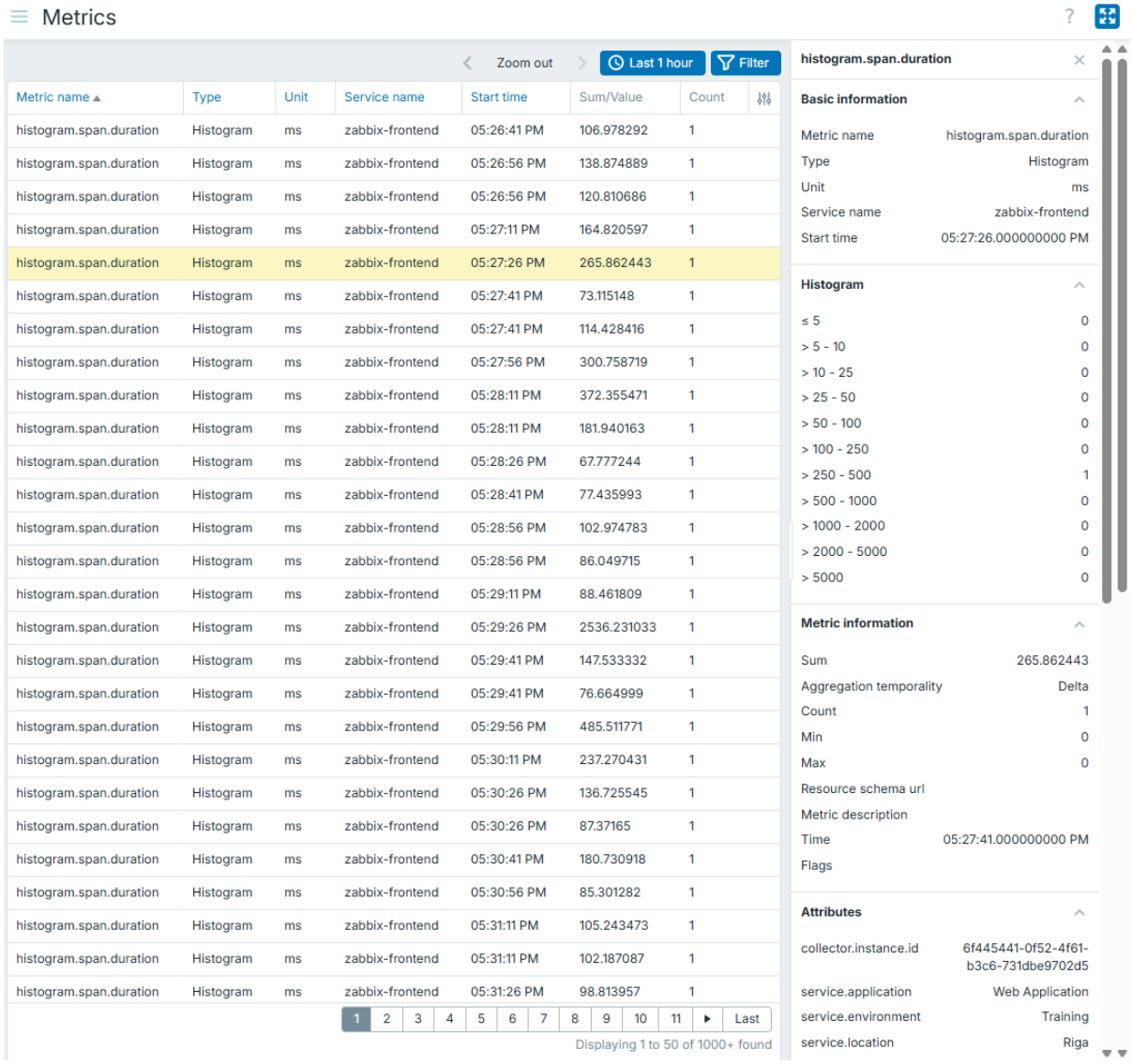
- **Jauge** - une valeur mesurée à un moment précis (par exemple, l'utilisation actuelle de la mémoire).
- **Somme** - une somme de mesures sur une période donnée (par exemple, le nombre total de requêtes).
- **Histogramme** - le nombre de mesures sur une période donnée qui se situent dans chacune de plusieurs plages de valeurs fixes, appelées compartiments (par exemple, le nombre de requêtes ayant duré entre 0 et 100 ms, entre 100 et 250 ms ou entre 250 et 500 ms).

- **Histogramme exponentiel** - similaire à un histogramme, mais chaque compartiment est plus large que le précédent selon le même facteur (par exemple, 1-2 ms, 2-4 ms, 4-8 ms).

Chaque point de données est produit par un **service** (par exemple, `zabbix-frontend`; nommé par l'attribut de ressource OpenTelemetry `service.name`) et par un périmètre d'instrumentation, c'est-à-dire la partie du code qui l'a créé (par exemple, une bibliothèque qui mesure la durée des requêtes HTTP).

Pour fournir un contexte sur chaque mesure, les points de données possèdent également des **attributs** clé-valeur : les attributs des points de données décrivent ce qui a été mesuré et permettent de distinguer les points de données d'une même métrique (par exemple, `http.request.method: POST`), tandis que les attributs de ressource décrivent le service et l'environnement dans lequel il s'exécute (par exemple, `host.name: eabf8c2a5ddc`).

Pour plus d'informations sur les métriques, consultez la documentation OpenTelemetry.



La section *APM* > *Métriques* contient les éléments suivants :

- Bouton Mode kiosque - affiche uniquement le contenu de la page, sans le menu ni l'en-tête.
- Sélecteur de période - sélectionne la période pour laquelle afficher les points de données.
- Filtre - affiche uniquement les points de données qui vous intéressent.
- Liste des métriques - permet de consulter les points de données.
- Détails de la métrique - permet de consulter les détails du point de données sélectionné.

Liste des métriques

Chaque ligne de la liste des métriques représente un point de données.

Metric name	Type	Unit	Service name	Start time ▾	Sum/Value	Cou...
histogram.span.duration	Histogram	ms	zabbix-frontend	02:28:15 PM	71.575147	1
histogram.span.duration	Histogram	ms	zabbix-frontend	02:28:15 PM	18.663078	28
histogram.span.duration	Histogram	ms	zabbix-nginx	02:28:15 PM	154	2
histogram.span.duration	Histogram	ms	zabbix-nginx	02:28:15 PM	5704	8
exponential.span.duration	Exponential histogram	ms	zabbix-frontend	02:28:15 PM	18.663078	28
exponential.span.duration	Exponential histogram	ms	zabbix-frontend	02:28:15 PM	338.947032	1
exponential.span.duration	Exponential histogram	ms	zabbix-frontend	02:28:15 PM	71.575147	1
exponential.span.duration	Exponential histogram	ms	zabbix-frontend	02:28:15 PM	2.278072	1
exponential.span.duration	Exponential histogram	ms	zabbix-nginx	02:28:15 PM	154	2
exponential.span.duration	Exponential histogram	ms	zabbix-nginx	02:28:15 PM	5704	8
nginx.connections_accepted	Summary	{connections}	Web Application	02:03:30 AM	2345	
nginx.connections_current	Summary	{connections}	Web Application	02:03:30 AM	10	

Les points de données sont affichés dans un tableau personnalisable. Le bouton *Personnaliser le tableau* dans l'en-tête du tableau permet d'afficher, de masquer ou de réorganiser les colonnes, ainsi que de rétablir la disposition par défaut. Vous pouvez également redimensionner chaque colonne en faisant glisser sa bordure droite ; double-cliquez sur la bordure pour rétablir sa taille. Toutes les personnalisations sont appliquées immédiatement et mémorisées pour l'utilisateur actuel.

La liste des métriques peut contenir les colonnes suivantes :

Note:
Chaque colonne, à l'exception de *Type*, correspond à une colonne des tables de métriques (`otel_metrics_gauge`, `otel_metrics_sum`, `otel_metrics_histogram` et `otel_metrics_exponential_histogram`) de la source de données de télémétrie ; pour obtenir la liste des colonnes, consultez Colonnes de données de télémétrie prises en charge. La plupart des colonnes portent le même nom que dans la source de données (par exemple, *Nom de la métrique* correspond à la colonne `MetricName`). Lorsque les noms diffèrent, la colonne de la source de données est indiquée dans le tableau ci-dessous.

Colonne	Description
Nom de la métrique	Métrique à laquelle appartient le point de données (par exemple, `http.server.duration`).
Type	Type du point de données : *Gauge*, *Sum*, *Histogram* ou *Exponential histogram*. Déterminé par la table de métriques qui contient le point de données.
Unité	Unité éventuelle des valeurs de la métrique (par exemple, `s` ou `By` ; voir UCUM). Colonne de la source de données : `MetricUnit`.
Nom du service	Service ayant produit le point de données.
Nom de l'étendue	Étendue d'instrumentation ayant produit le point de données.
Heure de début	Date et heure auxquelles la période de mesure du point de données a commencé. Colonne de la source de données : `StartTimeUnix`.
Somme/Valeur	Somme de toutes les mesures (*Histogram*, *Exponential histogram*), ou valeur (*Gauge*, *Sum*) du point de données.
Nombre	Nombre de mesures dans un point de données de type *Histogram* ou *Exponential histogram*.
Heure	Date et heure du point de données ; pour les points de données de type *Sum*, *Histogram* et *Exponential histogram*, fin de la période de mesure. Colonne de la source de données : `TimeUnix`.
Indicateurs	Indicateurs du point de données : **0** - aucun indicateur n'est défini. **1** - aucune valeur n'a été enregistrée pour le point de données.
Temporalité de l'agrégation	Mode d'agrégation des valeurs d'un point de données de type *Sum*, *Histogram* ou *Exponential histogram* au fil du temps : **Unspecified** - non spécifiée. **Delta** - agrégées sur la période écoulée depuis le point de données précédent. **Cumulative** - agrégées sur la période écoulée depuis l'heure de début.
Description de la métrique	Description éventuelle de la métrique.
Attributs	Attributs du point de données. Cliquez sur le bouton à trois points pour afficher tous les attributs.

Colonne	Description
Attributs de la ressource	Attributs de ressource du point de données. Cliquez sur le bouton à trois points ••• pour afficher tous les attributs.
Attributs de l'étendue	Attributs de l'étendue d'instrumentation. Cliquez sur le bouton à trois points ••• pour afficher tous les attributs.

Détails de la métrique

Cliquez sur un point de données dans la liste des métriques pour ouvrir le panneau des détails de la métrique, qui affiche toutes les informations relatives au point de données.

exponential.span.calls

Basic information

Metric name	exponential.span.calls
Type	Summary
Unit	
Service name	zabbix-frontend
Start time	02:22:15.000000000 PM

Metric information

Value	1
Aggregation temporality	Delta
Is monotonic	true
Resource schema url	
Metric description	
Time	02:22:30.000000000 PM
Flags	

Attributes

Resource attributes

Scope attributes

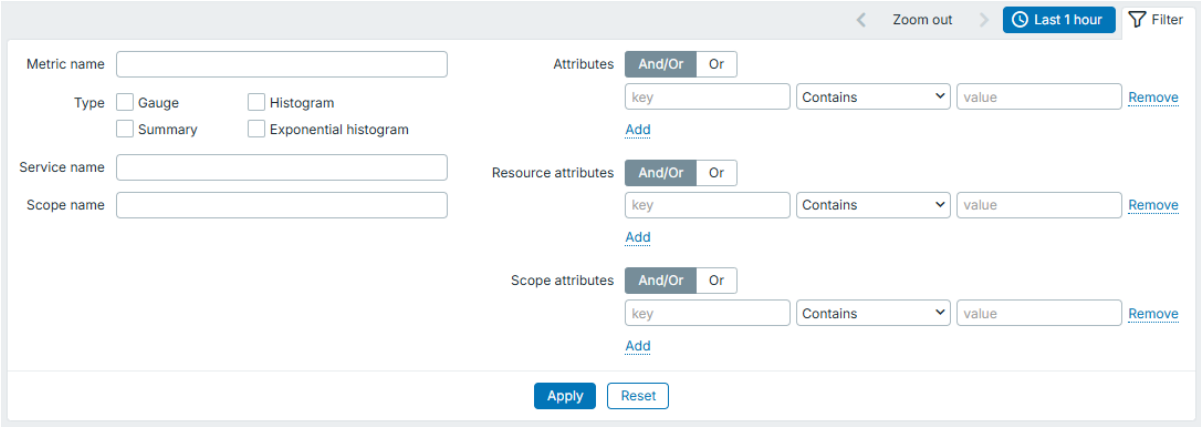
Scope

Le panneau des détails de la métrique peut contenir les sections suivantes :

Section	Description
Informations de base	Nom de la métrique, type, unité, nom du service et heure de début.
Histogramme	Compartiments (par exemple, `> 5 - 10`), chacun indiquant le nombre de mesures qu'il contient. Affiché uniquement pour les points de données de type *Histogramme*.
Histogramme exponentiel	Compartiments (par exemple, `>= 249.7885 - 249.7887`), chacun indiquant le nombre de mesures qu'il contient. Affiché uniquement pour les points de données de type *Histogramme exponentiel*.
Informations sur la métrique	Valeurs du point de données, qui dépendent de son type, ainsi que la description de la métrique, l'heure, les indicateurs et l'URL du schéma de ressource.
Attributs	Attributs du point de données.
Attributs de ressource	Attributs de ressource.
Attributs de portée	Attributs éventuels de la portée d'instrumentation.
Portée	Nom, version et URL du schéma de la portée d'instrumentation.

Utilisation du filtre

Vous pouvez utiliser le filtre pour afficher uniquement les points de données qui vous intéressent.



Pour filtrer par attributs, saisissez la clé de l'attribut, sélectionnez l'opérateur et saisissez la valeur de l'attribut.

Les opérateurs suivants sont pris en charge :

- **Contient** - *(par défaut)* la valeur de l'attribut contient la chaîne saisie.
- **Est égal à** - la valeur de l'attribut est égale à la chaîne saisie.
- **Ne contient pas** - la valeur de l'attribut ne contient pas la chaîne saisie.
- **N'est pas égal à** - la valeur de l'attribut n'est pas égale à la chaîne saisie.
- **Existe** - l'attribut avec la clé saisie existe.
- **N'existe pas** - l'attribut avec la clé saisie n'existe pas.

Si vous ajoutez plusieurs conditions, sélectionnez la manière de les évaluer :

- **Et/Ou** - *(par défaut)* toutes les conditions doivent être remplies ; les conditions ayant la même clé d'attribut sont combinées avec *Ou*.
- **Ou** - au moins une condition doit être remplie.

Journaux

Vue d'ensemble

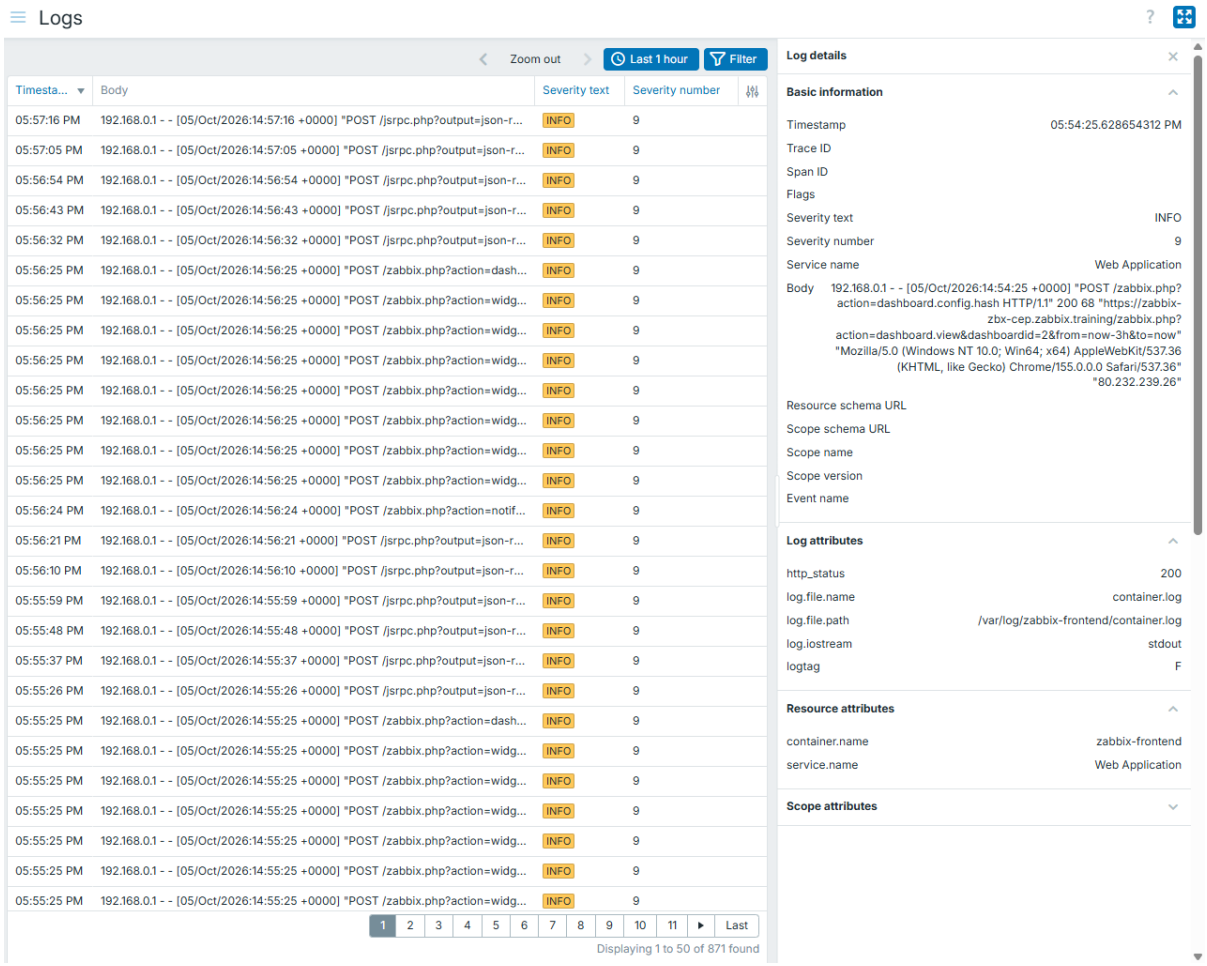
La section *APM > Journaux* affiche les enregistrements de journaux collectés à partir de vos applications et services.

Un **enregistrement de journal** est un enregistrement textuel d'un événement (par exemple, un paiement échoué ou un délai d'attente de la base de données). Chaque enregistrement de journal possède une gravité et un message (corps). Si un enregistrement de journal a été produit lors d'une opération faisant partie d'une trace, il contient également les identifiants de cette trace et de ce span.

Chaque enregistrement de journal est produit par un **service** (par exemple, `zabbix-frontend`; nommé par l'attribut de ressource OpenTelemetry `service.name`) et par un scope d'instrumentation, c'est-à-dire la partie du code qui l'a créé (par exemple, une bibliothèque de journalisation).

Pour fournir un contexte sur chaque événement, les enregistrements de journaux possèdent également des **attributs** clé-valeur : les attributs de journal décrivent l'événement lui-même (par exemple, `log.file.path: /var/log/zabbix/zabbix_frontend.log`), tandis que les attributs de ressource décrivent le service et l'environnement dans lequel il s'exécute (par exemple, `host.name: eabf8c2a5ddc`).

Pour plus de détails sur les journaux, consultez la documentation OpenTelemetry.

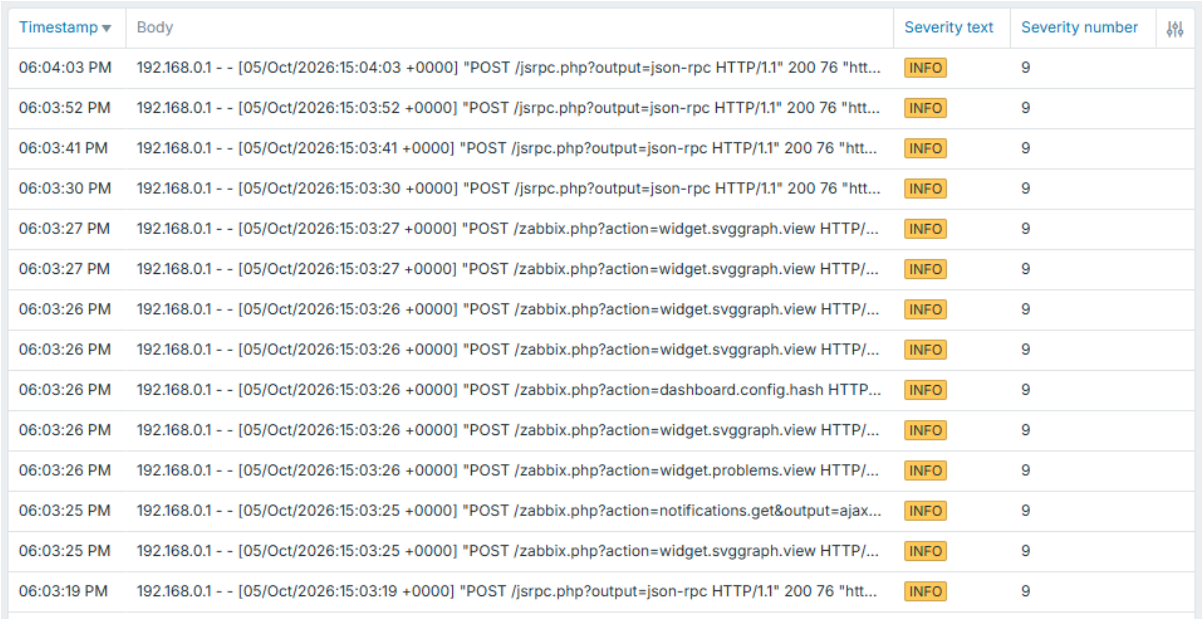


La section *APM > Journaux* contient les éléments suivants :

- Bouton Mode kiosque - affiche uniquement le contenu de la page, sans le menu ni l'en-tête.
- Sélecteur de période - sélectionne la période pour laquelle afficher les enregistrements de journaux.
- Filtre - affiche uniquement les enregistrements de journaux qui vous intéressent.
- Liste des journaux - permet de consulter les enregistrements de journaux.
- Détails du journal - permet de consulter les détails de l'enregistrement de journal sélectionné.

Liste des journaux

Chaque ligne de la liste des journaux représente un enregistrement de journal.



Timestamp ▾	Body	Severity text	Severity number
06:04:03 PM	192.168.0.1 - - [05/Oct/2026:15:04:03 +0000] "POST /jsrpc.php?output=json-rpc HTTP/1.1" 200 76 "htt...	INFO	9
06:03:52 PM	192.168.0.1 - - [05/Oct/2026:15:03:52 +0000] "POST /jsrpc.php?output=json-rpc HTTP/1.1" 200 76 "htt...	INFO	9
06:03:41 PM	192.168.0.1 - - [05/Oct/2026:15:03:41 +0000] "POST /jsrpc.php?output=json-rpc HTTP/1.1" 200 76 "htt...	INFO	9
06:03:30 PM	192.168.0.1 - - [05/Oct/2026:15:03:30 +0000] "POST /jsrpc.php?output=json-rpc HTTP/1.1" 200 76 "htt...	INFO	9
06:03:27 PM	192.168.0.1 - - [05/Oct/2026:15:03:27 +0000] "POST /zabbix.php?action=widget.svggraph.view HTTP/...	INFO	9
06:03:27 PM	192.168.0.1 - - [05/Oct/2026:15:03:27 +0000] "POST /zabbix.php?action=widget.svggraph.view HTTP/...	INFO	9
06:03:26 PM	192.168.0.1 - - [05/Oct/2026:15:03:26 +0000] "POST /zabbix.php?action=widget.svggraph.view HTTP/...	INFO	9
06:03:26 PM	192.168.0.1 - - [05/Oct/2026:15:03:26 +0000] "POST /zabbix.php?action=widget.svggraph.view HTTP/...	INFO	9
06:03:26 PM	192.168.0.1 - - [05/Oct/2026:15:03:26 +0000] "POST /zabbix.php?action=dashboard.config.hash HTTP...	INFO	9
06:03:26 PM	192.168.0.1 - - [05/Oct/2026:15:03:26 +0000] "POST /zabbix.php?action=widget.svggraph.view HTTP/...	INFO	9
06:03:26 PM	192.168.0.1 - - [05/Oct/2026:15:03:26 +0000] "POST /zabbix.php?action=widget.problems.view HTTP/...	INFO	9
06:03:25 PM	192.168.0.1 - - [05/Oct/2026:15:03:25 +0000] "POST /zabbix.php?action=notifications.get&output=ajax...	INFO	9
06:03:25 PM	192.168.0.1 - - [05/Oct/2026:15:03:25 +0000] "POST /zabbix.php?action=widget.svggraph.view HTTP/...	INFO	9
06:03:19 PM	192.168.0.1 - - [05/Oct/2026:15:03:19 +0000] "POST /jsrpc.php?output=json-rpc HTTP/1.1" 200 76 "htt...	INFO	9

Les enregistrements de journal sont affichés dans un tableau personnalisable. Le bouton *Personnaliser le tableau* dans l'en-tête du tableau permet d'afficher, de masquer ou de réorganiser les colonnes, ainsi que de rétablir la disposition par défaut. Vous pouvez également redimensionner chaque colonne en faisant glisser sa bordure droite ; double-cliquez sur la bordure pour rétablir sa taille. Toutes les personnalisations sont appliquées immédiatement et mémorisées pour l'utilisateur actuel.

La liste des journaux peut contenir les colonnes suivantes :

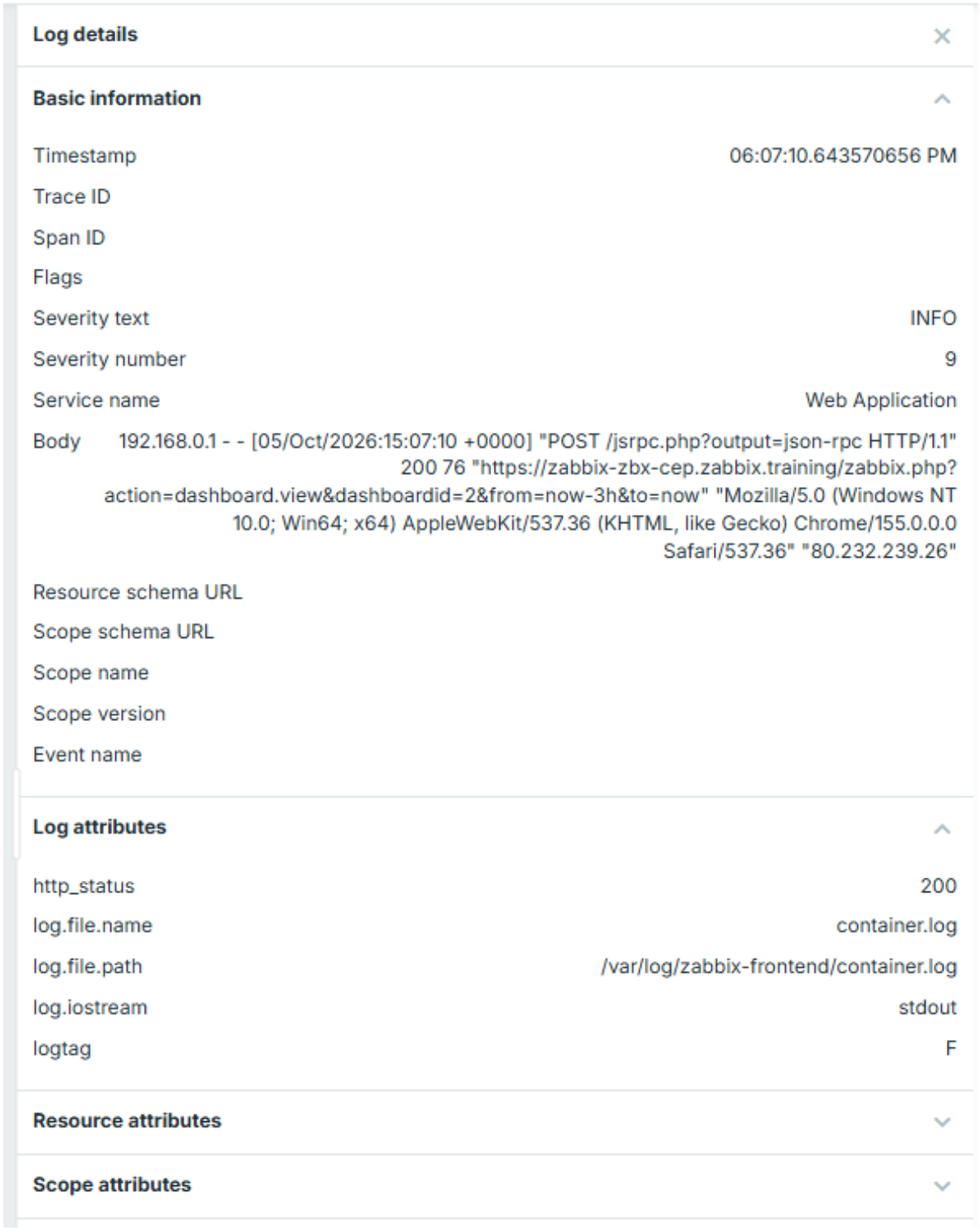
Note:
Chaque colonne correspond à une colonne de la table `otel_logs` de la source de données de télémétrie ; pour obtenir la liste des colonnes, consultez Colonnes de données de télémétrie prises en charge. La plupart des colonnes portent le même nom que dans la source de données (par exemple, *Texte de la gravité* correspond à la colonne `SeverityText`). Lorsque les noms diffèrent, la colonne de la source de données est indiquée dans le tableau ci-dessous.

Column	Description
Horodatage	Date et heure auxquelles l'événement s'est produit.
Corps	Message de l'enregistrement de journal.
Texte de la gravité	Niveau de gravité fourni par l'application (par exemple, `INFO`, `WARN` ou `ERROR`). La couleur de l'étiquette dépend de la gravité.
Nombre de gravité	Niveau de gravité sous forme de nombre : **0** - NON SPÉCIFIÉ. **1-4** - TRACE. **5-8** - DEBUG. **9-12** - INFO. **13-16** - WARN. **17-20** - ERROR. **21-24** - FATAL.
ID de trace	ID, le cas échéant, de la trace pendant laquelle l'enregistrement de journal a été produit.
ID de span	ID, le cas échéant, du span pendant lequel l'enregistrement de journal a été produit.
Indicateurs	Indicateurs de trace de l'enregistrement de journal : **0** - la trace n'est pas échantillonnée. **1** - la trace est échantillonnée (enregistrée). Colonne de la source de données : `TraceFlags`.
Nom du service	Service ayant produit l'enregistrement de journal.
Nom de la portée	Portée d'instrumentation ayant produit l'enregistrement de journal.
Version de la portée	Version de la portée d'instrumentation.
Nom de l'événement	Nom, le cas échéant, du type d'événement représenté par l'enregistrement de journal.
Attributs du journal	Attributs de journal de l'enregistrement de journal. Cliquez sur le bouton à trois points ••• pour afficher tous les attributs.
Attributs de la ressource	Attributs de ressource de l'enregistrement de journal. Cliquez sur le bouton à trois points ••• pour afficher tous les attributs.

Column	Description
Attributs de la portée	Attributs de la portée d'instrumentation. Cliquez sur le bouton à trois points ••• pour afficher tous les attributs.

Détails du journal

Cliquez sur un enregistrement de journal dans la liste des journaux pour ouvrir le panneau *Détails du journal*, qui affiche toutes les informations relatives à l'enregistrement.



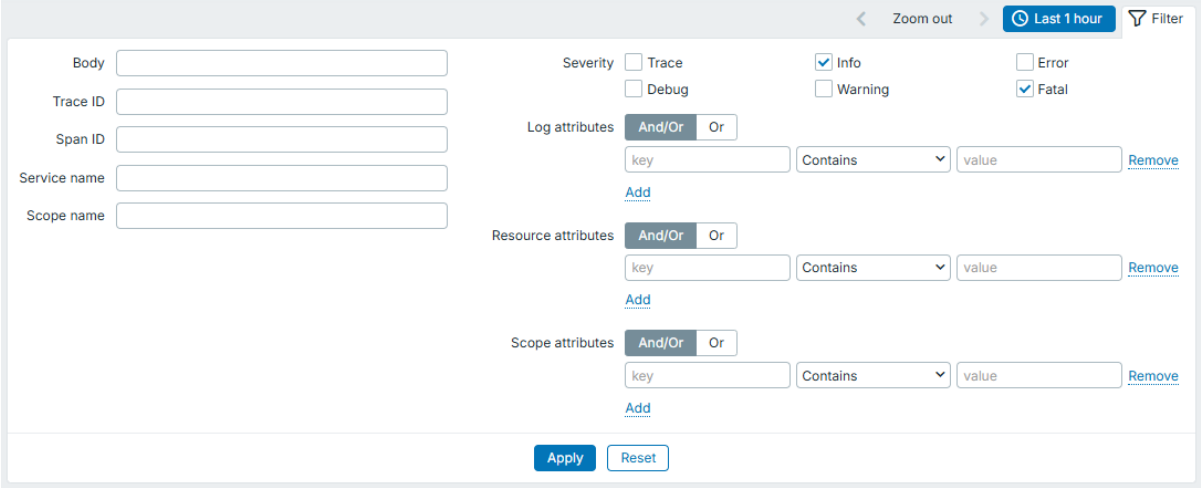
Le panneau *Détails du journal* peut comporter les sections suivantes :

Section	Description
Informations de base	ID de trace, ID de span et indicateurs de trace de l'enregistrement.
Attributs du journal	Attributs de journal de l'enregistrement.

Section	Description
Attributs de ressource	URL du schéma et attributs de ressource.
Attributs de portée	URL du schéma, nom, version et attributs de la portée d'instrumentation.

Utilisation du filtre

Vous pouvez utiliser le filtre pour afficher uniquement les enregistrements de journal qui vous intéressent.



Pour filtrer par attributs, saisissez la clé de l'attribut, sélectionnez l'opérateur et saisissez la valeur de l'attribut.

Les opérateurs suivants sont pris en charge :

- **Contient** - *(par défaut)* la valeur de l'attribut contient la chaîne saisie.
- **Est égal à** - la valeur de l'attribut est égale à la chaîne saisie.
- **Ne contient pas** - la valeur de l'attribut ne contient pas la chaîne saisie.
- **N'est pas égal à** - la valeur de l'attribut n'est pas égale à la chaîne saisie.
- **Existe** - l'attribut avec la clé saisie existe.
- **N'existe pas** - l'attribut avec la clé saisie n'existe pas.

Si vous ajoutez plusieurs conditions, sélectionnez la manière de les évaluer :

- **Et/Ou** - *(par défaut)* toutes les conditions doivent être remplies ; les conditions ayant la même clé d'attribut sont combinées avec *Ou*.
- **Ou** - au moins une condition doit être remplie.

Services

Aperçu

Le menu Services concerne les fonctions surveillance du service de Zabbix.

Services

Vue d'ensemble

Dans cette section, vous pouvez consulter l'état général de l'ensemble des services configurés dans Zabbix, en fonction de votre infrastructure.

Un service peut être une hiérarchie composée de plusieurs niveaux d'autres services, appelés services « enfants », qui contribuent à l'état global du service (voir également une présentation de la fonctionnalité de supervision des services).

Les principales catégories d'état d'un service sont *OK* ou *Problème*. L'état *Problème* est indiqué par le nom et la couleur correspondant à la gravité du problème.

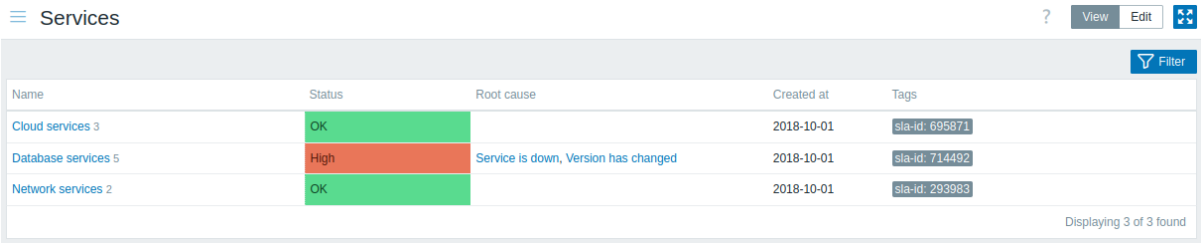
Le mode d'affichage permet de superviser les services ainsi que leur état et d'autres détails. Vous pouvez également configurer la hiérarchie des services dans cette section (ajouter ou modifier des services et des services enfants) en passant en mode édition.

Pour passer du mode d'affichage au mode édition (et inversement), cliquez sur le bouton correspondant dans le coin supérieur droit :

- View - afficher les services
- Edit - ajouter ou modifier des services et des services enfants

Notez que l'accès à la modification dépend des paramètres du rôle utilisateur.

Affichage des services



Une liste des services existants est affichée.

Données affichées :

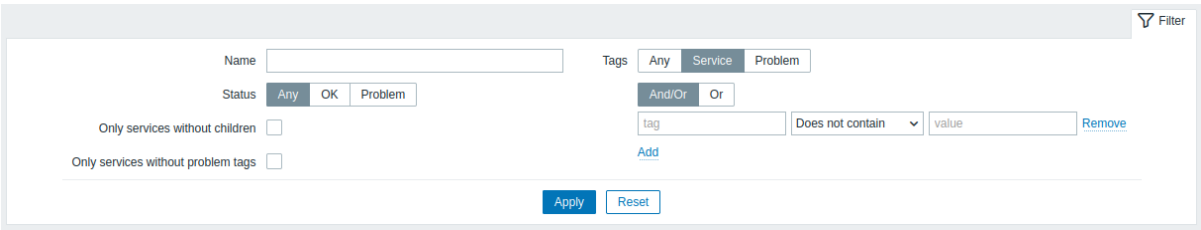
Parameter	Description
Name	Nom du service. Le nom du service est un lien vers les détails du service. Le nombre après le nom indique combien de services enfants le service possède.
Status	État du service : **OK** - aucun problème **<problem color and severity>** - indique un problème et sa gravité. En cas de problèmes multiples, la couleur et la gravité du problème le plus critique sont affichées.
Root cause	Les problèmes sous-jacents qui affectent directement ou indirectement l'état du service sont □□□□és. Les mêmes problèmes sont listés comme renvoyés par la macro {SERVICE.ROOTCAUSE}. Cliquez sur le nom du problème pour voir plus de détails à son sujet dans *Monitoring* > *Problems*. Les problèmes qui n'affectent pas l'état du service ne figurent pas dans la liste.
Created at	L'heure à laquelle le service a été créé est affichée.
Tags	Les tags du service sont affichés. Les tags sont utilisés pour identifier un service dans les actions de service et les SLA.

Boutons

Les boutons du mode d'affichage, communs à toutes les sections, sont décrits sur la page Monitoring.

Utilisation du filtre

Vous pouvez utiliser le filtre pour afficher uniquement les services qui vous intéressent.



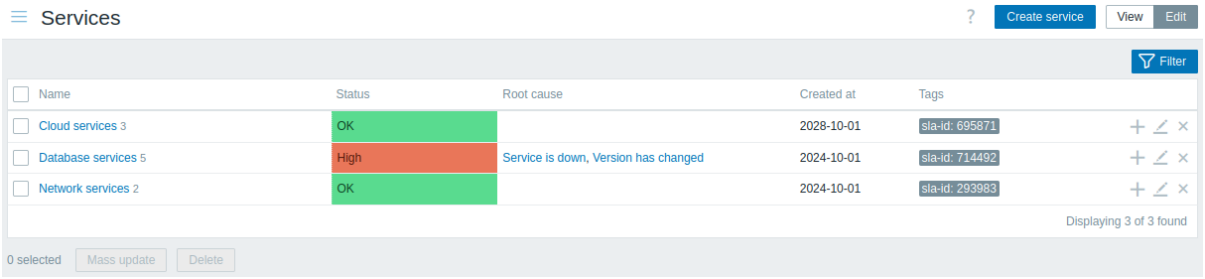
Paramètre	Description
Nom	Filtrer par nom de service.
Statut	Filtrer par état du service.
Uniquement les services sans enfants	Cochez la case pour afficher uniquement les services sans services enfants. Ce paramètre est disponible uniquement en mode d'édition du service.
Uniquement les services sans tags de problème	Cochez la case pour afficher uniquement les services sans tags de problème. Ce paramètre est disponible uniquement en mode d'édition du service.

Paramètre	Description
Tags	Filtrer par nom et valeur du tag de service, ou par nom et valeur du tag de problème du service (en mode d'édition du service). Il est possible d'inclure comme d'exclure des tags et des valeurs de tag spécifiques. Plusieurs conditions peuvent être définies. La correspondance des noms de tag est toujours sensible à la casse. Plusieurs opérateurs sont disponibles pour chaque condition : **Existe** - inclure les noms de tag spécifiés ; **Égal à** - inclure les noms et valeurs de tag spécifiés (sensible à la casse) ; **Contient** - inclure les noms de tag spécifiés lorsque les valeurs de tag contiennent la chaîne saisie (correspondance de sous-chaîne, insensible à la casse) ; **N'existe pas** - exclure les noms de tag spécifiés ; **N'est pas égal à** - exclure les noms et valeurs de tag spécifiés (sensible à la casse) ; **Ne contient pas** - exclure les noms de tag spécifiés lorsque les valeurs de tag contiennent la chaîne saisie (correspondance de sous-chaîne, insensible à la casse). Deux types de calcul sont disponibles pour les conditions : **Et/Ou** - toutes les conditions doivent être remplies ; les conditions ayant le même nom de tag seront regroupées par la condition Ou ; **Ou** - il suffit qu'une seule condition soit remplie.

Modification des services

Cliquez sur le bouton *Edit* pour accéder au mode de modification. En mode de modification, la liste est complétée par des cases à cocher devant les entrées, ainsi que par les options supplémentaires suivantes :

-  - ajouter un service enfant à ce service
-  - modifier ce service
-  - supprimer ce service

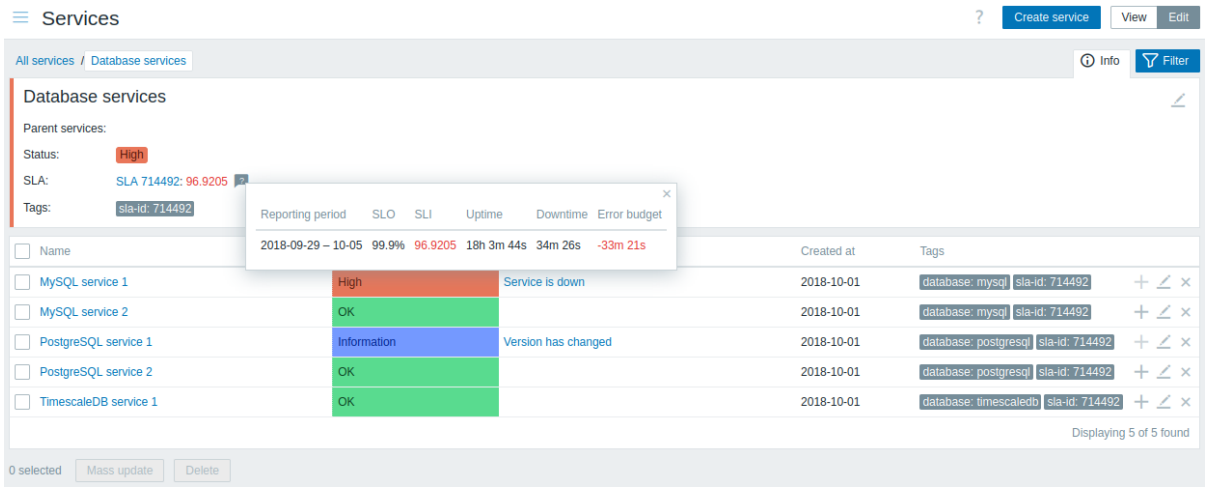


Pour configurer un nouveau service, cliquez sur le bouton *Create service* dans le coin supérieur droit.

Détails du service

Pour accéder aux détails du service, cliquez sur le nom du service. Pour revenir à la liste de tous les services, cliquez sur *Tous les services*.

Les détails du service incluent la zone d'informations et la liste des services enfants.



Pour accéder à la zone d'informations, cliquez sur l'onglet *Info*. La zone d'informations contient les entrées suivantes :

- Noms des services parents (le cas échéant)
- État actuel de ce service
- SLA actuel(s) de ce service, au format `nom du SLA:indicateur de niveau de service`. Le "nom du SLA" est également un lien vers le rapport SLA de ce service. Si vous placez la souris sur la zone d'informations à côté de l'indicateur de niveau de service (SLI), une liste d'informations contextuelle s'affiche avec les détails du SLI. L'indicateur de niveau de service affiche le niveau de service actuel, en pourcentage.
- Balises de service

La zone d'informations contient également un lien vers la configuration du service.

Pour utiliser le filtre des services enfants, cliquez sur l'onglet *Filtre*.

En mode d'édition, la liste des services enfants est complétée par des options d'édition supplémentaires :

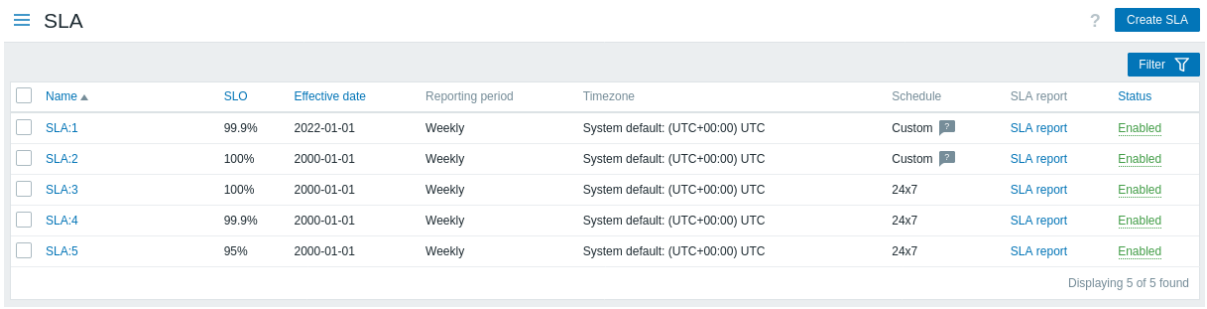
-  - ajouter un service enfant à ce service
-  - modifier ce service
-  - supprimer ce service

SLA

Vue d'ensemble

Cette section permet d'afficher et de configurer les SLA.

SLAs



Une liste des SLAs configurés est affichée. *Note* que seuls les SLAs liés aux services accessibles à l'utilisateur seront affichés (en lecture seule, sauf si *Manage SLA* est activé pour le rôle utilisateur).

Données affichées :

Parameter	Description
Name	Le nom du SLA est affiché. Le nom est un lien vers la configuration du SLA.
SLO	L'objectif de niveau de service (SLO) est affiché.
Effective date	La date de début du calcul du SLA est affichée.

Parameter	Description
Reporting period	La période utilisée dans le rapport SLA est affichée - *daily*, *weekly*, *monthly*, *quarterly*, ou *annually*.
Time zone	Le fuseau horaire du SLA est affiché.
Schedule	Le planning du SLA est affiché - 24x7 ou personnalisé.
SLA report	Cliquez sur le lien pour voir le rapport SLA de ce SLA.
Status	Le statut du SLA est affiché - activé ou désactivé.

Rapport SLA

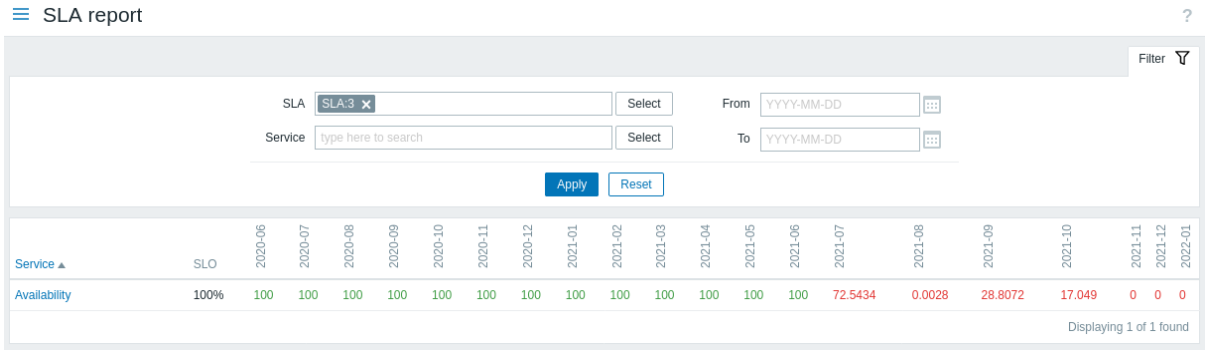
Vue d'ensemble

Cette section permet d'afficher les rapports SLA en fonction des critères sélectionnés dans le filtre.

Les rapports SLA peuvent également être affichés sous forme de widget de tableau de bord.

Rapport

Le filtre vous permet de sélectionner le rapport en fonction du nom du SLA ainsi que du nom du service. Il est également possible de limiter la période affichée.

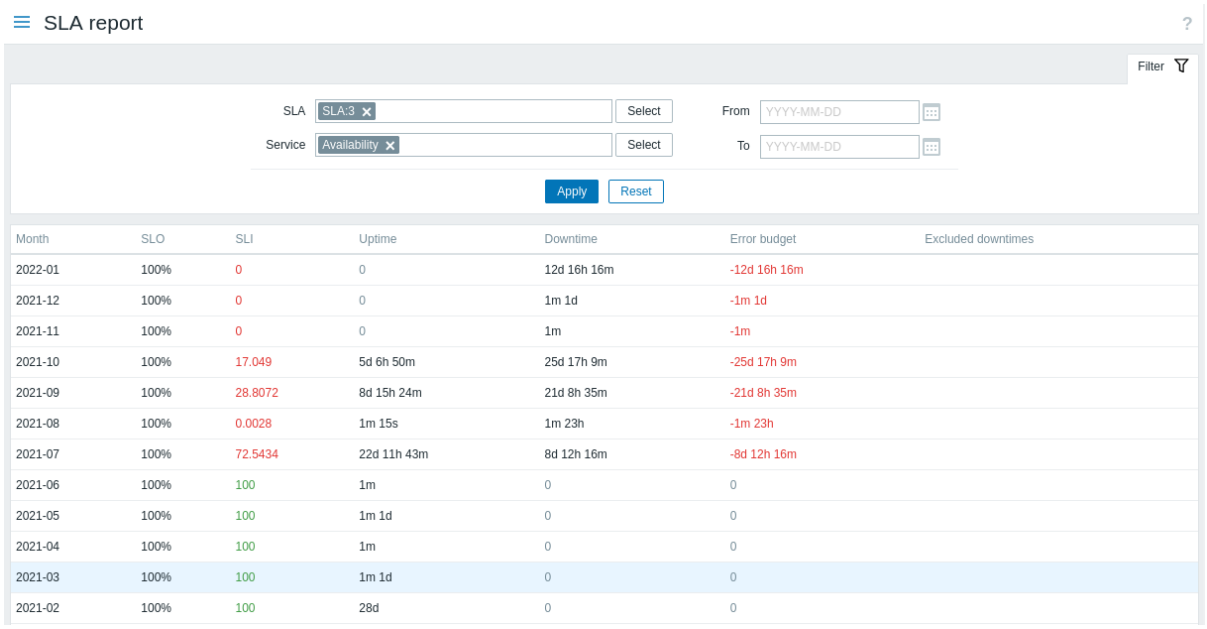


Chaque colonne (période) affiche le SLI correspondant à cette période. Les SLI qui ne respectent pas le SLO défini sont mis en évidence en rouge.

20 périodes sont affichées dans le rapport. Un maximum de 100 périodes peut être affiché si les dates *Du* et *Au* sont toutes deux spécifiées.

Détails du rapport

Si vous cliquez sur le nom du service dans le rapport, vous pouvez accéder à un autre rapport qui affiche une vue plus détaillée.



Notez que la durée négative du problème n'affecte pas le calcul ni le reporting du SLA.

Inventaire

Aperçu

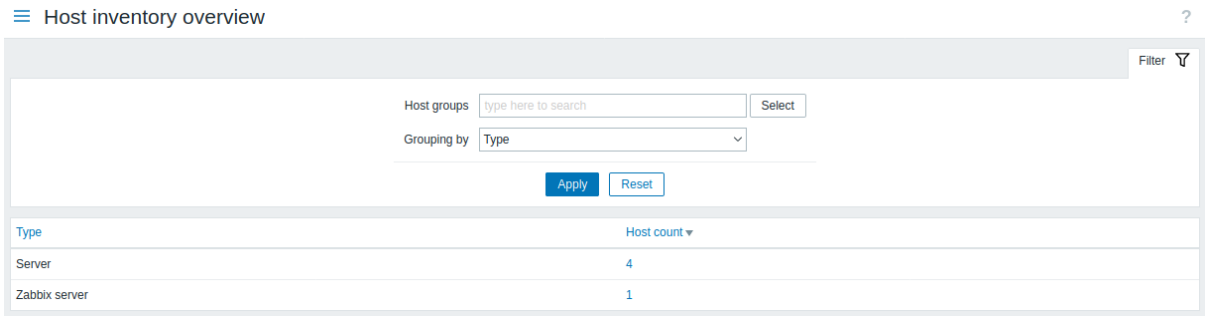
Le menu Inventaire comprend des sections offrant un aperçu des données d'inventaire de l'hôte selon un paramètre choisi, ainsi que la possibilité d'afficher les détails de l'inventaire de l'hôte.

Vue d'ensemble

Vue d'ensemble

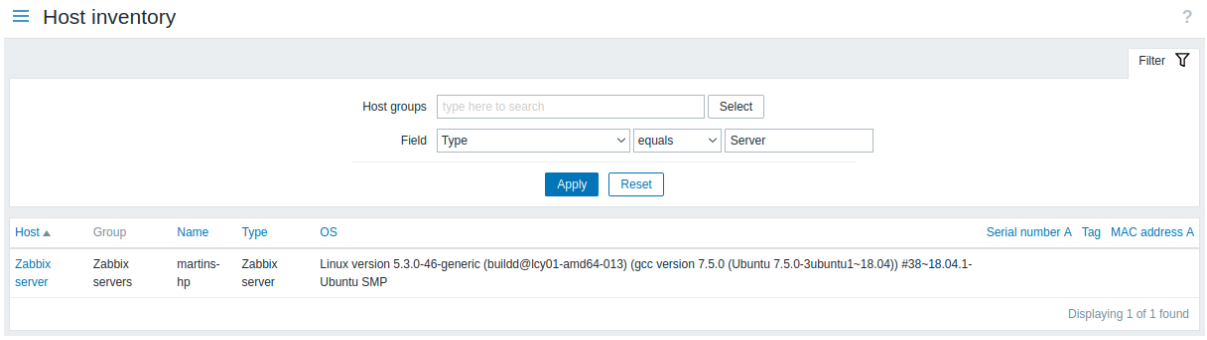
La section *Inventory* > *Overview* fournit des moyens d'obtenir une vue d'ensemble des données de l'inventaire des hôtes.

Pour afficher une vue d'ensemble, choisissez des groupes d'hôtes (ou aucun) ainsi que le champ d'inventaire selon lequel afficher les données. Le nombre d'hôtes correspondant à chaque entrée du champ choisi sera affiché.



L'exhaustivité d'une vue d'ensemble dépend de la quantité d'informations d'inventaire renseignées pour les hôtes.

Les nombres dans la colonne *Host count* sont des liens ; ils permettent d'afficher ces hôtes filtrés dans le tableau *Host Inventories*.

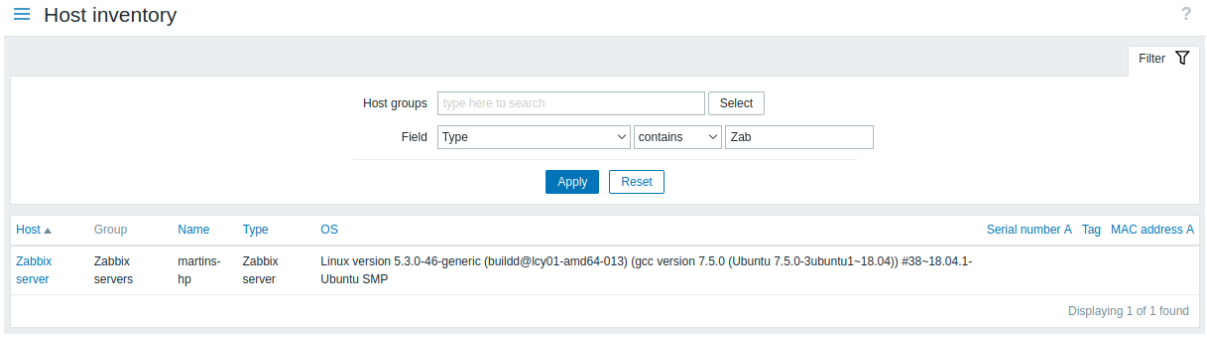


Hôtes

Vue d'ensemble

Dans la section *Inventory* > *Hosts*, les données d'inventaire des hôtes sont affichées.

Vous pouvez filtrer les hôtes par groupe(s) d'hôtes et par n'importe quel champ d'inventaire afin d'afficher uniquement les hôtes qui vous intéressent.

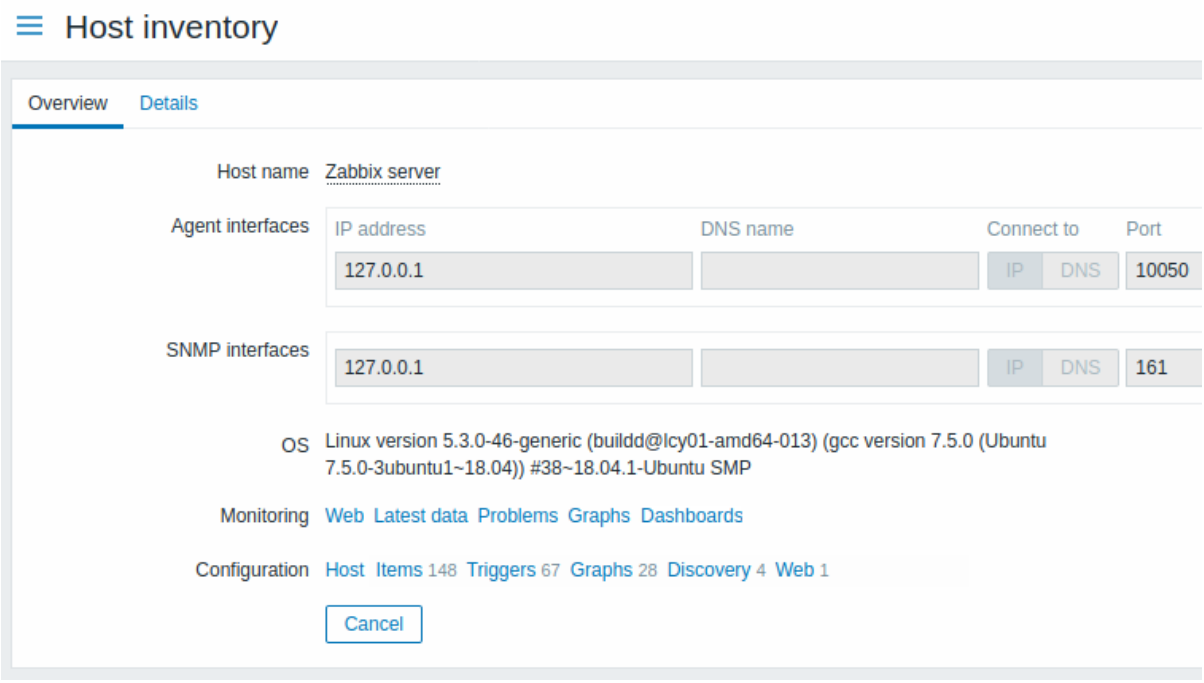


Pour afficher tous les inventaires des hôtes, cliquez sur le bouton *Reset*.

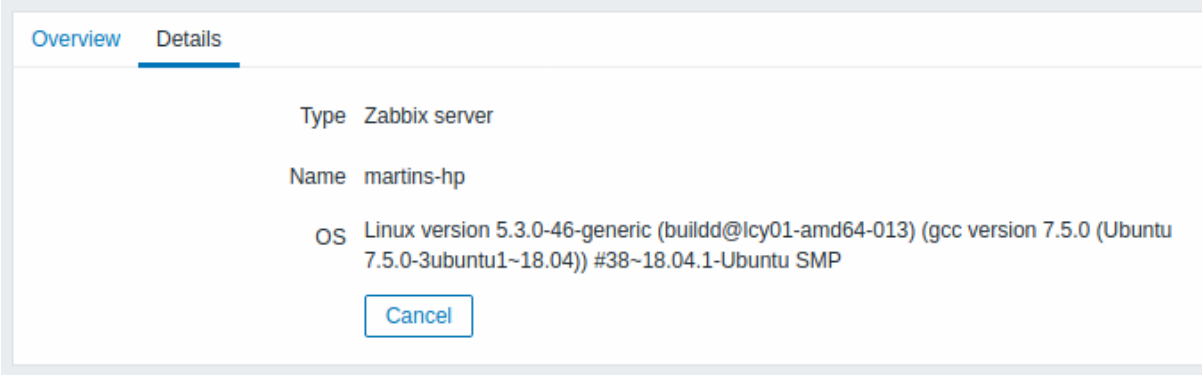
Bien que seuls certains champs clés de l'inventaire soient affichés dans le tableau, vous pouvez également consulter toutes les informations d'inventaire disponibles pour cet hôte. Pour ce faire, cliquez sur le nom de l'hôte dans la première colonne.

Détails de l'inventaire

L'onglet **Vue d'ensemble** contient des informations générales sur l'hôte, les dernières données de surveillance et les options de configuration de l'hôte :



L'onglet **Détails** contient tous les détails d'inventaire disponibles pour l'hôte :



L'exhaustivité des données d'inventaire dépend de la quantité d'informations d'inventaire conservées pour l'hôte. Si aucune information n'est conservée, l'onglet *Détails* est désactivé.

Rapports

Aperçu

Le menu Rapports comprend plusieurs sections contenant une variété de rapports prédéfinis et personnalisables par l'utilisateur, qui affichent une vue d'ensemble de paramètres tels que les informations système, les déclencheurs et les données collectées.

Informations système

Vue d'ensemble

Dans *Rapports* > *Informations système*, un résumé des principales données du serveur Zabbix et du système est affiché. Les données système sont collectées à l'aide d'éléments internes.

Le rapport d'informations système est visible uniquement par les utilisateurs Zabbix *Super administrateur* et Zabbix *Administrateur*.

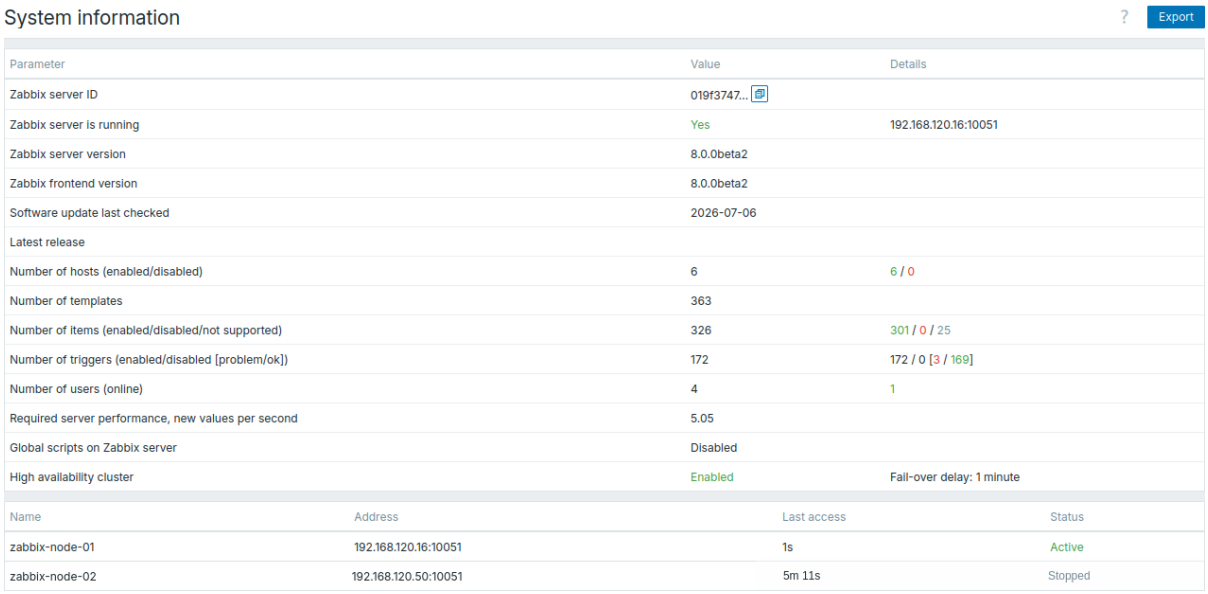
Notez que dans une configuration haute disponibilité, il est possible de rediriger la source des informations système (instance du serveur). Pour ce faire, modifiez le fichier de configuration de l'interface : décommentez et définissez `$ZBX_SERVER`, ou à

la fois $ZBX_SERVER et $ZBX_SERVER_PORT, sur un serveur autre que celui indiqué comme actif. Notez que si vous définissez uniquement $ZBX_SERVER, la valeur par défaut (10051) sera utilisée pour $ZBX_SERVER_PORT.

Lorsque la configuration haute disponibilité est activée, un bloc distinct contenant les détails des nœuds haute disponibilité est affiché sous les statistiques système.

Les *Informations système* sont également disponibles sous forme de widget de tableau de bord.

Statistiques du système



Données affichées :

Paramètre	Valeur	Détails
ID du serveur Zabbix	Identifiant unique du serveur Zabbix (UUIDv7).	L'ID du serveur est généré au premier démarrage du serveur. Les 8 premiers caractères sont affichés, suivis de points de suspension. Lorsque vous survolez ou cliquez sur une valeur d'ID de serveur, une infobulle affiche la valeur complète. Cliquer sur l'icône à côté de la valeur copie l'ID dans le presse-papiers. Si la haute disponibilité est activée, chaque nœud possède son propre ID de nœud pour l'adressage en ligne de commande, mais tous les nœuds sont regroupés sous un seul ID de serveur généré par le nœud lancé en premier (ou qui était auparavant un serveur autonome), quel que soit le nœud actuellement actif.
Le serveur Zabbix est en cours d'exécution	État du serveur Zabbix : **Oui** - le serveur est en cours d'exécution **Non** - le serveur n'est pas en cours d'exécution *Remarque :* Pour afficher le reste des informations, l'interface web a besoin que le serveur soit en cours d'exécution et qu'au moins un processus trapper soit démarré sur le serveur (paramètre StartTrappers dans le fichier `zabbix_server.conf` > 0).	Emplacement et port du serveur Zabbix.

Paramètre	Valeur	Détails
Version du serveur Zabbix	Le numéro de version actuelle du serveur est affiché. *Remarque :* Il est affiché uniquement lorsque le serveur Zabbix est en cours d'exécution.	L'état de la version du serveur est affiché : **À jour** - la dernière version est utilisée ; **Nouvelle mise à jour disponible** - une version plus récente est disponible ; **Obsolète** - la période complète de support de cette version a expiré. Ces informations sont disponibles uniquement si la vérification des mises à jour logicielles est activée dans la configuration du serveur Zabbix. Rien n'est affiché si la dernière vérification des mises à jour logicielles a été effectuée il y a plus d'une semaine ou si aucune donnée sur la version actuelle n'est disponible.
Version de l'interface Zabbix	Le numéro de version de l'interface Zabbix est affiché.	L'état de la version de l'interface Zabbix est affiché : **À jour** - la dernière version est utilisée ; **Nouvelle mise à jour disponible** - une version plus récente est disponible ; **Obsolète** - la période complète de support de cette version a expiré. Ces informations sont disponibles uniquement si la vérification des mises à jour logicielles est activée dans la configuration du serveur Zabbix. Rien n'est affiché si la dernière vérification des mises à jour logicielles a été effectuée il y a plus d'une semaine ou si aucune donnée sur la version actuelle n'est disponible.
Dernière vérification des mises à jour logicielles	La date de la dernière vérification des mises à jour logicielles de Zabbix est affichée. Ces informations sont disponibles uniquement si la vérification des mises à jour logicielles est activée dans la configuration du serveur.	
Dernière version	Le numéro d'une version plus récente (si disponible) pour la version actuelle de Zabbix est affiché. Ces informations sont disponibles uniquement si la vérification des mises à jour logicielles est activée dans la configuration du serveur. Rien n'est affiché si la dernière vérification des mises à jour logicielles a été effectuée il y a plus d'une semaine ou si aucune donnée sur la version actuelle n'est disponible.	Un lien vers les notes de version de la dernière version de Zabbix disponible est affiché.
Nombre d'hôtes	Le nombre total d'hôtes configurés est affiché.	Nombre d'hôtes surveillés/non surveillés.
Nombre de modèles	Le nombre total de modèles est affiché.	
Nombre d'éléments	Le nombre total d'éléments est affiché.	Nombre d'éléments de niveau hôte surveillés/désactivés/non pris en charge. Les éléments des hôtes désactivés sont comptabilisés comme désactivés.
Nombre de déclencheurs	Le nombre total de déclencheurs est affiché.	Nombre de déclencheurs de niveau hôte activés/désactivés ; répartition des déclencheurs activés selon les états « Problème »/« OK ». Les déclencheurs répertoriés sous l'état « OK » incluent les déclencheurs dont l'état est « Inconnu ». Les déclencheurs dépendant d'éléments désactivés ou affectés à des hôtes désactivés sont comptabilisés comme désactivés.
Nombre d'utilisateurs	Le nombre total d'utilisateurs configurés est affiché.	Nombre d'utilisateurs en ligne.

Paramètre	Valeur	Détails
Performances requises du serveur, nouvelles valeurs par seconde	Le nombre attendu de nouvelles valeurs traitées par le serveur Zabbix par seconde est affiché.	Les *performances requises du serveur* constituent une estimation et peuvent servir de référence. Pour obtenir le nombre précis de valeurs traitées, utilisez l'élément interne `zabbix[wcache,values,all]`. Les éléments activés des hôtes surveillés sont inclus dans le calcul. Les éléments de journal sont comptabilisés comme une valeur par intervalle de mise à jour de l'élément. Les valeurs des intervalles réguliers sont comptabilisées ; les valeurs des intervalles flexibles et planifiés ne le sont pas. Le calcul n'est pas ajusté pendant une période de maintenance « nodata ». Les éléments trapper ne sont pas comptabilisés.
Scripts globaux sur le serveur Zabbix	**Désactivé** est affiché dans ce champ si les scripts globaux sont désactivés sur le serveur Zabbix en définissant EnableGlobalScripts=0 dans la configuration du serveur.	
Cluster à haute disponibilité	État du cluster à haute disponibilité du serveur Zabbix : **Désactivé** - serveur autonome **Activé** - au moins un nœud de haute disponibilité existe	Si cette option est activée, le délai de basculement est affiché.

Les informations système affichent également un message d'erreur dans les conditions suivantes :

- La base de données utilisée ne possède pas le jeu de caractères ou la collation requis (UTF-8).
- La version de la base de données est inférieure ou supérieure à la plage prise en charge (disponible uniquement pour les utilisateurs ayant le type de rôle *Super administrateur*).
- Le housekeeping de TimescaleDB est mal configuré (les tables d'historique ou de tendances contiennent des blocs compressés, mais les options *Remplacer la période d'historique de l'élément* ou *Remplacer la période de tendance de l'élément* sont désactivées).

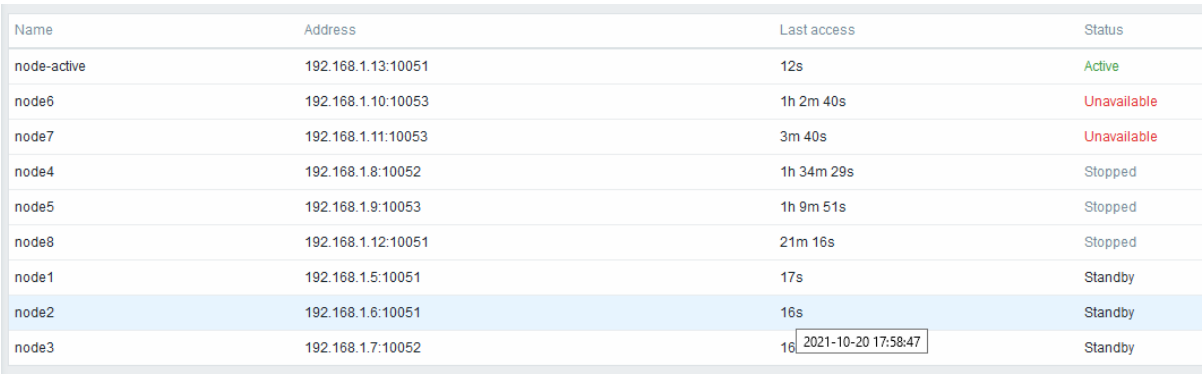
Boutons

Le bouton situé dans le coin supérieur droit de la page propose l'option suivante :

Export	Exporter les informations système vers un fichier JSON. Le fichier est enregistré sous le nom `Zabbix-system-information-{time}.json`, où `{time}` est l'horodatage Unix indiquant quand les données ont été préparées sur le serveur. Si la haute disponibilité est activée, les détails des nœuds de haute disponibilité sont également exportés.

Nœuds de haute disponibilité

Si le cluster de haute disponibilité est activé, un bloc de données supplémentaire s'affiche et montre l'état de chaque nœud de haute disponibilité.



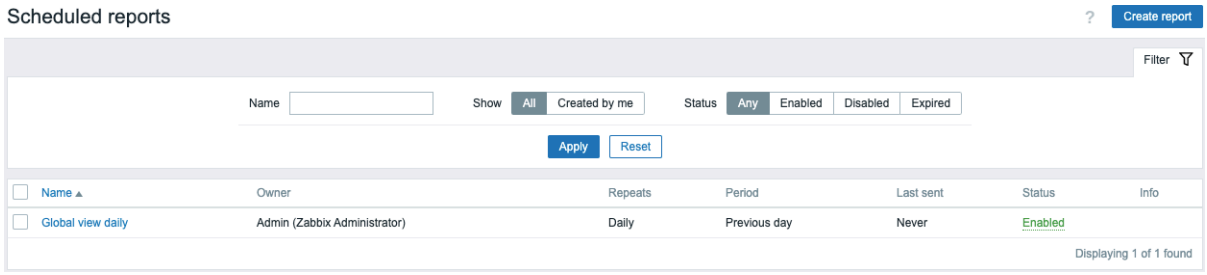
Données affichées :

Column	Description
Name	Nom du nœud, tel que défini dans la configuration du serveur.
Address	Adresse IP et port du nœud.
Last access	Heure du dernier accès au nœud. Le survol de la cellule affiche l'horodatage du dernier accès au format long.
Status	État du nœud. Les lignes du tableau sont triées selon ces états dans l'ordre de priorité suivant : **Active** - le nœud est en fonctionnement et opérationnel **Unavailable** - le nœud n'a pas été vu depuis plus que le délai de basculement (vous pouvez vouloir en déterminer la cause) **Stopped** - le nœud a été arrêté ou n'a pas pu démarrer (vous pouvez vouloir le démarrer ou le supprimer) **Standby** - le nœud est en fonctionnement et en attente

Rapports planifiés

Vue d'ensemble

Dans *Rapports* > *Rapports planifiés*, les utilisateurs disposant des autorisations suffisantes peuvent configurer la génération planifiée de versions PDF des tableaux de bord, qui seront envoyées par e-mail aux destinataires spécifiés.



L'écran d'accueil affiche des informations sur les rapports planifiés, qui peuvent être filtrées pour faciliter la navigation - voir la section Utilisation du filtre ci-dessous.

Données affichées :

Colonne	Description
Nom	Nom du rapport. Cliquer dessus ouvre le formulaire de configuration du rapport.
Propriétaire	Utilisateur ayant créé le rapport.
Répétition	Fréquence de génération du rapport (quotidienne/hebdomadaire/mensuelle/annuelle).
Période	Période pour laquelle le rapport est préparé (jour, semaine, mois ou année précédente).
Dernier envoi	Date et heure auxquelles le dernier rapport a été envoyé.
État	État actuel du rapport (activé/désactivé/expiré). Les utilisateurs disposant des autorisations suffisantes peuvent modifier l'état en cliquant dessus - de « Activé » à « Désactivé » (et inversement) ; de « Expiré » à « Désactivé » (et inversement). Pour les utilisateurs ne disposant pas des droits suffisants, l'état n'est pas cliquable.
Informations	Affiche des icônes informatives : Une icône rouge indique que la génération du rapport a échoué ; en la survolant, une infobulle affiche les informations relatives à l'erreur. Une icône jaune indique qu'un rapport a été généré, mais que l'envoi à certains destinataires (ou à tous) a échoué, ou que le rapport a expiré ; en la survolant, une infobulle affiche des informations supplémentaires.

Utilisation du filtre

Vous pouvez utiliser le filtre pour affiner la liste des rapports. Pour de meilleures performances de recherche, les données sont recherchées avec les macros non résolues.

Les options de filtrage suivantes sont disponibles :

- *Nom* - la correspondance partielle du nom est autorisée
- *Afficher* - créés par l'utilisateur actuel ou tous les rapports
- *Statut* - sélectionnez entre « Any » (afficher tous les rapports), « Enabled », « Disabled » ou « Expired »

Le filtre se trouve sous le nom de la section *Scheduled reports*. Il peut être ouvert et réduit en cliquant sur l'onglet *Filter* dans le coin supérieur droit.

Mise à jour en masse

Il peut arriver que vous souhaitiez supprimer ou modifier le statut de plusieurs rapports en une seule fois. Au lieu d'ouvrir chaque rapport individuellement pour le modifier, vous pouvez utiliser la fonction de mise à jour en masse à cet effet.

Pour effectuer une mise à jour en masse de certains rapports, procédez comme suit :

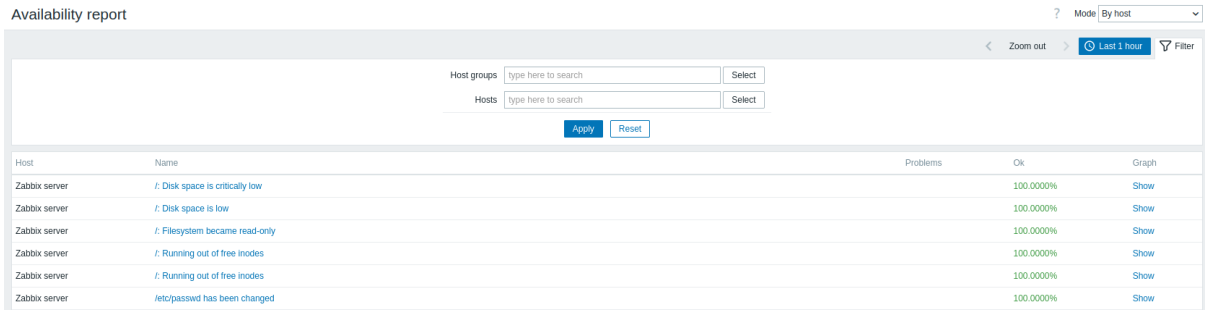
- Cochez les cases des rapports à mettre à jour dans la liste
- Cliquez sur le bouton requis sous la liste pour appliquer les modifications (*Activer*, *Désactiver* ou *Supprimer*)

Rapport de disponibilité

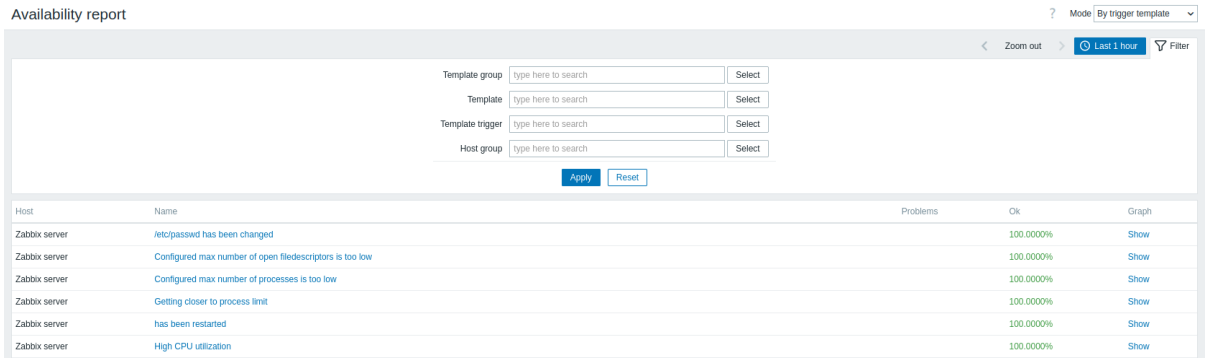
Aperçu

Dans *Reports* > *Availability report*, vous pouvez voir quelle proportion du temps chaque déclencheur a été dans l'état problème/OK.

Pour chaque état, un pourcentage du temps est affiché, ce qui permet de déterminer facilement la disponibilité de divers éléments de votre système.



Dans la liste déroulante située dans le coin supérieur droit, vous pouvez choisir le mode de sélection : afficher les déclencheurs par hôtes ou par déclencheurs appartenant à un modèle.



Le nom du déclencheur est un lien vers les derniers événements de ce déclencheur.

Utilisation du filtre

Le filtre peut aider à réduire le nombre d'hôtes et/ou de déclencheurs affichés. Pour de meilleures performances de recherche, les données sont recherchées avec des macros non résolues.

Le filtre se trouve sous le nom de la section *Rapport de disponibilité*. Il peut être ouvert et réduit en cliquant sur l'onglet *Filtre* à droite.

Filtrage par modèle de déclencheur

En mode *Par modèle de déclencheur*, les résultats peuvent être filtrés par un ou plusieurs des paramètres listés ci-dessous.

Paramètre	Description
Groupe de modèles	Filtre les hôtes selon les déclencheurs hérités de modèles appartenant au groupe de modèles sélectionné. La spécification d'un groupe de modèles parent sélectionne implicitement tous les groupes de modèles imbriqués.

Paramètre	Description
Modèle	Filtre les hôtes selon les déclencheurs hérités du modèle sélectionné, y compris les modèles imbriqués. Si un modèle imbriqué possède ses propres déclencheurs, ces déclencheurs ne seront pas affichés.
Déclencheur du modèle	Filtre les hôtes selon le déclencheur sélectionné. Les autres déclencheurs des hôtes filtrés ne seront pas affichés.
Groupe d'hôtes	Filtre les hôtes appartenant au groupe d'hôtes sélectionné.

Filtrage par hôte

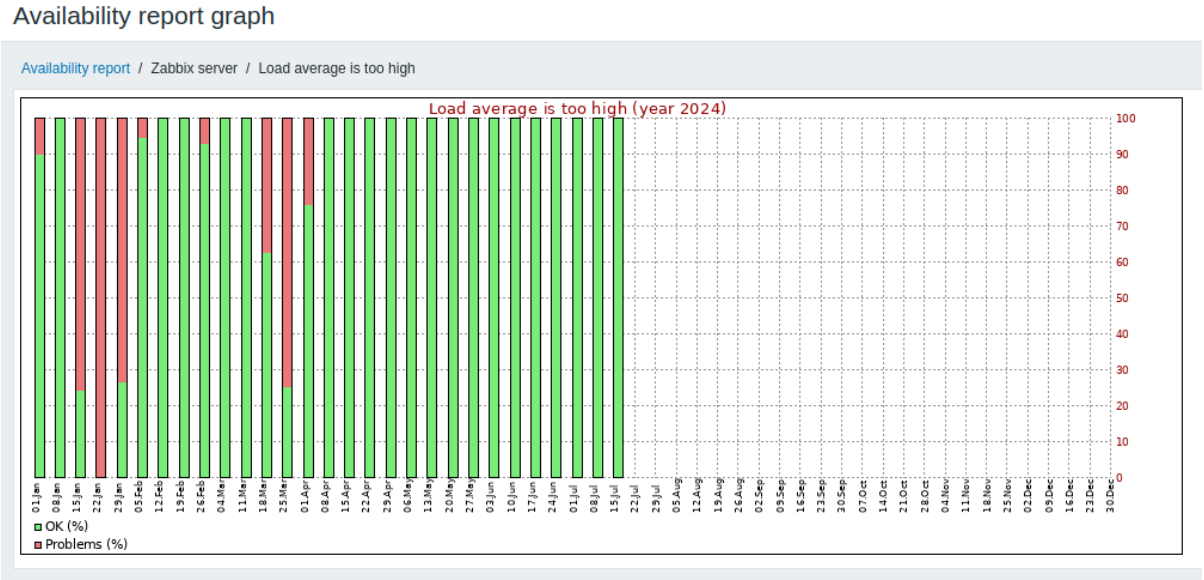
En mode *Par hôte*, les résultats peuvent être filtrés par hôte ou par groupe d'hôtes.
La spécification d'un groupe d'hôtes parent sélectionne implicitement tous les groupes d'hôtes imbriqués.

Sélecteur de période

Le sélecteur de période vous permet de sélectionner en un clic les périodes couramment utilisées. Le sélecteur peut être développé et réduit en cliquant sur l'onglet *Période* à côté du filtre.

En cliquant sur *Afficher* dans la colonne *Graphique*, les informations de disponibilité s'affichent sous forme de graphique à barres, où chaque barre représente une semaine écoulée de l'année en cours.

Availability report graph



La couleur verte d'une barre représente le temps OK et la couleur rouge, le temps en problème.

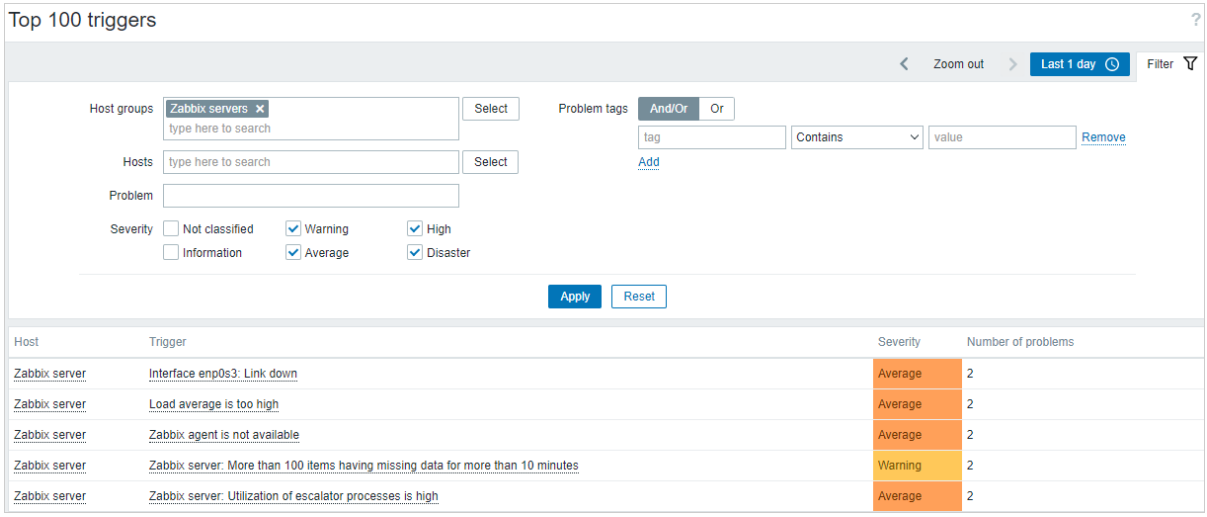
Effet des périodes de maintenance

La maintenance n'exclut pas automatiquement le temps du rapport de disponibilité. Seule la maintenance configurée comme *Aucune collecte de données* interrompt le polling (aucun problème n'est donc généré) et laisse par conséquent le pourcentage de disponibilité inchangé pour les déclencheurs concernés.

Top 100 déclencheurs

Vue d'ensemble

Dans *Rapports* > *Top 100 des déclencheurs*, vous pouvez voir les déclencheurs ayant généré le plus grand nombre de problèmes détectés au cours de la période sélectionnée.



Les entrées des colonnes hôte et déclencheur sont des liens qui proposent plusieurs options utiles :

- pour l'hôte - cliquer sur le nom de l'hôte ouvre le menu de l'hôte
- pour le déclencheur - cliquer sur le nom du déclencheur affiche des liens vers les événements les plus récents, le graphique simple de chaque élément du déclencheur, ainsi que les formulaires de configuration du déclencheur lui-même et de chaque élément du déclencheur

Utilisation du filtre

Vous pouvez utiliser le filtre pour afficher les déclencheurs par groupe d'hôtes, hôte, nom du problème, étiquettes ou gravité du déclencheur. La spécification d'un groupe d'hôtes parent sélectionne implicitement tous les groupes d'hôtes imbriqués. Pour améliorer les performances de recherche, les données sont recherchées avec les macros non résolues.

Le filtre se trouve sous le nom de la section *Top 100 des déclencheurs*. Il peut être ouvert et réduit en cliquant sur l'onglet *Filtre* à droite.

Sélecteur de période

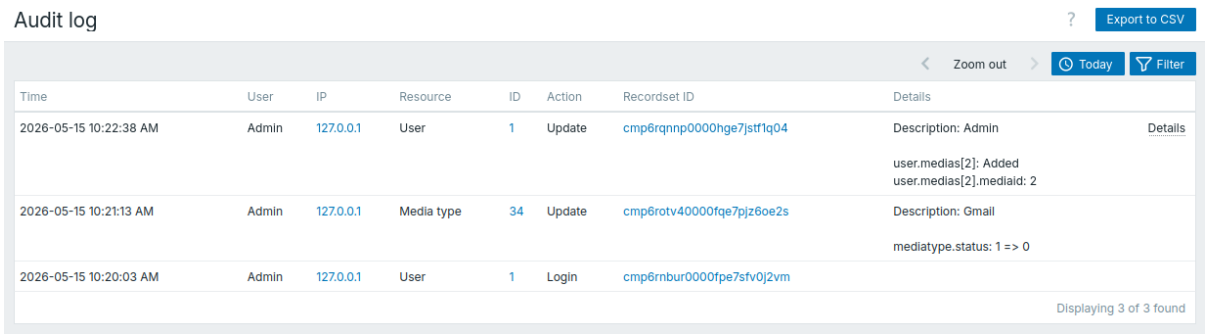
Le sélecteur de période situé à côté du filtre vous permet de sélectionner en un clic les périodes fréquemment utilisées. Pour plus d'informations, consultez Sélecteurs de période et d'hôte.

Journal d'audit

Aperçu

Dans la section *Reports* > *Audit log*, les enregistrements des activités des utilisateurs et du système peuvent être consultés.

> **Note:**
> Pour que les enregistrements d'audit soient collectés et affichés, la case à cocher *Enable audit logging* doit être activée dans la section *Administration* > *Audit log*. Sans ce paramètre, l'historique des activités ne sera pas enregistré dans la base de données et n'apparaîtra pas dans le journal d'audit.



Le journal d'audit affiche les données suivantes :

Column	Description
Time	Horodatage de l'enregistrement d'audit.
User	Utilisateur ayant effectué l'activité.

Column	Description
IP	Adresse IP depuis laquelle l'activité a été initiée. Cliquer sur le lien hypertexte filtrera les enregistrements du journal d'audit par cette adresse IP.
Resource	Type de la ressource affectée (*API token*, *Action*, *Authentication*, *Autoregistration*, etc.).
ID	ID de la ressource affectée. Cliquer sur le lien hypertexte filtrera les enregistrements du journal d'audit par cet ID de ressource.
Action	Type de l'activité (*Add*, *Configuration refresh*, *Delete*, *Execute*, *Failed login*, *History clear*, *Login*, *Logout*, *Push*, *Update*).
Recordset ID	ID partagé pour tous les enregistrements du journal d'audit créés à la suite de la même opération. Par exemple, lors de la liaison d'un modèle à un hôte, un enregistrement distinct du journal d'audit est créé pour chaque entité de modèle héritée (élément, déclencheur, etc.) - tous ces enregistrements auront le même *Recordset ID*. Cliquer sur le lien hypertexte filtrera les enregistrements du journal d'audit par ce *Recordset ID*.
Details	Description de la ressource et informations détaillées sur l'activité effectuée. Si un enregistrement contient plus de deux lignes, un lien *Details* supplémentaire sera affiché. Cliquez sur ce lien pour voir la liste complète des modifications.

Note:
Lorsqu'un élément trapper ou un élément HTTP agent (avec la capture activée) a reçu des données, une entrée dans le journal d'audit ne sera ajoutée que si les données ont été envoyées à l'aide de la méthode API `history.push`, et non de l'utilitaire Zabbix sender.

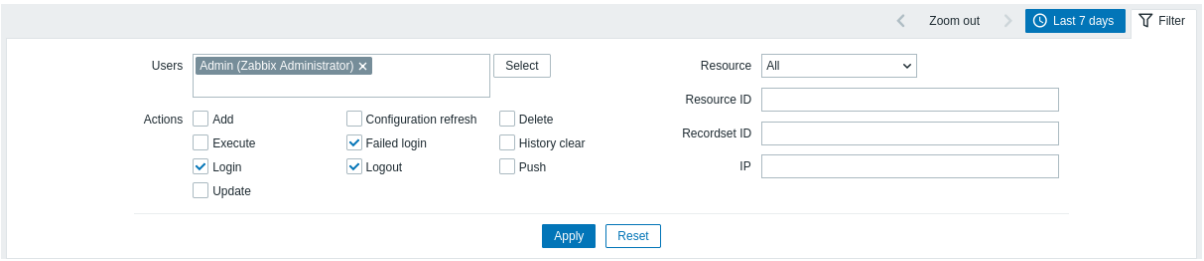
Boutons

Le bouton situé dans le coin supérieur droit de la page propose l'option suivante :

Export to CSV	Exporter les enregistrements du journal d'audit de toutes les pages vers un fichier CSV. Si un filtre est appliqué, seuls les enregistrements filtrés seront exportés.

Utilisation du filtre

Le filtre se trouve sous la barre du *Journal d'audit*. Il peut être ouvert et réduit en cliquant sur l'onglet *Filtre* dans le coin supérieur droit.



Vous pouvez utiliser le filtre pour restreindre les enregistrements par utilisateur, ressource affectée, ID de ressource, opération effectuée (*ID de l'ensemble d'enregistrements*) et IP. Selon la ressource, une ou plusieurs actions spécifiques peuvent être sélectionnées dans le filtre.

Pour de meilleures performances de recherche, toutes les données sont recherchées avec les macros non résolues.

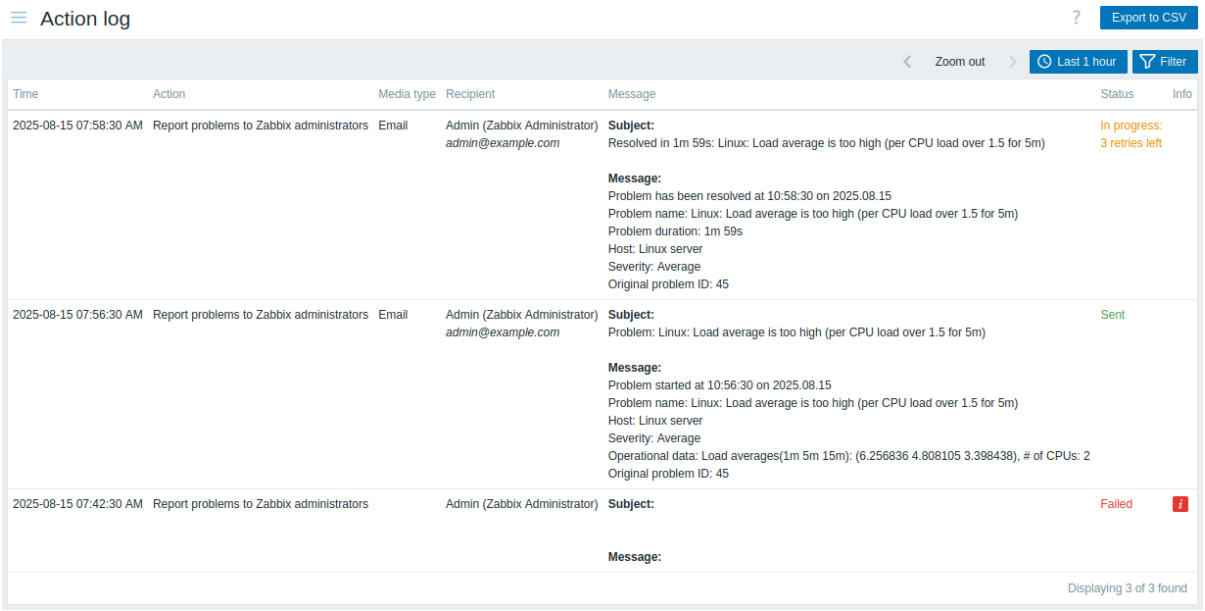
Sélecteur de période

Le sélecteur de période situé à côté du filtre vous permet de sélectionner les périodes fréquemment utilisées en un seul clic de souris. Pour plus d'informations, consultez la page Sélecteurs de période et d'hôte.

Journal des actions

Aperçu

Dans *Rapports* > *Journal des actions*, vous pouvez consulter les détails des opérations (notifications, commandes à distance) exécutées dans le cadre d'une action.



Données affichées :

Column	Description
Time	Horodatage de l'opération.
Action	Nom de l'action à l'origine des opérations.
Media type	Type de média (par exemple Email, Jira, etc.) utilisé pour l'envoi d'une notification. Pour les opérations ayant exécuté des commandes à distance, cette colonne sera vide.
Recipient	Informations sur le destinataire de la notification - nom d'utilisateur, prénom et nom (entre parenthèses), ainsi que des informations supplémentaires selon le type de média (e-mail, nom d'utilisateur, etc.). Pour les opérations ayant exécuté des commandes à distance, cette colonne sera vide.
Message	Le contenu du message, de la commande à distance ou du nom du script global. Une commande à distance est séparée de l'hôte cible par un symbole deux-points : `<host>:<command>`. Par exemple, si la commande à distance a été exécutée sur le serveur Zabbix, l'information aura le format suivant : `Zabbix server:<command>`.
Status	Statut de l'opération : *In progress* - l'opération d'envoi d'une notification est en cours (le nombre restant de tentatives du serveur pour envoyer la notification est également affiché) *Sent* - la notification a été envoyée *Executed* - la commande à distance a été exécutée *Failed* - l'opération n'a pas été terminée
Info	Informations d'erreur (le cas échéant) concernant l'exécution de l'opération.

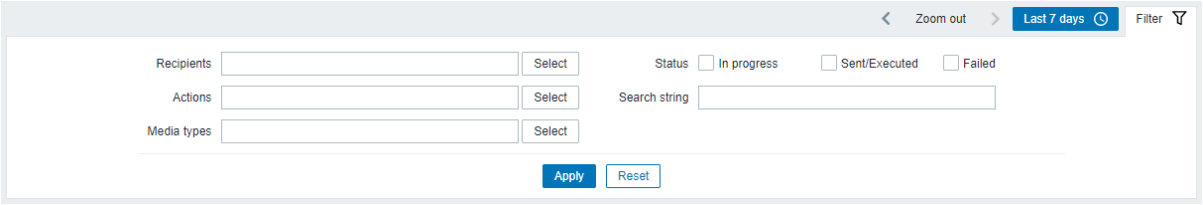
Boutons

Le bouton situé dans le coin supérieur droit de la page propose l'option suivante :

Export to CSV	Exporter les enregistrements du journal des actions de la page actuelle vers un fichier CSV. Si un filtre est appliqué, seuls les enregistrements filtrés seront exportés. Dans le fichier CSV exporté, les colonnes « Recipient » et « Message » sont divisées en plusieurs colonnes : « Recipient's Zabbix username », « Recipient's name », « Recipient's surname », « Recipient », ainsi que « Subject », « Message », « Command ».

Utilisation du filtre

Le filtre se trouve sous la barre *Action log*. Il peut être ouvert et réduit en cliquant sur l'onglet *Filter* dans le coin supérieur droit de la page.



Vous pouvez utiliser le filtre pour restreindre les enregistrements par destinataires des notifications, actions, types de média, statut, ou par le contenu du message/de la commande distante (*Search string*). Pour de meilleures performances de recherche, les données sont recherchées avec les macros non résolues.

Sélecteur de période

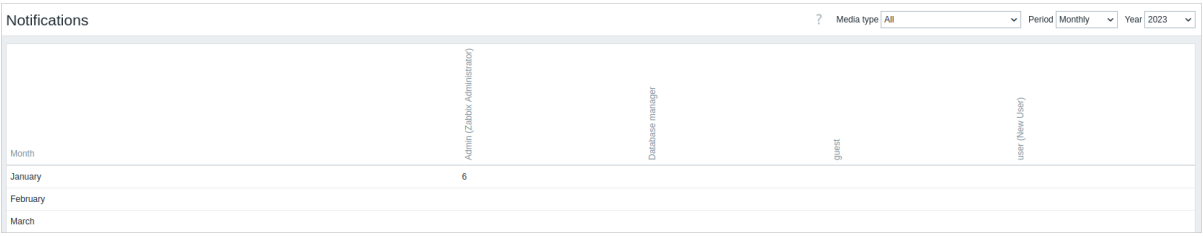
Le sélecteur de période situé à côté du filtre vous permet de sélectionner les périodes fréquemment requises en un seul clic de souris. Pour plus d'informations, consultez Sélecteurs de période et de hôte.

Notifications

Vue d'ensemble

Dans la section *Reports* > *Notifications*, un rapport sur le nombre de notifications envoyées à chaque utilisateur est affiché.

À partir des listes déroulantes dans le coin supérieur droit, vous pouvez choisir le type de média (ou tous), la période (données pour chaque jour/semaine/mois/année) et l'année pour les notifications envoyées.



Chaque colonne affiche les totaux pour un utilisateur système.

Collecte de données

Aperçu

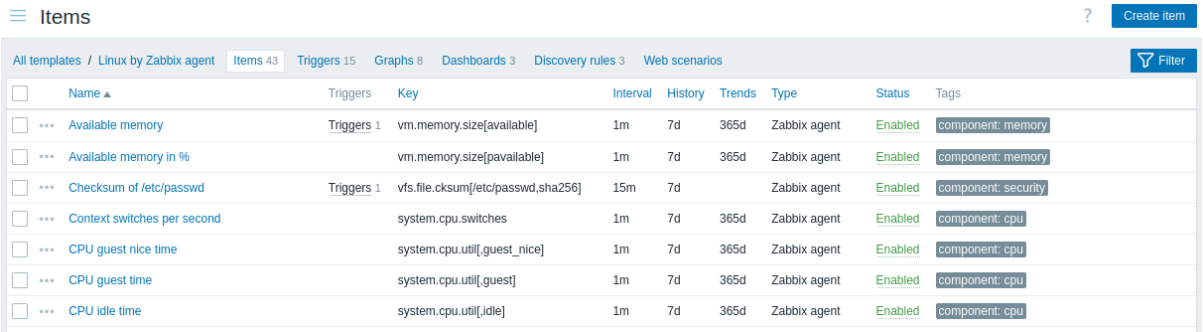
Ce menu comporte des sections liées à la configuration de la collecte de données.

Éléments

Vue d'ensemble

La liste des éléments d'un modèle est accessible depuis *Collecte de données* > *Modèles*, en cliquant sur *Éléments* pour le modèle concerné.

Une liste des éléments existants s'affiche.



Données affichées :

Column	Description
Menu de l'élément	Cliquez sur l'icône à trois points pour ouvrir le menu de l'élément.
Modèle	Modèle auquel appartient l'élément. Cliquer sur le nom du modèle ouvre le formulaire de configuration du modèle. Cette colonne s'affiche uniquement si plusieurs modèles ou aucun modèle ne sont sélectionnés dans le filtre.
Nom	Nom de l'élément affiché sous forme de lien bleu vers les détails de l'élément. Cliquer sur le lien du nom de l'élément ouvre le formulaire de configuration de l'élément. Si l'élément est hérité d'un autre modèle, le nom du modèle est affiché avant le nom de l'élément sous forme de lien gris. Cliquer sur le lien du modèle ouvre la liste des éléments à ce niveau du modèle.
Déclencheurs	Placer le curseur sur Déclencheurs affiche une infobulle contenant les déclencheurs associés à l'élément. Le nombre de déclencheurs est affiché en gris.
Clé	La clé de l'élément est affichée.
Intervalle	La fréquence de la vérification est affichée.
Historique	Le nombre de jours pendant lesquels l'historique des données de l'élément sera conservé est affiché.
Tendances	Le nombre de jours pendant lesquels l'historique des tendances de l'élément sera conservé est affiché.
Type	Le type d'élément est affiché (agent Zabbix, agent SNMP, vérification simple, etc.).
État	L'état de l'élément est affiché : *Activé* ou *Désactivé*. Cliquer sur l'état permet de le modifier, d'Activé à Désactivé (et inversement).
Étiquettes	Les étiquettes de l'élément sont affichées. Jusqu'à trois étiquettes (paires nom:valeur) peuvent être affichées. Si l'élément comporte davantage d'étiquettes, un lien « ... » s'affiche et permet de consulter toutes les étiquettes en plaçant le curseur dessus.

Pour configurer un nouvel élément, cliquez sur le bouton *Créer un élément* dans l'angle supérieur droit.

Options de modification en masse

Les boutons sous la liste offrent plusieurs options de modification en masse :

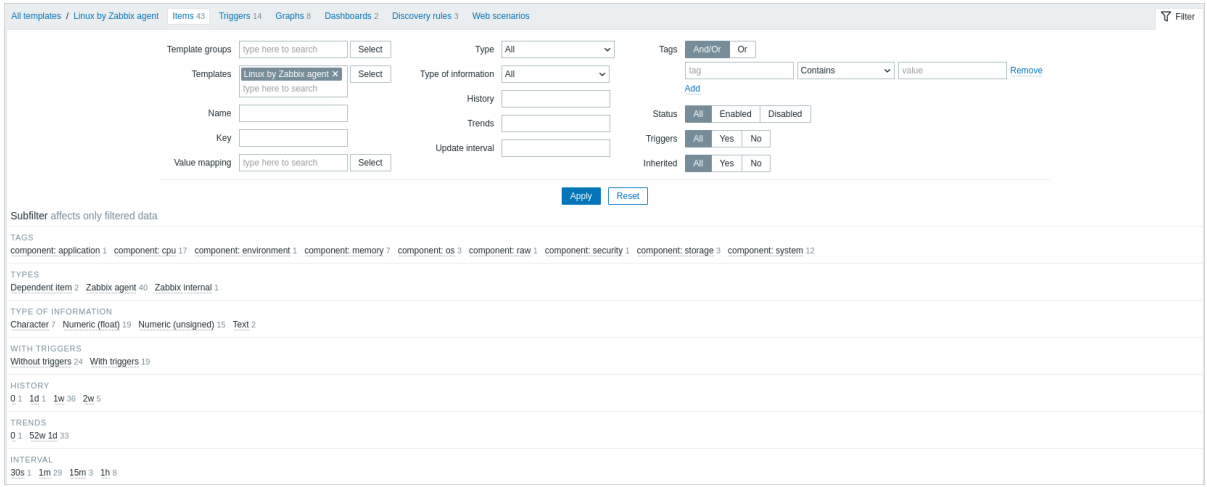
- *Enable* - changer le statut de l'élément en *Enabled*.
- *Disable* - changer le statut de l'élément en *Disabled*.
- *Copy* - copier les éléments vers d'autres hôtes ou modèles.
- *Mass update* - mettre à jour plusieurs propriétés pour un certain nombre d'éléments en une seule fois.
- *Delete* - supprimer les éléments.

Pour utiliser ces options, cochez les cases devant les éléments concernés, puis cliquez sur le bouton requis.

Utilisation du filtre

La liste des éléments peut contenir un grand nombre d'éléments. En utilisant le filtre, vous pouvez en filtrer certains afin de localiser rapidement les éléments que vous recherchez. Pour de meilleures performances de recherche, les données sont recherchées avec les macros non résolues.

L'icône *Filtre* est disponible dans le coin supérieur droit. En cliquant dessus, vous ouvrirez un filtre dans lequel vous pouvez spécifier les critères de filtrage souhaités.

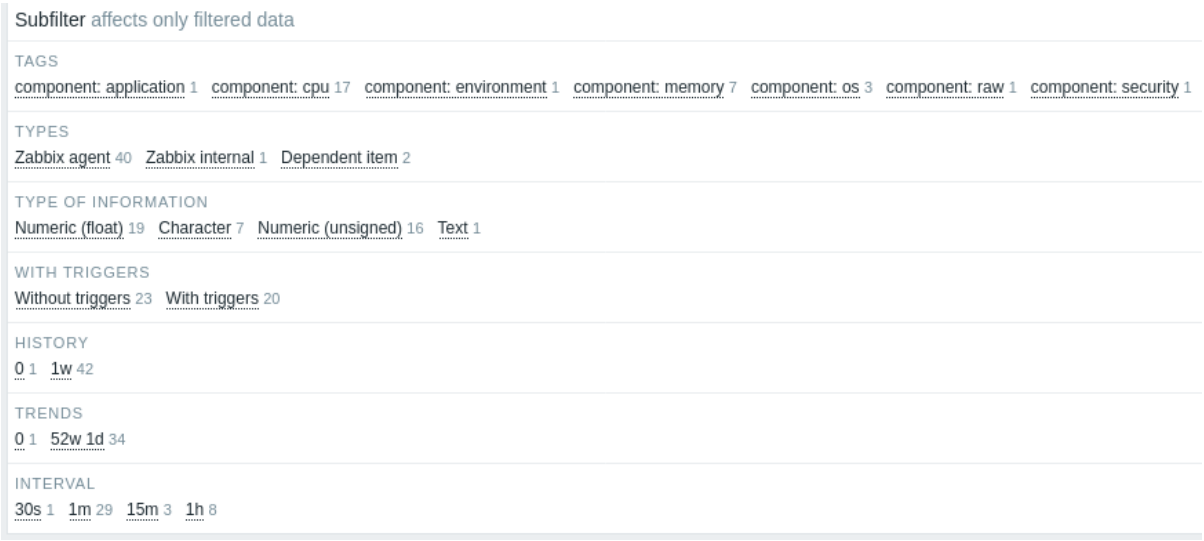


Parameter	Description
Template groups	Filtrer par un ou plusieurs groupes de modèles. La spécification d'un groupe de modèles parent sélectionne implicitement tous les groupes imbriqués.
Templates	Filtrer par un ou plusieurs modèles.
Name	Filtrer par nom d'élément.
Key	Filtrer par clé d'élément.
Value mapping	Filtrer par la carte de valeurs utilisée. Ce paramètre n'est pas affiché si l'option *Templates* est vide.
Type	Filtrer par type d'élément (Zabbix agent, agent SNMP, etc.).
Type of information	Filtrer par type d'information (entier non signé, flottant, etc.).
History	Filtrer par la durée de conservation de l'historique de l'élément.
Trends	Filtrer par la durée de conservation des tendances de l'élément.
Update interval	Filtrer par l'intervalle de mise à jour de l'élément.
Tags	Spécifiez des balises pour limiter le nombre d'éléments affichés. Il est possible d'inclure ainsi que d'exclure des balises et des valeurs de balise spécifiques. Plusieurs conditions peuvent être définies. La correspondance du nom de balise est toujours sensible à la casse. Plusieurs opérateurs sont disponibles pour chaque condition : **Exists** - inclure les noms de balise spécifiés **Equals** - inclure les noms de balise et les valeurs spécifiés (sensible à la casse) **Contains** - inclure les noms de balise spécifiés dont les valeurs de balise contiennent la chaîne saisie (correspondance de sous-chaîne, insensible à la casse) **Does not exist** - exclure les noms de balise spécifiés **Does not equal** - exclure les noms de balise et les valeurs spécifiés (sensible à la casse) **Does not contain** - exclure les noms de balise spécifiés dont les valeurs de balise contiennent la chaîne saisie (correspondance de sous-chaîne, insensible à la casse) Il existe deux types de calcul pour les conditions : **And/Or** - toutes les conditions doivent être remplies, les conditions ayant le même nom de balise seront regroupées par la condition Or **Or** - il suffit qu'une condition soit remplie
Status	Filtrer par statut de l'élément - *Enabled* ou *Disabled*.
Triggers	Filtrer les éléments avec des déclencheurs (ou sans déclencheurs).
Inherited	Filtrer les éléments hérités (ou non hérités) des modèles liés.

Utilisation du sous-filtre

Le sous-filtre permet de modifier davantage le filtrage appliqué par le filtre principal.

Il contient des liens cliquables permettant d'accéder rapidement aux éléments associés. Les éléments sont associés par une entité commune : balise, type d'élément, intervalle de mise à jour de l'élément, etc. Lorsqu'une entité est sélectionnée, elle est mise en évidence par un arrière-plan gris et les éléments sont immédiatement filtrés (il n'est pas nécessaire de cliquer sur *Appliquer* dans le filtre principal). La sélection d'une autre entité l'ajoute aux résultats filtrés. La sélection de l'entité une nouvelle fois supprime le filtrage.



Les sous-filtres sont générés à partir des données filtrées, limitées à 1000 enregistrements. Si vous souhaitez afficher davantage d'enregistrements dans le sous-filtre, vous devez augmenter la valeur du paramètre *Limite des résultats de recherche et de filtrage* (dans *Administration* > *Général* > *Interface*).

Contrairement au filtre principal, le sous-filtre est mis à jour à chaque demande d'actualisation du tableau afin de toujours afficher des informations à jour sur les options de filtrage disponibles et leur nombre.

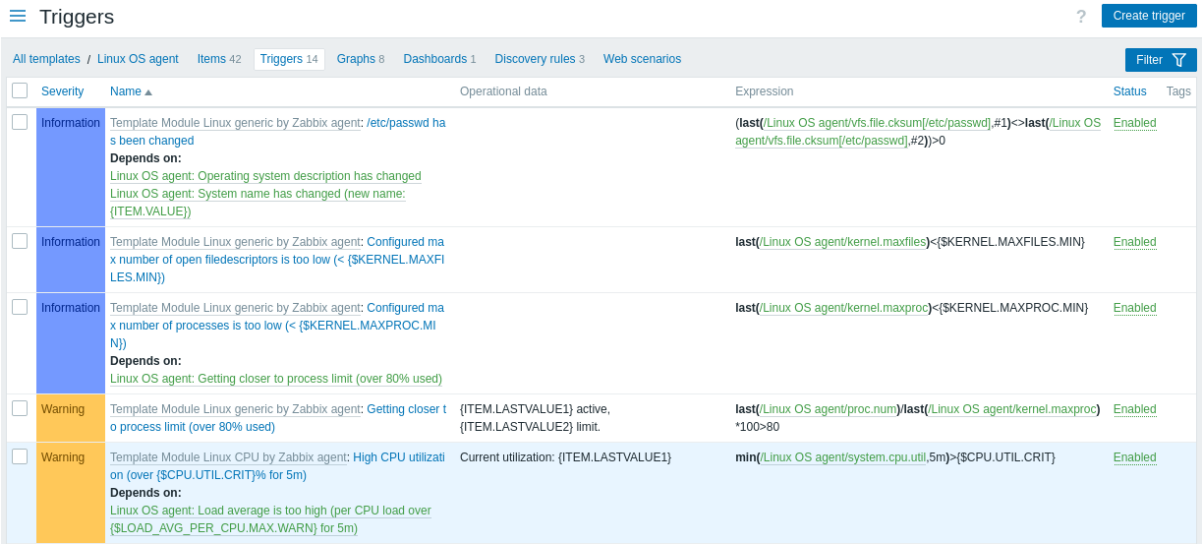
Le nombre d'entités affichées est limité à 100 horizontalement. S'il y en a davantage, une icône représentant trois points est affichée à la fin ; elle n'est pas cliquable.

Un nombre placé à côté de chaque entité cliquable indique le nombre d'éléments qui y sont regroupés (selon les résultats du filtre principal). Lorsqu'une entité est sélectionnée, les nombres associés aux autres entités disponibles sont affichés avec un signe plus indiquant combien d'éléments peuvent être ajoutés à la sélection actuelle. Les entités sans éléments ne sont pas affichées, sauf si elles ont été sélectionnées auparavant dans le sous-filtre.

Déclencheurs

Vue d'ensemble

La liste des déclencheurs d'un modèle est accessible depuis *Collecte de données* > *Modèles* en cliquant sur *Déclencheurs* pour le modèle concerné.



Données affichées :

Colonne	Description
Gravité	La gravité du déclencheur est affichée à la fois par son nom et par la couleur d'arrière-plan de la cellule.

Colonne	Description
Modèle	Modèle auquel appartient le déclencheur. Cliquer sur le nom du modèle ouvre le formulaire de configuration du modèle. Cette colonne est affichée uniquement si plusieurs modèles ou aucun modèle ne sont sélectionnés dans le filtre.
Nom	Nom du déclencheur affiché sous forme de lien bleu vers les détails du déclencheur. Cliquer sur le lien du nom du déclencheur ouvre le formulaire de configuration du déclencheur. Si le déclencheur est hérité d'un autre modèle, le nom du modèle est affiché avant le nom du déclencheur sous forme de lien gris. Cliquer sur le lien du modèle ouvre la liste des déclencheurs à ce niveau du modèle.
Données opérationnelles	Définition des données opérationnelles du déclencheur, contenant des chaînes arbitraires et des macros qui seront résolues dynamiquement dans *Supervision* > *Problèmes*.
Expression	L'expression du déclencheur est affichée. La partie de l'expression correspondant à l'élément du modèle est affichée sous forme de lien menant au formulaire de configuration de l'élément.
État	L'état du déclencheur est affiché : *Activé* ou *Désactivé*. Cliquer sur l'état permet de le modifier, d'Activé à Désactivé (et inversement).
Étiquettes	Si un déclencheur contient des étiquettes, leur nom et leur valeur sont affichés dans cette colonne.

Pour configurer un nouveau déclencheur, cliquez sur le bouton *Créer un déclencheur* dans l'angle supérieur droit.

Options de modification en masse

Les boutons sous la liste offrent plusieurs options de modification en masse :

- *Enable* - changer l'état du déclencheur en *Enabled*
- *Disable* - changer l'état du déclencheur en *Disabled*
- *Copy* - copier les déclencheurs vers d'autres hôtes ou modèles
- *Mass update* - mettre à jour plusieurs propriétés pour un certain nombre de déclencheurs en une seule fois
- *Delete* - supprimer les déclencheurs

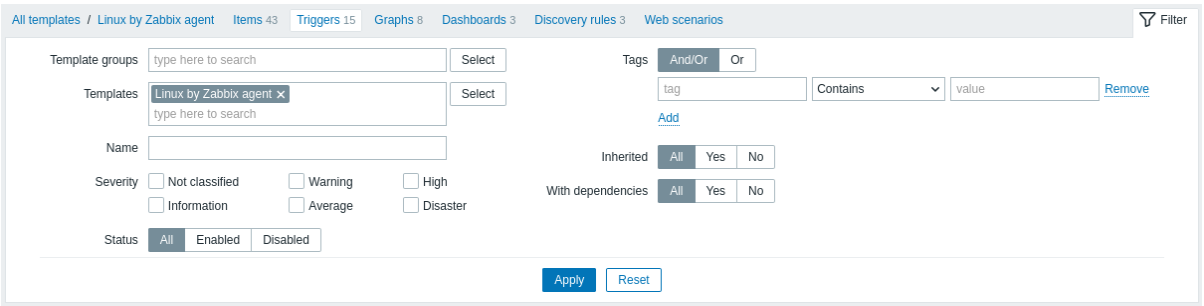
Pour utiliser ces options, cochez les cases devant les déclencheurs concernés, puis cliquez sur le bouton requis.

Utilisation du filtre

Vous pouvez utiliser le filtre pour afficher uniquement les déclencheurs qui vous intéressent.

Pour de meilleures performances de recherche, les données sont recherchées avec des macros non résolues.

L'icône *Filtre* est disponible dans le coin supérieur droit. En cliquant dessus, vous ouvrirez un filtre dans lequel vous pouvez spécifier les critères de filtrage souhaités.



Parameter	Description
Template groups	Filtrer par un ou plusieurs groupes de modèles. La spécification d'un groupe de modèles parent sélectionne implicitement tous les groupes imbriqués.
Templates	Filtrer par un ou plusieurs modèles. Si des groupes de modèles sont déjà sélectionnés ci-dessus, la sélection des modèles est limitée à ces groupes.
Name	Filtrer par nom de déclencheur.
Severity	Sélectionnez pour filtrer par une ou plusieurs sévérités de déclencheur.
Status	Filtrer par statut d'activation du déclencheur (Enabled/Disabled).

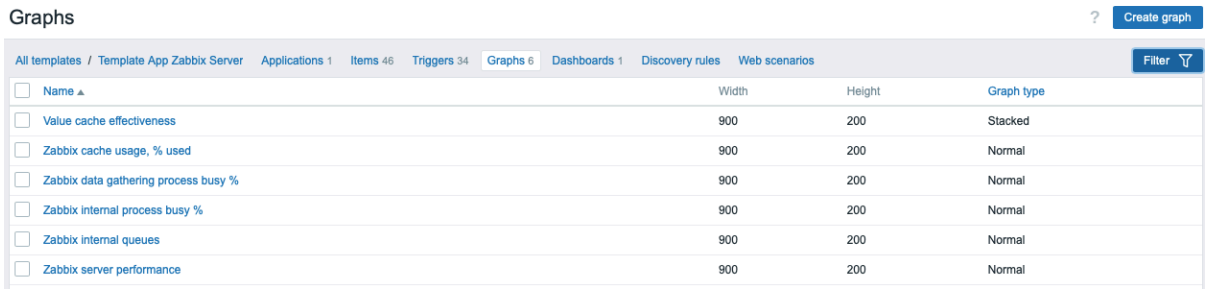
Parameter	Description
Tags	Filtrer par nom et valeur de tag de déclencheur. Il est possible d'inclure ainsi que d'exclure des tags et des valeurs de tag spécifiques. Plusieurs conditions peuvent être définies. La correspondance du nom du tag est toujours sensible à la casse. Plusieurs opérateurs sont disponibles pour chaque condition : **Exists** - inclure les noms de tag spécifiés **Equals** - inclure les noms et valeurs de tag spécifiés (sensible à la casse) **Contains** - inclure les noms de tag spécifiés dont les valeurs de tag contiennent la chaîne saisie (correspondance de sous-chaîne, insensible à la casse) **Does not exist** - exclure les noms de tag spécifiés **Does not equal** - exclure les noms et valeurs de tag spécifiés (sensible à la casse) **Does not contain** - exclure les noms de tag spécifiés dont les valeurs de tag contiennent la chaîne saisie (correspondance de sous-chaîne, insensible à la casse) Il existe deux types de calcul pour les conditions : **And/Or** - toutes les conditions doivent être remplies, les conditions ayant le même nom de tag seront regroupées par la condition Or **Or** - une seule condition suffit Les macros et les fonctions de macro sont prises en charge dans les champs nom de tag et valeur de tag.
Inherited	Filtrer les déclencheurs hérités (ou non hérités) des modèles liés.
With dependencies	Filtrer les déclencheurs avec dépendances (ou sans dépendances).

Graphiques

Vue d'ensemble

La liste des graphes personnalisés d'un modèle est accessible depuis *Collecte de données* > *Modèles* en cliquant sur *Graphes* pour le modèle concerné.

Une liste des graphes existants s'affiche.



Données affichées :

Colonne	Description
Modèle	Modèle auquel appartient le graphe. Cliquer sur le nom du modèle ouvre le formulaire de configuration du modèle. Cette colonne s'affiche uniquement si plusieurs modèles ou aucun modèle ne sont sélectionnés dans le filtre.
Nom	Nom du graphe personnalisé, affiché sous forme de lien bleu vers les détails du graphe. Cliquer sur le lien du nom du graphe ouvre le formulaire de configuration du graphe. Si le graphe est hérité d'un autre modèle, le nom du modèle est affiché avant le nom du graphe sous forme de lien gris. Cliquer sur le lien du modèle ouvre la liste des graphes à ce niveau de modèle.
Largeur	La largeur du graphe est affichée.
Hauteur	La hauteur du graphe est affichée.
Type de graphe	Le type de graphe est affiché : *Normal*, *Empilé*, *Camembert* ou *Éclaté*.

Pour configurer un nouveau graphe, cliquez sur le bouton *Créer un graphe* dans l'angle supérieur droit.

Options de modification en masse

Les boutons sous la liste offrent plusieurs options de modification en masse :

- *Copier* - copier les graphiques vers d'autres hôtes ou modèles
- *Supprimer* - supprimer les graphiques

Pour utiliser ces options, cochez les cases devant les graphiques concernés, puis cliquez sur le bouton souhaité.

Utilisation du filtre

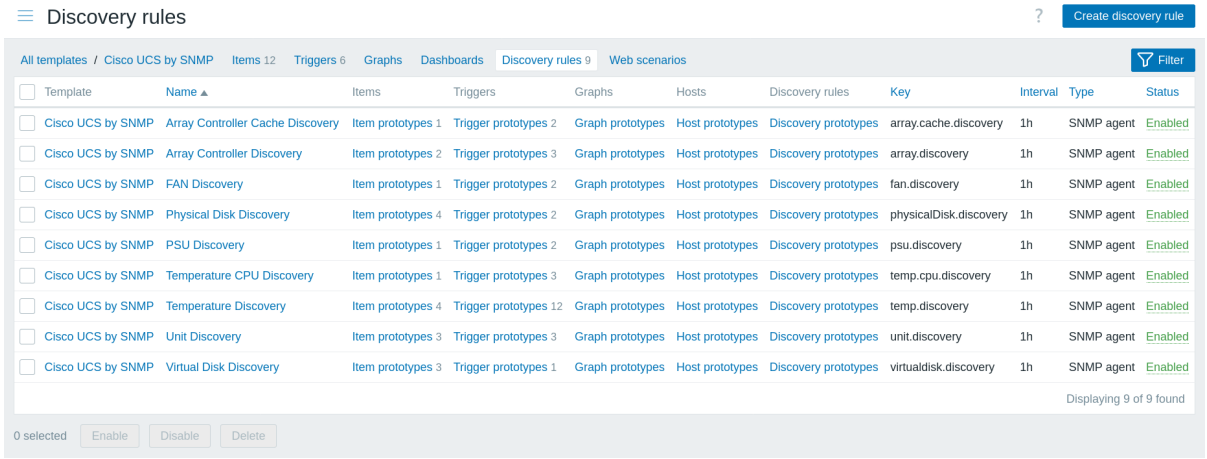
Vous pouvez filtrer les graphiques par groupe de modèles et par modèle. Pour de meilleures performances de recherche, les données sont recherchées avec les macros non résolues.

Règles de découverte

Vue d'ensemble

La liste des règles de découverte de bas niveau d'un modèle est accessible depuis *Collecte de données* > *Modèles* en cliquant sur *Découverte* pour le modèle correspondant.

Une liste des règles de découverte de bas niveau existantes s'affiche. Il est également possible d'afficher toutes les règles de découverte indépendamment du modèle, ou toutes les règles de découverte d'un groupe de modèles spécifique en modifiant les paramètres du filtre.



Données affichées :

Colonne	Description
Modèle	Modèle auquel appartient la règle de découverte. Cliquer sur le nom du modèle ouvre le formulaire de configuration du modèle.
Nom	Nom de la règle, affiché sous forme de lien bleu. Cliquer sur le nom de la règle ouvre le formulaire de configuration de la règle de découverte de bas niveau. Si la règle de découverte est héritée d'un autre modèle, le nom du modèle est affiché avant le nom de la règle sous forme de lien gris. Cliquer sur le lien du modèle ouvre la liste des règles de découverte à ce niveau du modèle.
Éléments	Un lien vers la liste des prototypes d'éléments est affiché. Le nombre de prototypes d'éléments existants est affiché en gris.
Déclencheurs	Un lien vers la liste des prototypes de déclencheurs est affiché. Le nombre de prototypes de déclencheurs existants est affiché en gris.
Graphiques	Un lien vers la liste des prototypes de graphiques est affiché. Le nombre de prototypes de graphiques existants est affiché en gris.
Hôtes	Un lien vers la liste des prototypes d'hôtes est affiché. Le nombre de prototypes d'hôtes existants est affiché en gris.
Clé	La clé de l'élément utilisée pour la découverte est affichée.
Intervalle	La fréquence d'exécution de la découverte est affichée.
Type	Le type d'élément utilisé pour la découverte est affiché (agent Zabbix, agent SNMP, etc.).
État	L'état de la règle de découverte est affiché : *Activée* ou *Désactivée*. Cliquer sur l'état permet de le modifier, d'Activée à Désactivée (et inversement).

Pour configurer une nouvelle règle de découverte de bas niveau, cliquez sur le bouton *Créer une règle de découverte* dans l'angle supérieur droit.

Options de modification en masse

Les boutons sous la liste offrent plusieurs options de modification en masse :

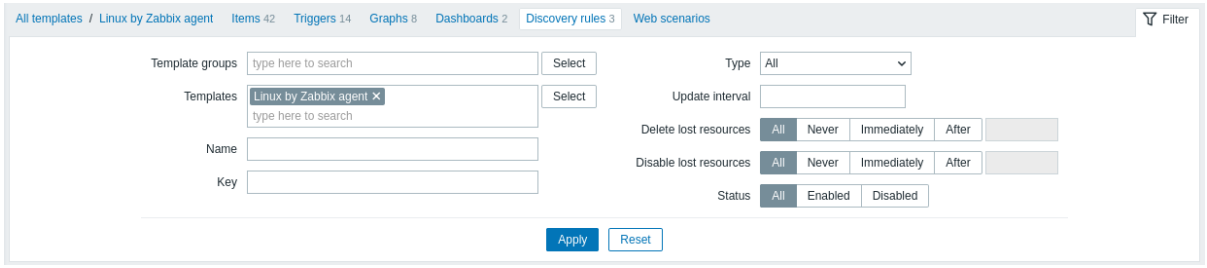
- *Enable* - changer l'état de la règle de découverte de bas niveau en *Enabled*
- *Disable* - changer l'état de la règle de découverte de bas niveau en *Disabled*
- *Delete* - supprimer les règles de découverte de bas niveau

Pour utiliser ces options, cochez les cases devant les règles de découverte concernées, puis cliquez sur le bouton requis.

Utilisation du filtre

Vous pouvez utiliser le filtre pour afficher uniquement les règles de découverte qui vous intéressent. Pour de meilleures performances de recherche, la recherche des données s'effectue avec les macros non résolues.

L'icône *Filter* est disponible dans le coin supérieur droit. En cliquant dessus, vous ouvrirez un filtre dans lequel vous pouvez spécifier les critères de filtrage souhaités, tels que le modèle, le nom de la règle de découverte, la clé de l'élément, le type d'élément, etc.



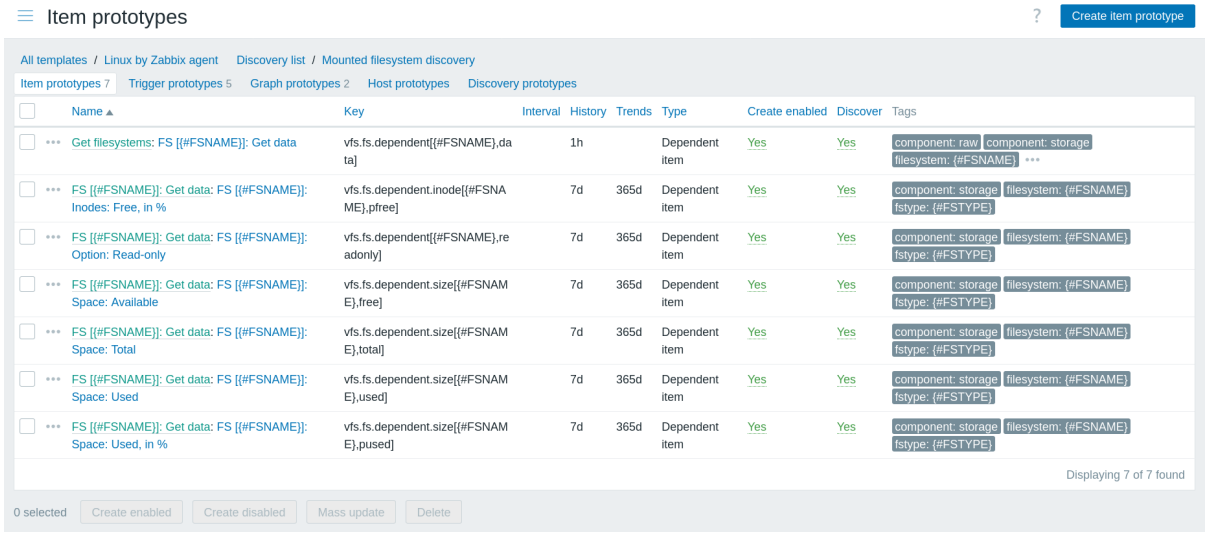
Parameter	Description
Template groups	Filtrer par un ou plusieurs groupes de modèles. La spécification d'un groupe de modèles parent sélectionne implicitement tous les groupes imbriqués.
Templates	Filtrer par un ou plusieurs modèles.
Name	Filtrer par le nom de la règle de découverte.
Key	Filtrer par la clé de l'élément de découverte.
Type	Filtrer par le type de l'élément de découverte.
Update interval	Filtrer par l'intervalle de mise à jour. Non disponible pour les éléments Zabbix trapper et dépendants.
Delete lost resources	Filtrer par la période *Delete lost resources*.
Disable lost resources	Filtrer par la période *Disable lost resources*.
Status	Filtrer par le statut de la règle de découverte (All/Enabled/Disabled).

Prototypes d'élément

Vue d'ensemble

Dans cette section, les prototypes d'éléments configurés d'une règle de découverte de bas niveau sur le modèle sont affichés.

Si le modèle est lié à l'hôte, les prototypes d'éléments serviront de base à la création des éléments réels de l'hôte lors de la découverte de bas niveau.



Données affichées :

Column	Description
Name	Nom du prototype d'élément, affiché sous forme de lien bleu. Cliquer sur le nom ouvre le formulaire de configuration du prototype d'élément. Si le prototype d'élément appartient à un modèle lié, le nom du modèle est affiché avant le nom du prototype d'élément sous forme de lien gris. Cliquer sur le lien du modèle ouvre la liste des prototypes d'éléments au niveau du modèle lié. Si le prototype d'élément est de type « Élément dépendant », le nom de l'élément maître est affiché en vert avant le nom du prototype d'élément (comme dans le cas des prototypes d'éléments au niveau de l'hôte).
Key	La clé du prototype d'élément est affichée.
Interval	La fréquence de la vérification est affichée.
History	Le nombre de jours pendant lesquels conserver l'historique des données de l'élément est affiché.
Trends	Le nombre de jours pendant lesquels conserver l'historique des tendances de l'élément est affiché.
Type	Le type du prototype d'élément est affiché (agent Zabbix, agent SNMP, vérification simple, etc.).
Create enabled	Créer l'élément sur la base de ce prototype comme suit : **Yes** - activé **No** - désactivé. Vous pouvez basculer entre « Yes » et « No » en cliquant dessus.
Discover	Découvrir l'élément sur la base de ce prototype : **Yes** - découvrir **No** - ne pas découvrir. Vous pouvez basculer entre « Yes » et « No » en cliquant dessus.
Tags	Les étiquettes du prototype d'élément sont affichées.

Pour configurer un nouveau prototype d'élément, cliquez sur le bouton *Create item prototype* dans l'angle supérieur droit.

Options de modification en masse

Les boutons sous la liste offrent plusieurs options de modification en masse :

- *Créer activé* - créer ces éléments comme *Activé*
- *Créer désactivé* - créer ces éléments comme *Désactivé*
- *Mise à jour en masse* - mettre à jour en masse ces prototypes d'élément
- *Supprimer* - supprimer ces prototypes d'élément

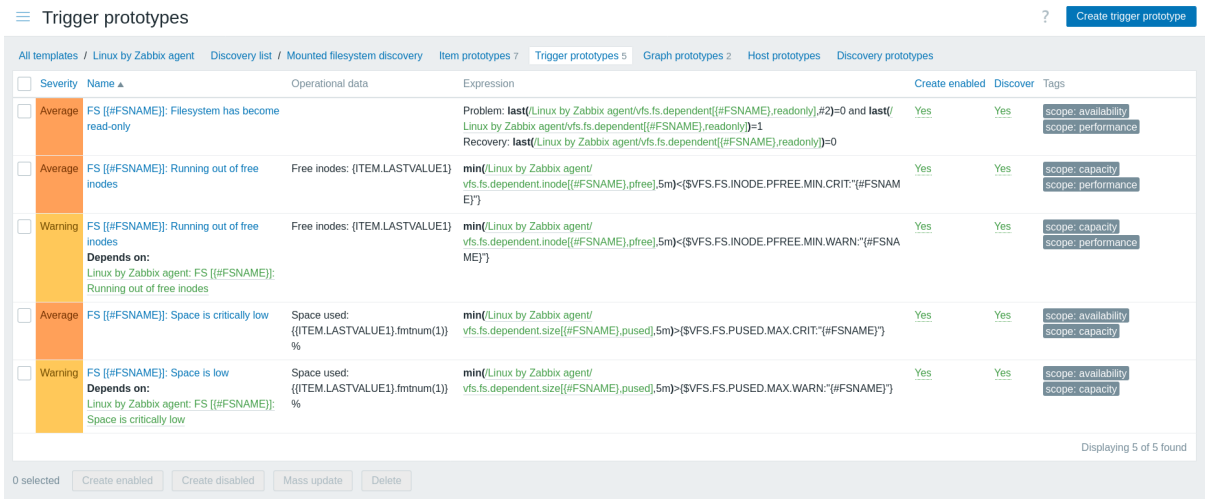
Pour utiliser ces options, cochez les cases à cocher devant les prototypes d'élément concernés, puis cliquez sur le bouton requis.

Prototypes de déclencheur

Vue d'ensemble

Dans cette section, les prototypes de déclencheurs configurés d'une règle de découverte de bas niveau sur le modèle sont affichés.

Si le modèle est lié à l'hôte, les prototypes de déclencheurs serviront de base à la création de véritables déclencheurs sur l'hôte lors de la découverte de bas niveau.



Données affichées :

Column	Description
Gravité	La gravité du déclencheur est affichée à la fois par son nom et par la couleur d'arrière-plan de la cellule.
Nom	Nom du prototype de déclencheur, affiché sous forme de lien bleu. Cliquer sur le nom ouvre le formulaire de configuration du prototype de déclencheur. Si le prototype de déclencheur appartient à un modèle lié, le nom du modèle est affiché avant le nom du déclencheur sous forme de lien gris. Cliquer sur le lien du modèle ouvre la liste des prototypes de déclencheurs au niveau du modèle lié.
Expression	L'expression du déclencheur est affichée. La partie de l'expression correspondant à l'élément du modèle est affichée sous forme de lien menant au formulaire de configuration de l'élément.
Données opérationnelles	Le format des données opérationnelles du déclencheur est affiché. Il peut contenir des chaînes arbitraires et des macros qui seront résolues dynamiquement dans *Supervision > Problèmes*.
Création activée	Créer le déclencheur sur la base de ce prototype comme suit : **Oui** - activé **Non** - désactivé. Vous pouvez basculer entre « Oui » et « Non » en cliquant dessus.
Découverte	Découvrir le déclencheur sur la base de ce prototype : **Oui** - découvrir **Non** - ne pas découvrir. Vous pouvez basculer entre « Oui » et « Non » en cliquant dessus.
Étiquettes	Les étiquettes du prototype de déclencheur sont affichées.

Pour configurer un nouveau prototype de déclencheur, cliquez sur le bouton *Créer un prototype de déclencheur* dans l'angle supérieur droit.

Options de modification en masse

Les boutons sous la liste offrent plusieurs options de modification en masse :

- *Créer activé* - créer ces déclencheurs comme *Activé*
- *Créer désactivé* - créer ces déclencheurs comme *Désactivé*
- *Mise à jour en masse* - mettre à jour en masse ces prototypes de déclencheur
- *Supprimer* - supprimer ces prototypes de déclencheur

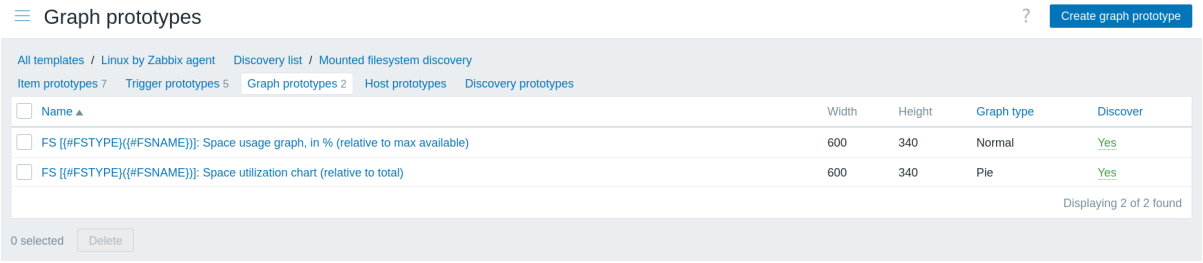
Pour utiliser ces options, cochez les cases à cocher devant les prototypes de déclencheur correspondants, puis cliquez sur le bouton requis.

Prototypes de graphique

Vue d'ensemble

Dans cette section, les prototypes de graphiques configurés d'une règle de découverte de bas niveau sur le modèle sont affichés.

Si le modèle est lié à l'hôte, les prototypes de graphiques serviront de base à la création de graphiques réels sur l'hôte lors de la découverte de bas niveau.



Données affichées :

Column	Description
Name	Nom du prototype de graphique, affiché sous forme de lien bleu. Cliquer sur le nom ouvre le formulaire de configuration du prototype de graphique. Si le prototype de graphique appartient à un modèle lié, le nom du modèle est affiché avant le nom du graphique sous forme de lien gris. Cliquer sur le lien du modèle ouvre la liste des prototypes de graphiques au niveau du modèle lié.
Width	La largeur du prototype de graphique est affichée.
Height	La hauteur du prototype de graphique est affichée.
Type	Le type du prototype de graphique est affiché : *Normal*, *Stacked*, *Pie* ou *Exploded*.
Discover	Découvrir le graphique sur la base de ce prototype : **Yes** - découvrir **No** - ne pas découvrir. Vous pouvez basculer entre « Yes » et « No » en cliquant dessus.

Pour configurer un nouveau prototype de graphique, cliquez sur le bouton *Create graph prototype* dans l'angle supérieur droit.

Options de modification en masse

Les boutons sous la liste offrent plusieurs options de modification en masse :

- *Supprimer* - supprimer ces prototypes de graphique

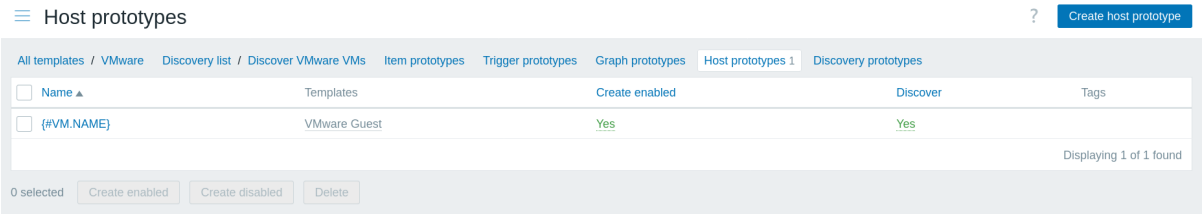
Pour utiliser ces options, cochez les cases devant les prototypes de graphique concernés, puis cliquez sur le bouton requis.

Modèles d'hôte

Vue d'ensemble

Dans cette section, les prototypes d'hôtes configurés d'une règle de découverte de bas niveau sur le modèle sont affichés.

Si le modèle est lié à l'hôte, les prototypes d'hôtes serviront de base à la création d'hôtes réels lors de la découverte de bas niveau.



Données affichées :

Column	Description
Name	Nom du prototype d'hôte, affiché sous forme de lien bleu. Cliquer sur le nom ouvre le formulaire de configuration du prototype d'hôte. Si le prototype d'hôte appartient à un modèle lié, le nom du modèle est affiché avant le nom de l'hôte sous forme de lien gris. Cliquer sur le lien du modèle ouvre la liste des prototypes d'hôtes au niveau du modèle lié.
Templates	Les modèles du prototype d'hôte sont affichés.
Create enabled	Créer l'hôte sur la base de ce prototype comme suit : **Yes** - activé **No** - désactivé. Vous pouvez basculer entre « Yes » et « No » en cliquant dessus.

Column	Description
Discover	Découvrir l'hôte sur la base de ce prototype : **Yes** - découvrir **No** - ne pas découvrir. Vous pouvez basculer entre « Yes » et « No » en cliquant dessus.
Tags	Les tags du prototype d'hôte sont affichés.

Pour configurer un nouveau prototype d'hôte, cliquez sur le bouton *Create host prototype* dans l'angle supérieur droit.

Options de modification en masse

Les boutons sous la liste offrent plusieurs options de modification en masse :

- *Créer activé* - créer ces hôtes comme *Activé*
- *Créer désactivé* - créer ces hôtes comme *Désactivé*
- *Supprimer* - supprimer ces prototypes d'hôte

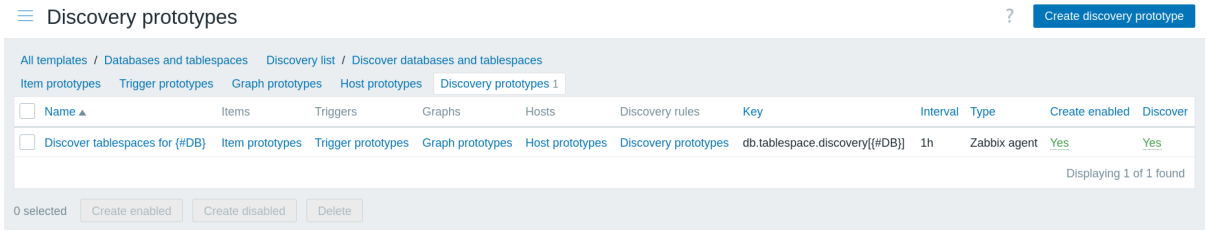
Pour utiliser ces options, cochez les cases devant les prototypes d'hôte correspondants, puis cliquez sur le bouton requis.

Prototypes de découverte

Vue d'ensemble

Dans cette section, les prototypes de découverte configurés d'une règle de découverte de bas niveau sur le modèle sont affichés.

Si le modèle est lié à l'hôte, les prototypes de découverte serviront de base à la création des prototypes de découverte réels de la règle de découverte parente.



Données affichées :

Column	Description
Name	Nom du prototype de découverte, affiché sous forme de lien bleu. Cliquer sur le nom ouvre le formulaire de configuration du prototype de découverte. Si le prototype de découverte appartient à un modèle lié, le nom du modèle est affiché avant le nom du prototype de découverte sous forme de lien gris. Cliquer sur le lien du modèle ouvre la liste des prototypes de découverte au niveau du modèle lié.
Items	Un lien vers la liste des prototypes d'éléments est affiché. Le nombre de prototypes d'éléments existants est affiché en gris.
Triggers	Un lien vers la liste des prototypes de déclencheurs est affiché. Le nombre de prototypes de déclencheurs existants est affiché en gris.
Graphs	Un lien vers la liste des prototypes de graphiques est affiché. Le nombre de prototypes de graphiques existants est affiché en gris.
Hosts	Un lien vers la liste des prototypes d'hôtes est affiché. Le nombre de prototypes d'hôtes existants est affiché en gris.
Discovery rules	Un lien vers la liste des prototypes de découverte est affiché. Le nombre de prototypes de découverte existants est affiché en gris.
Key	La clé de l'élément utilisée pour la découverte est affichée.
Interval	La fréquence d'exécution de la découverte est affichée.
Type	Le type d'élément utilisé pour la découverte est affiché (agent Zabbix, agent SNMP, etc.).
Create enabled	L'état de création du prototype de découverte est affiché : création activée (*Yes*) ou création désactivée (*No*). Cliquer sur l'état permet de le modifier, de Yes à No (et inversement).
Discover	L'état de découverte du prototype de découverte est affiché : découvrir (*Yes*) ou ne pas découvrir (*No*). Cliquer sur l'état permet de le modifier, de Yes à No (et inversement).

Pour configurer un nouveau prototype de découverte, cliquez sur le bouton *Create discovery prototype* dans l'angle supérieur droit.

Options de modification en masse

Les boutons sous la liste offrent plusieurs options de modification en masse :

- *Créer activé* - créer ces prototypes de découverte comme *Activé*
- *Créer désactivé* - créer ces prototypes de découverte comme *Désactivé*
- *Supprimer* - supprimer ces prototypes de découverte

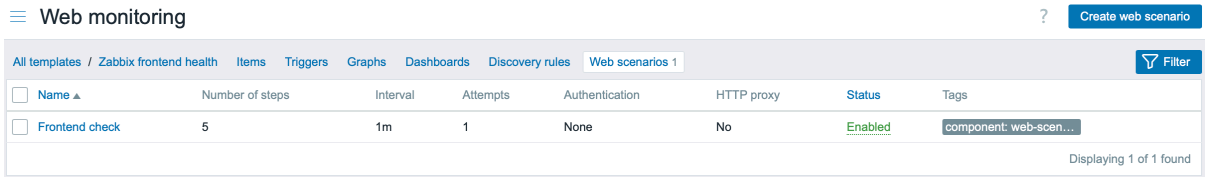
Pour utiliser ces options, cochez les cases devant les prototypes de découverte concernés, puis cliquez sur le bouton requis.

Scénarios web

Vue d'ensemble

La liste des scénarios web d'un modèle est accessible depuis *Collecte de données* > *Modèles*, en cliquant sur *Web* pour le modèle concerné.

Une liste des scénarios web existants s'affiche.



Données affichées :

Column	Description
Name	Nom du scénario web. Cliquez sur le nom du scénario web pour ouvrir le formulaire de configuration du scénario web. Si le scénario web est hérité d'un autre modèle, le nom du modèle est affiché avant le nom du scénario web sous forme de lien gris. Cliquez sur le lien du modèle pour ouvrir la liste des scénarios web à ce niveau de modèle.
Number of steps	Nombre d'étapes que contient le scénario.
Update interval	Fréquence d'exécution du scénario.
Attempts	Nombre de tentatives effectuées pour exécuter les étapes du scénario web.
Authentication	Méthode d'authentification affichée : Basic, NTLM, Kerberos, Digest ou None.
HTTP proxy	Affiche le proxy HTTP ou « No » s'il n'est pas utilisé.
Status	État du scénario web affiché : *Enabled* ou *Disabled*. Cliquez sur l'état pour le modifier.
Tags	Les tags du scénario web sont affichés. Jusqu'à trois tags (paires nom:valeur) peuvent être affichés. S'il y a davantage de tags, un lien « ... » permet de tous les afficher au survol de la souris.

Pour configurer un nouveau scénario web, cliquez sur le bouton *Create web scenario* dans l'angle supérieur droit.

Options de modification en masse

Les boutons sous la liste offrent plusieurs options de modification en masse :

- *Enable* - changer l'état du scénario en *Enabled*
- *Disable* - changer l'état du scénario en *Disabled*
- *Delete* - supprimer les scénarios web

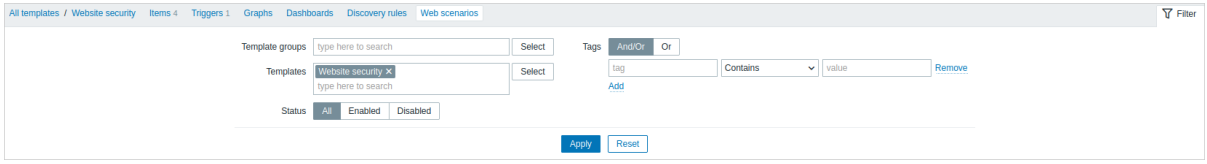
Pour utiliser ces options, cochez les cases devant les scénarios web concernés, puis cliquez sur le bouton souhaité.

Utilisation du filtre

Vous pouvez utiliser le filtre pour afficher uniquement les scénarios qui vous intéressent.

Pour de meilleures performances de recherche, les données sont recherchées avec les macros non résolues.

Le lien *Filtre* est disponible au-dessus de la liste des scénarios web. Si vous cliquez dessus, un filtre devient disponible, où vous pouvez filtrer les scénarios par groupe de modèles, modèle, statut et balises.

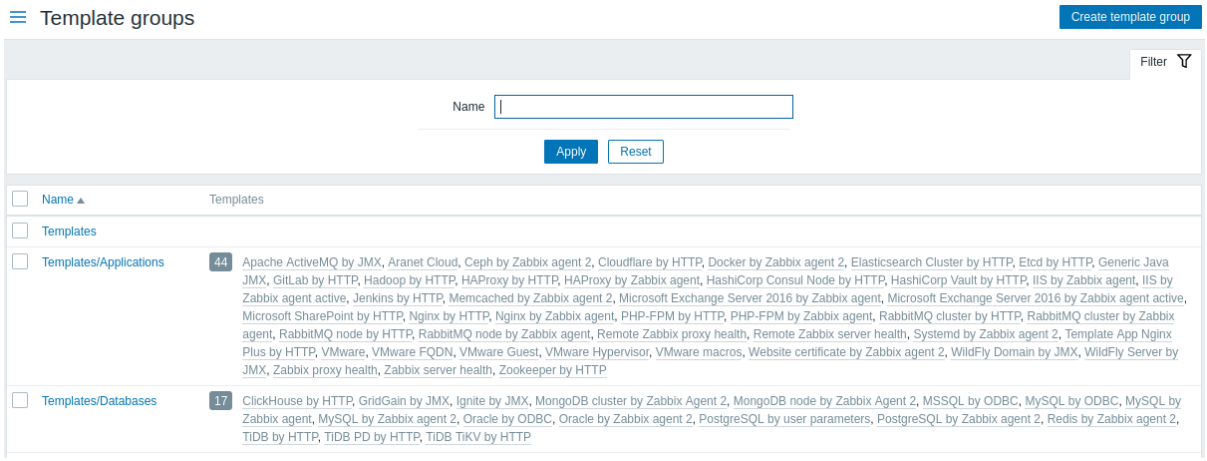


Groupes de modèles

Vue d'ensemble

Dans la section *Collecte de données* > *Groupes de modèles*, les utilisateurs peuvent configurer et gérer les groupes de modèles.

Une liste des groupes de modèles existants avec leurs détails est affichée. Vous pouvez rechercher et filtrer les groupes de modèles par nom.



Données affichées :

Colonne	Description
Nom	Nom du groupe de modèles. Cliquer sur le nom du groupe ouvre le formulaire de configuration du groupe de modèles.
Modèles	Nombre de modèles dans le groupe (affiché en gris), suivi de la liste des membres du groupe. Cliquer sur le nom d'un modèle ouvre le formulaire de configuration du modèle. Cliquer sur le nombre ouvre la liste des modèles de ce groupe.

Options de modification en masse

Pour supprimer plusieurs groupes de modèles en une seule fois, cochez les cases situées devant les groupes concernés, puis cliquez sur le bouton Supprimer sous la liste.

Utilisation du filtre

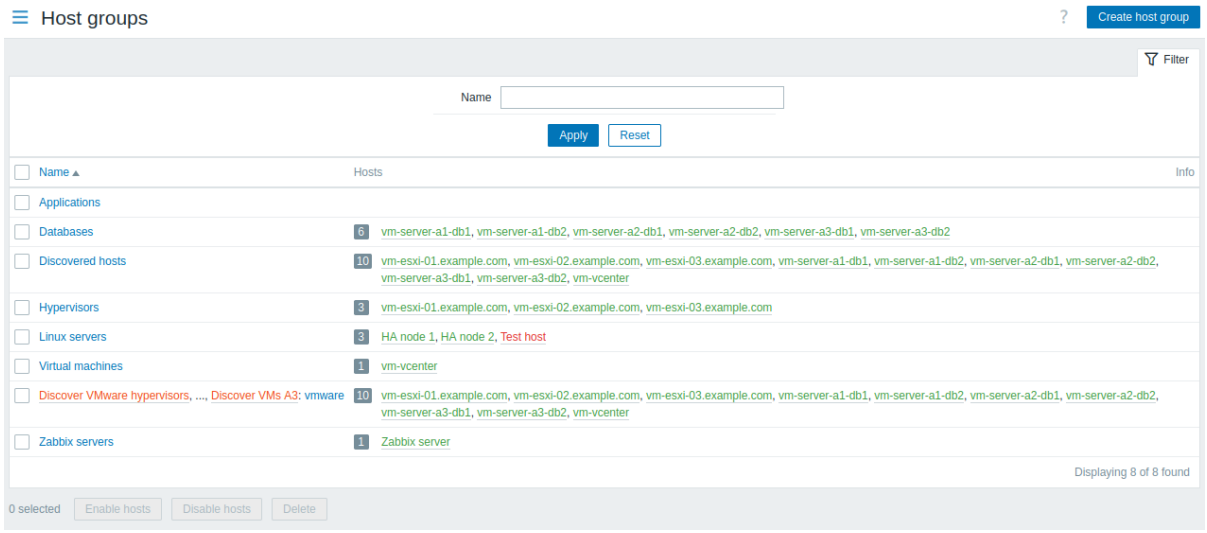
Vous pouvez utiliser le filtre pour afficher uniquement les groupes de modèles qui vous intéressent. Pour de meilleures performances de recherche, les données sont recherchées avec les macros non résolues.

Groupes d'hôtes

Vue d'ensemble

Dans la section *Collecte de données* > *Groupes d'hôtes*, les utilisateurs peuvent configurer et gérer les groupes d'hôtes.

Une liste des groupes d'hôtes existants avec leurs détails est affichée. Vous pouvez rechercher et filtrer les groupes d'hôtes par nom.



Données affichées :

Column	Description
Name	Nom du groupe d'hôtes. Cliquer sur le nom du groupe ouvre le formulaire de configuration du groupe d'hôtes. Les groupes d'hôtes découverts sont affichés avec les noms des règles de découverte de bas niveau comme préfixe. Cliquer sur le nom de la règle de découverte de bas niveau ouvre le formulaire de configuration du prototype d'hôte. Notez que les groupes d'hôtes découverts sont supprimés lorsque toutes les règles de découverte de bas niveau qui les ont découverts sont supprimées.
Hosts	Nombre d'hôtes dans le groupe (affiché en gris), suivi de la liste des membres du groupe. Cliquer sur le nom d'un hôte ouvre le formulaire de configuration de l'hôte. Cliquer sur le nombre filtre, dans la liste complète des hôtes, ceux qui appartiennent au groupe.
Info	Les informations d'erreur, le cas échéant, concernant le groupe d'hôtes sont affichées.

Options de modification en masse

Les boutons sous la liste offrent plusieurs options de modification en masse :

- *Activer les hôtes* - changer l'état de tous les hôtes du groupe en "Surveillé"
- *Désactiver les hôtes* - changer l'état de tous les hôtes du groupe en "Non surveillé"
- *Supprimer* - supprimer les groupes d'hôtes

Pour utiliser ces options, cochez les cases devant les groupes d'hôtes correspondants, puis cliquez sur le bouton requis.

Utilisation de filtes

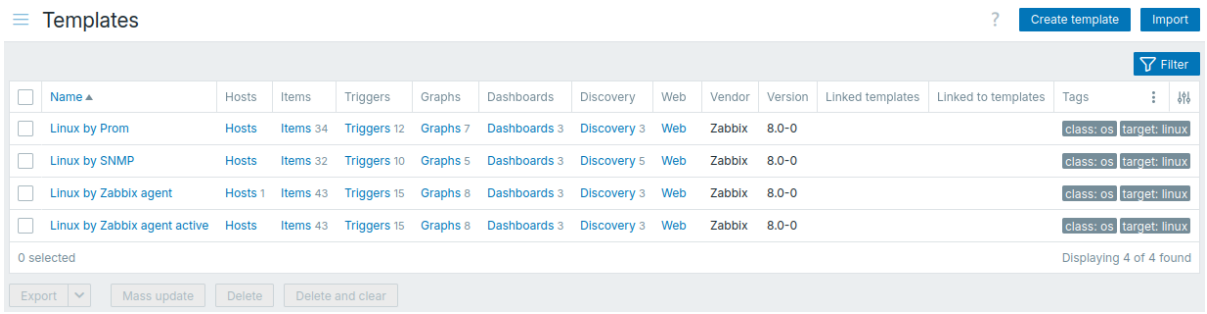
Vous pouvez utiliser le filtre pour afficher uniquement les groupes d'hôtes qui vous intéressent. Pour de meilleures performances de recherche, les données sont recherchées avec des macros non résolues.

Modèles

Présentation générale

Dans la section *Collecte de données* > *Modèles*, les utilisateurs peuvent configurer et gérer les modèles.

Une liste des modèles existants avec leurs détails est affichée.



Les détails des modèles sont affichés dans un tableau personnalisable. Le bouton *Personnaliser le tableau* dans l'en-tête du tableau permet d'afficher, de masquer ou de réorganiser les colonnes, ainsi que de rétablir la disposition par défaut. Vous pouvez également redimensionner chaque colonne en faisant glisser sa bordure droite ; double-cliquez sur la bordure pour rétablir sa taille. Toutes les personnalisations sont appliquées immédiatement et toujours mémorisées pour l'utilisateur actuel.

Les colonnes suivantes sont disponibles :

Colonne	Description
Nom	Nom du modèle. Cliquer sur le nom du modèle ouvre le formulaire de configuration du modèle.
Hôtes	Nombre d'hôtes modifiables auxquels le modèle est lié ; les hôtes en lecture seule ne sont pas inclus. Cliquer sur *Hôtes* ouvre la liste des hôtes, en filtrant uniquement ceux qui sont liés au modèle.
Entités (éléments, déclencheurs, graphiques, tableaux de bord, découverte, Web)	Nombre des entités correspondantes dans le modèle (affiché en gris). Cliquer sur le nom de l'entité filtre, dans la liste complète de cette entité, celles qui appartiennent au modèle.
Modèles liés	Modèles qui sont liés au modèle.
Lié aux modèles	Modèles auxquels le modèle est lié.
Fournisseur, *Version*	Fournisseur et version du modèle ; affichés si la configuration du modèle contient ces informations, et uniquement pour les modèles prêts à l'emploi, les modèles importés ou les modèles modifiés via l'API Template. Pour les modèles prêts à l'emploi, la version est affichée comme suit : version majeure de Zabbix, délimiteur ("-"), numéro de révision (incrémenté à chaque nouvelle version du modèle et réinitialisé à chaque version majeure de Zabbix). Par exemple : 7.0-0, 7.0-3, 8.0-0, 8.0-3.
Étiquettes	Les étiquettes du modèle sont affichées, sans résolution des macros. Placer le curseur sur l'icône à trois points ou cliquer dessus après les étiquettes affichées ouvre une fenêtre contextuelle contenant toutes les étiquettes. Cliquer sur le bouton *Options* dans l'en-tête de la colonne permet de définir le nombre d'étiquettes affichées (1 à 3), de choisir le mode d'affichage des noms d'étiquettes (complets, abrégés ou aucun), de définir la priorité d'affichage des étiquettes ou de dupliquer la colonne (les colonnes dupliquées peuvent être renommées ou supprimées).
Valeur de l'étiquette	Les valeurs des étiquettes du modèle sont affichées, sans résolution des macros. Cliquer sur le bouton *Options* dans l'en-tête de la colonne permet de renommer la colonne, de choisir les valeurs d'étiquettes à afficher ou de dupliquer la colonne (les colonnes dupliquées peuvent être renommées ou supprimées).
Texte personnalisé	Un texte personnalisé, des valeurs de macros ou une combinaison des deux sont affichés. Cliquer sur le bouton *Options* dans l'en-tête de la colonne permet de renommer la colonne, de saisir le texte personnalisé ou les macros, ou de dupliquer la colonne (les colonnes dupliquées peuvent être renommées ou supprimées). Macros prises en charge : `{HOST.ID}`, `{HOST.HOST}`, `{HOST.NAME}`, `{HOST.DESCRIPTION}` et macros utilisateur.

Pour configurer un nouveau modèle, cliquez sur le bouton *Créer un modèle* dans l'angle supérieur droit.

Pour importer un modèle depuis un fichier YAML, XML ou JSON, cliquez sur le bouton *Importer* dans l'angle supérieur droit.

Options de modification en masse

Les boutons sous la liste offrent plusieurs options de modification en masse :

- *Exporter* - exporter le modèle vers un fichier YAML, XML ou JSON ;
- *Mise à jour en masse* - mettre à jour plusieurs propriétés pour plusieurs modèles à la fois ;
- *Supprimer* - supprimer le modèle tout en laissant ses entités liées (éléments, déclencheurs, etc.) sur les hôtes ;
- *Supprimer et vider* - supprimer le modèle et ses entités liées des hôtes.

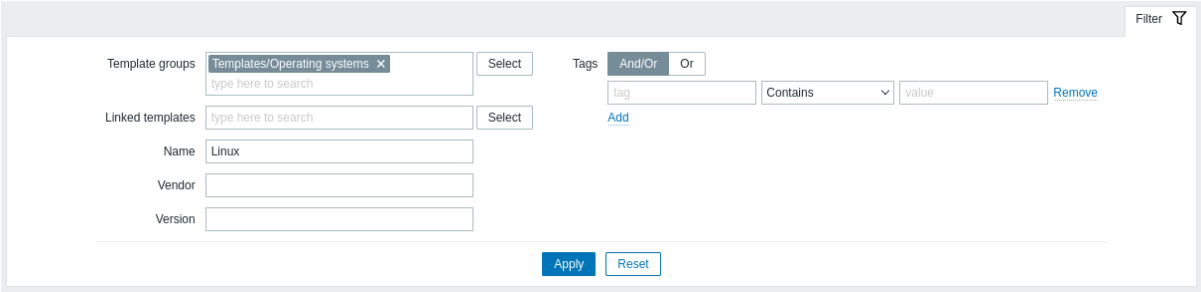
Pour utiliser ces options, cochez les cases devant les modèles concernés, puis cliquez sur le bouton requis.

Utilisation du filtre

Vous pouvez utiliser le filtre pour afficher uniquement les modèles qui vous intéressent.

Pour de meilleures performances de recherche, les données sont recherchées avec les macros non résolues.

Le lien *Filtre* est disponible sous les boutons *Créer un modèle* et *Importer*. Si vous cliquez dessus, un filtre devient disponible, où vous pouvez filtrer les modèles par groupe de modèles, modèles liés, nom et tags.



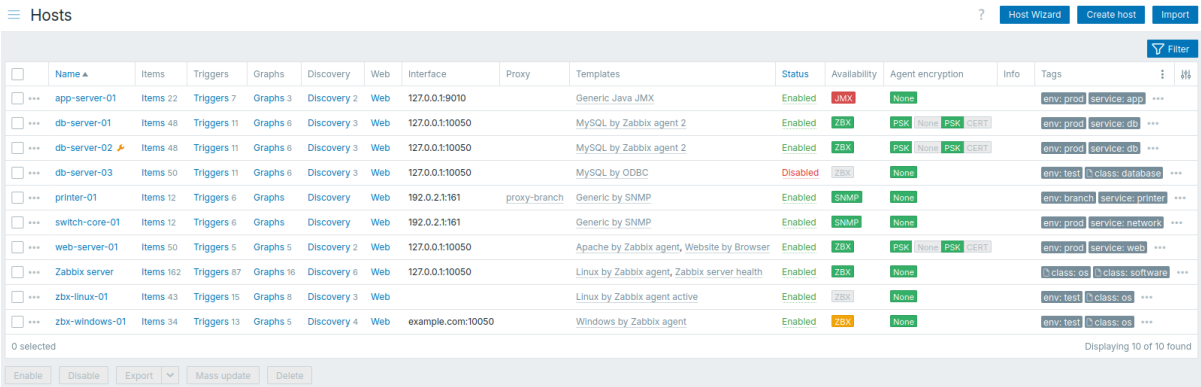
Parameter	Description
Template groups	Filtrer par un ou plusieurs groupes de modèles. La spécification d'un groupe de modèles parent sélectionne implicitement tous les groupes imbriqués.
Linked templates	Filtrer par modèles liés directement.
Name	Filtrer par nom de modèle.
Vendor	Filtrer par fournisseur du modèle.
Version	Filtrer par version du modèle.
Tags	Filtrer par nom et valeur de tag du modèle. Le filtrage est possible uniquement par les tags au niveau du modèle (et non par ceux hérités). Il est possible d'inclure ainsi que d'exclure des tags et des valeurs de tag spécifiques. Plusieurs conditions peuvent être définies. La correspondance du nom du tag est toujours sensible à la casse. Plusieurs opérateurs sont disponibles pour chaque condition : **Exists** - inclure les noms de tag spécifiés ; **Equals** - inclure les noms et valeurs de tag spécifiés (sensible à la casse) ; **Contains** - inclure les noms de tag spécifiés dont les valeurs de tag contiennent la chaîne saisie (correspondance de sous-chaîne, insensible à la casse) ; **Does not exist** - exclure les noms de tag spécifiés ; **Does not equal** - exclure les noms et valeurs de tag spécifiés (sensible à la casse) ; **Does not contain** - exclure les noms de tag spécifiés dont les valeurs de tag contiennent la chaîne saisie (correspondance de sous-chaîne, insensible à la casse). Il existe deux types de calcul pour les conditions : **And/Or** - toutes les conditions doivent être remplies, les conditions ayant le même nom de tag seront regroupées par la condition Or ; **Or** - une seule condition doit être remplie.

Hôtes

Vue d'ensemble

Dans la section *Collecte de données* > *Hôtes*, les utilisateurs peuvent configurer et gérer les hôtes.

Une liste des hôtes existants avec leurs détails est affichée.



Les détails des hôtes sont affichés dans un tableau personnalisable. Le bouton *Personnaliser le tableau* dans l'en-tête du tableau permet d'afficher, de masquer ou de réorganiser les colonnes, ainsi que de rétablir la disposition par défaut. Vous pouvez

également redimensionner chaque colonne en faisant glisser sa bordure droite ; double-cliquez sur la bordure pour rétablir sa taille. Toutes les personnalisations sont appliquées immédiatement et mémorisées pour l'utilisateur actuel.

Les colonnes suivantes sont disponibles :

Colonne	Description
Nom	Nom de l'hôte. Cliquer sur le nom de l'hôte ouvre le formulaire de configuration de l'hôte. Une icône de clé à molette orange  après le nom de l'hôte indique que cet hôte est en maintenance. Les détails de la maintenance s'affichent lorsque le pointeur de la souris est placé sur l'icône.
Entités (Éléments, Déclencheurs, Graphiques, Découverte, Web)	Cliquer sur le nom de l'entité affiche les éléments, déclencheurs, etc. de l'hôte. Le nombre des entités correspondantes est affiché en gris.
Interface	L'interface principale de l'hôte est affichée.
Proxy	Les proxy attribués sont affichés dans cette colonne : <Nom du proxy> - l'hôte est surveillé par un proxy autonome (même si le proxy fait partie d'un groupe de proxy) ; <Nom du groupe de proxy : nom du proxy> - l'hôte est surveillé par un groupe de proxy et le serveur Zabbix a attribué un proxy pour surveiller l'hôte ; <Nom du groupe de proxy> - l'hôte est surveillé par un groupe de proxy sans proxy, ou le serveur Zabbix n'a pas attribué de proxy pour surveiller l'hôte ; Rien - l'hôte n'est surveillé ni par un proxy ni par un groupe de proxy. Cette colonne s'affiche uniquement si l'option de filtre *Surveillé par* est définie sur « N'importe lequel », « Proxy » ou « Groupe de proxy ».
Modèles	Les modèles liés à l'hôte sont affichés. Si d'autres modèles sont contenus dans le modèle lié, ils sont affichés entre parenthèses et séparés par une virgule. Cliquer sur le nom d'un modèle ouvre son formulaire de configuration.
État	L'état de l'hôte est affiché : *Activé* ou *Désactivé*. Cliquer sur l'état permet de le modifier manuellement. Les hôtes découverts qui ont été perdus sont marqués d'une icône d'information. Le texte de l'infobulle fournit des détails sur leur état.

Colonne	Description
Disponibilité	La disponibilité de l'hôte pour chaque interface configurée est affichée. Les icônes de disponibilité représentent l'état actuel de l'interface de l'hôte sur le serveur Zabbix. Par conséquent, si vous désactivez un hôte dans l'interface, sa disponibilité sera mise à jour après que le serveur Zabbix aura synchronisé les modifications de configuration. De même, si vous activez un hôte, sa disponibilité sera mise à jour après que le serveur Zabbix aura synchronisé les modifications de configuration et interrogé l'hôte. Les icônes de disponibilité représentent uniquement les types d'interface (Agent, SNMP, IPMI, JMX) qui sont configurés. Le survol de l'icône affiche une fenêtre contextuelle contenant la liste de toutes les interfaces du même type, avec leurs détails, leur état et leurs erreurs. Pour l'interface Agent, la fenêtre contextuelle affiche les interfaces (passives) et les vérifications actives. Si un hôte possède uniquement des vérifications actives, l'icône de l'interface Agent est affichée même si aucune interface Agent n'est configurée pour l'hôte. La colonne est vide pour les hôtes sans interface. L'état d'une interface d'hôte individuelle est déterminé par la connexion entre un élément activé utilisant l'interface et l'hôte. L'état peut être : **Disponible** - la connexion à l'hôte a réussi ; **Non disponible** - la connexion à l'hôte a échoué (délai d'attente, problèmes de pare-feu, etc.) ; **Inconnu** - la connexion à l'hôte n'a pas été tentée ou le résultat est inconnu. Pour plus de détails sur la manière dont le serveur Zabbix détermine l'état de l'interface, consultez État d'interface inconnu et Paramètres des interfaces d'hôte inaccessibles/non disponibles. L'état de toutes les interfaces d'hôte d'un même type (Agent, SNMP, IPMI, JMX) est déterminé par les interfaces utilisées par au moins un élément activé. L'état est indiqué par la couleur de l'icône : **Vert** - toutes les interfaces sont disponibles ; **Jaune** - au moins une interface n'est pas disponible et au moins une autre est disponible ou inconnue ; **Rouge** - aucune interface n'est disponible ; **Gris** - au moins une interface est inconnue, mais aucune n'est indisponible. **Disponibilité des vérifications actives.** Si au moins une vérification active est activée sur l'hôte, les vérifications actives influencent également la disponibilité totale de l'interface Agent, comme décrit ci-dessus. Pour déterminer la disponibilité des vérifications actives, des messages de pulsation sont envoyés dans le thread des vérifications actives de l'agent. La fréquence des messages de pulsation est contrôlée par le paramètre `HeartbeatFrequency` dans la configuration de l'agent ou de l'agent 2 Zabbix (60 secondes par défaut, plage de 0 à 3600). Les vérifications actives sont considérées comme non disponibles lorsque la pulsation de la vérification active date de plus de 2 x HeartbeatFrequency secondes. **Remarque :** les agents Zabbix antérieurs à la version 6.2.x n'envoient pas de pulsations pour les vérifications actives ; la disponibilité de leurs hôtes reste donc inconnue.
Chiffrement de l'agent	L'état du chiffrement des connexions vers l'hôte et depuis l'hôte est affiché : **Aucun** - aucun chiffrement ; **PSK** - utilisation d'une clé prépartagée ; **Cert** - utilisation d'un certificat.
Informations	Les informations d'erreur éventuelles concernant l'hôte sont affichées.
Étiquettes	Les étiquettes de l'hôte sont affichées, sans résolution des macros. Le survol ou le clic sur l'icône représentant trois points ••• après les étiquettes affichées ouvre une fenêtre contextuelle contenant toutes les étiquettes. Cliquer sur le bouton ⋮ *Options* dans l'en-tête de la colonne permet de définir le nombre d'étiquettes affichées (1 à 3), de choisir le mode d'affichage des noms d'étiquettes (complets, abrégés ou aucun), de définir la priorité d'affichage des étiquettes ou de dupliquer la colonne (les colonnes dupliquées peuvent être renommées ou supprimées).
Valeur de l'étiquette	Les valeurs des étiquettes de l'hôte sont affichées, sans résolution des macros. Cliquer sur le bouton ⋮ *Options* dans l'en-tête de la colonne permet de renommer la colonne, de choisir les valeurs d'étiquettes à afficher ou de dupliquer la colonne (les colonnes dupliquées peuvent être renommées ou supprimées).

Colonne	Description
Texte personnalisé	Le texte personnalisé, les valeurs de macros ou une combinaison des deux sont affichés. Cliquer sur le bouton *Options* dans l'en-tête de la colonne permet de renommer la colonne, de saisir le texte personnalisé ou les macros, ou de dupliquer la colonne (les colonnes dupliquées peuvent être renommées ou supprimées). Macros prises en charge : `{HOST.*}` (à l'exception de `{HOST.METADATA}`), `{INVENTORY.*}` et macros utilisateur.

Pour ouvrir le menu de l'hôte, cliquez sur le bouton *Options* situé avant le nom de l'hôte.

Pour configurer un nouvel hôte, cliquez sur le bouton *Assistant de l'hôte* ou *Créer un hôte* dans l'angle supérieur droit.

Pour importer un hôte depuis un fichier YAML, XML ou JSON, cliquez sur le bouton *Importer* dans l'angle supérieur droit.

Options de modification en masse

Les boutons sous la liste offrent plusieurs options de modification en masse :

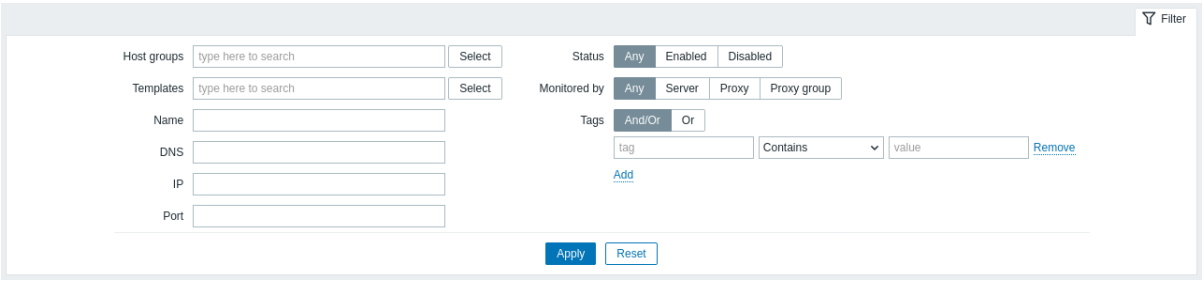
- *Enable* - changer l'état de l'hôte en *Monitored* ;
- *Disable* - changer l'état de l'hôte en *Not monitored* ;
- *Export* - exporter les hôtes vers un fichier YAML, XML ou JSON ;
- *Mass update* - mettre à jour plusieurs propriétés pour plusieurs hôtes à la fois ;
- *Delete* - supprimer les hôtes.

Pour utiliser ces options, cochez les cases devant les hôtes concernés, puis cliquez sur le bouton requis.

Utilisation du filtre

Vous pouvez utiliser le filtre pour afficher uniquement les hôtes qui vous intéressent. Pour de meilleures performances de recherche, les données sont recherchées avec les macros non résolues.

L'icône *Filter* est disponible dans le coin supérieur droit. En cliquant dessus, vous ouvrirez un filtre dans lequel vous pouvez spécifier les critères de filtrage souhaités.



Parameter	Description
Host groups	Filtrer par un ou plusieurs groupes d'hôtes. La spécification d'un groupe d'hôtes parent sélectionne implicitement tous les groupes d'hôtes imbriqués.
Templates	Filtrer par les modèles liés.
Name	Filtrer par le nom d'hôte visible.
DNS	Filtrer par le nom DNS.
IP	Filtrer par adresse IP.
Port	Filtrer par numéro de port.
Status	Filtrer par statut de l'hôte.
Monitored by	Filtrer les hôtes surveillés par le serveur Zabbix, le proxy ou le groupe de proxy.
Proxies	Filtrer les hôtes surveillés par les proxies spécifiés ici. Ce champ n'est disponible que si "Proxy" est sélectionné dans le champ *Monitored by*.
Proxy groups	Filtrer les hôtes surveillés par les groupes de proxy spécifiés ici. Ce champ n'est disponible que si "Proxy group" est sélectionné dans le champ *Monitored by*.

Parameter	Description
Tags	Filtrer par nom et valeur de tag d'hôte. Il est possible d'inclure ainsi que d'exclure des tags et des valeurs de tag spécifiques. Plusieurs conditions peuvent être définies. La correspondance du nom du tag est toujours sensible à la casse. Plusieurs opérateurs sont disponibles pour chaque condition : **Exists** - inclure les noms de tag spécifiés ; **Equals** - inclure les noms et valeurs de tag spécifiés (sensible à la casse) ; **Contains** - inclure les noms de tag spécifiés dont les valeurs de tag contiennent la chaîne saisie (correspondance de sous-chaîne, insensible à la casse) ; **Does not exist** - exclure les noms de tag spécifiés ; **Does not equal** - exclure les noms et valeurs de tag spécifiés (sensible à la casse) ; **Does not contain** - exclure les noms de tag spécifiés dont les valeurs de tag contiennent la chaîne saisie (correspondance de sous-chaîne, insensible à la casse). Il existe deux types de calcul pour les conditions : **And/Or** - toutes les conditions doivent être remplies, les conditions ayant le même nom de tag seront regroupées par la condition Or ; **Or** - il suffit qu'une condition soit remplie.

Statut d'interface inconnu

Le serveur Zabbix affiche le statut "Unknown" pour une interface d'hôte (Agent, SNMP, IPMI, JMX) dans les cas suivants :

- L'hôte est désactivé.
- L'hôte est configuré pour être surveillé par un proxy, un proxy différent, ou le serveur s'il était précédemment surveillé par un proxy.
- L'hôte est surveillé par un proxy qui semble hors ligne (aucune mise à jour reçue du proxy pendant l'intervalle maximal de battement de coeur - 1 heure).
- Tous les éléments de l'hôte avec ce type d'interface sont désactivés.
- Aucun poller pour ce type d'interface n'a été configuré (par exemple, le paramètre de configuration du serveur `StartAgentPollers` ou `StartSNMPPollers` est défini sur 0).

La disponibilité de l'interface est définie sur "Unknown" après la synchronisation du cache de configuration du serveur Zabbix.

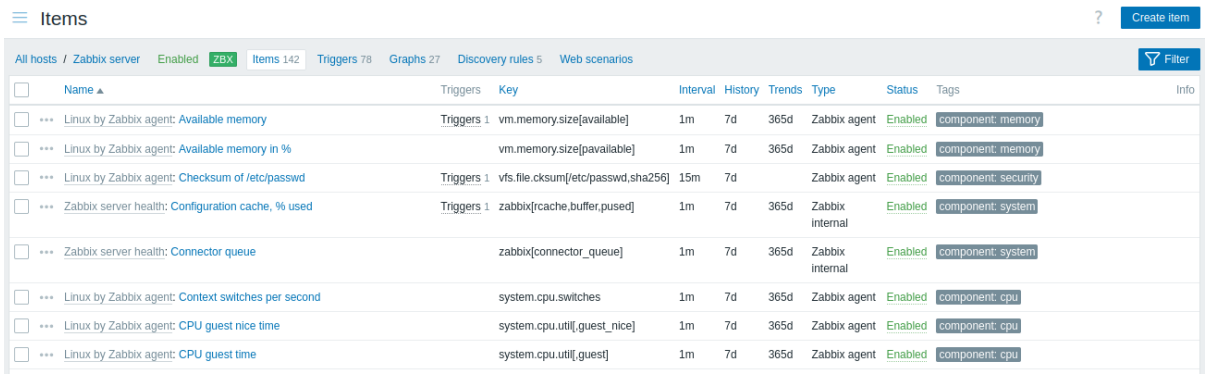
La disponibilité de l'interface (Available/Not available) sur les hôtes surveillés par des proxies est rétablie après la synchronisation du cache de configuration du proxy.

Éléments

Vue d'ensemble

La liste des éléments d'un hôte est accessible depuis *Collecte de données* > *Hôtes* en cliquant sur *Éléments* pour l'hôte concerné.

Une liste des éléments existants s'affiche.



Données affichées :

Colonne	Description
Menu de l'élément	Cliquez sur l'icône à trois points pour ouvrir le menu de l'élément.
Hôte	Hôte de l'élément. Cliquer sur le nom de l'hôte ouvre le formulaire de configuration de l'hôte. Cette colonne s'affiche uniquement si plusieurs hôtes ou aucun hôte sont sélectionnés dans le filtre.
Nom	Nom de l'élément affiché sous forme de lien bleu vers les détails de l'élément. Cliquer sur le lien du nom de l'élément ouvre le formulaire de configuration de l'élément. Si l'élément de l'hôte appartient à un modèle, le nom du modèle est affiché avant le nom de l'élément sous forme de lien gris. Cliquer sur le lien du modèle ouvre la liste des éléments au niveau du modèle. Si l'élément a été créé à partir d'un prototype d'élément, son nom est précédé du nom de la règle de découverte de bas niveau, en orange. Cliquer sur le nom de la règle de découverte ouvre la liste des prototypes d'éléments.
Déclencheurs	Placer le pointeur de la souris sur Déclencheurs affiche une infobulle contenant les déclencheurs associés à l'élément. Le nombre de déclencheurs est affiché en gris.
Clé	La clé de l'élément est affichée.
Intervalle	La fréquence de la vérification est affichée. *Remarque* : les éléments passifs peuvent également être vérifiés immédiatement en cliquant sur le bouton *Exécuter maintenant*.
Historique	Le nombre de jours pendant lesquels l'historique des données de l'élément sera conservé est affiché.
Tendances	Le nombre de jours pendant lesquels l'historique des tendances de l'élément sera conservé est affiché.
Type	Le type d'élément est affiché (agent Zabbix, agent SNMP, vérification simple, etc.).
État	L'état de l'élément est affiché : *Activé*, *Désactivé* ou *Non pris en charge*. Vous pouvez modifier manuellement l'état en cliquant dessus : d'Activé à Désactivé (et inversement), ou de Non pris en charge à Désactivé (et inversement). Les éléments découverts qui ont été perdus sont marqués d'une icône d'information. Le texte de l'infobulle fournit des détails sur leur état.
Étiquettes	Les étiquettes de l'élément sont affichées. Jusqu'à trois étiquettes (paires nom:valeur) peuvent être affichées. S'il y a davantage d'étiquettes, un lien « ... » s'affiche et permet de consulter toutes les étiquettes en plaçant le pointeur de la souris dessus.
Informations	Si l'élément fonctionne correctement, aucune icône ne s'affiche dans cette colonne. En cas d'erreur, une icône carrée contenant la lettre « i » s'affiche. Placez le pointeur sur l'icône pour afficher une infobulle contenant la description de l'erreur.

Pour configurer un nouvel élément, cliquez sur le bouton *Créer un élément* dans l'angle supérieur droit.

Options de modification en masse

Les boutons sous la liste offrent plusieurs options de modification en masse :

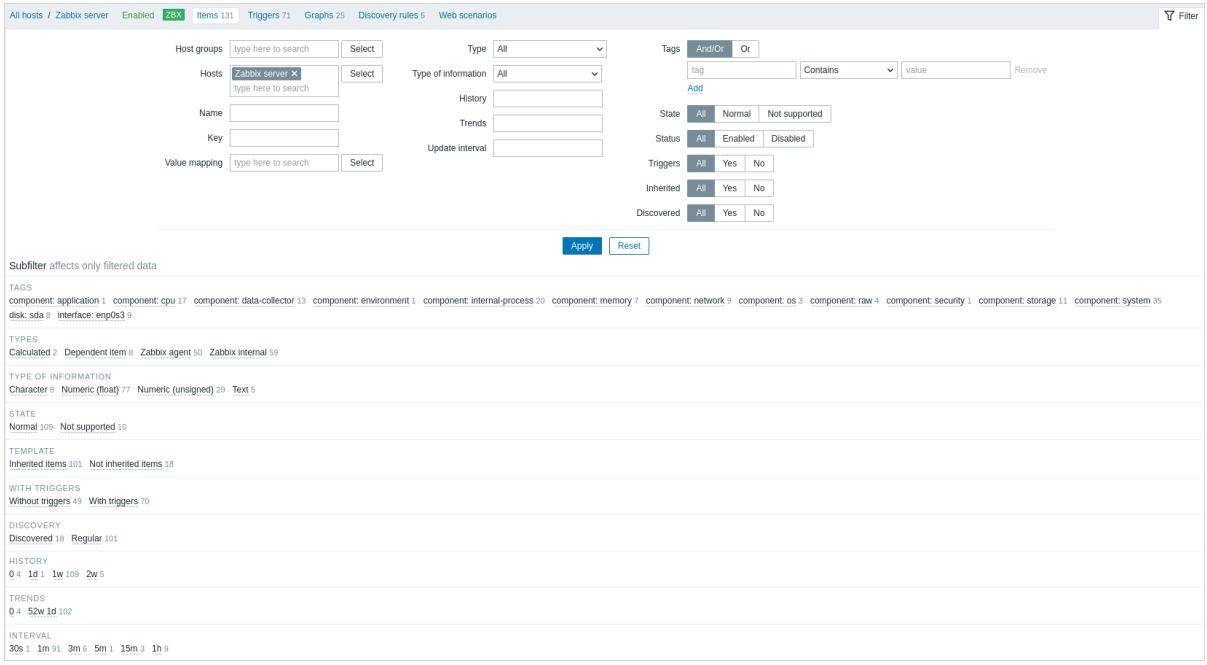
- *Enable* - changer l'état de l'élément en *Enabled*
- *Disable* - changer l'état de l'élément en *Disabled*
- *Execute now* - exécuter immédiatement une vérification pour les nouvelles valeurs de l'élément. Pris en charge uniquement pour les vérifications **passives** (voir plus de détails). Notez que lors de la vérification immédiate des valeurs, le cache de configuration n'est pas mis à jour ; les valeurs ne refléteront donc pas les changements très récents apportés à la configuration de l'élément.
- *Clear history and trends* - supprimer les données d'historique et de tendances des éléments.
- *Copy* - copier les éléments vers d'autres hôtes ou modèles.
- *Mass update* - mettre à jour plusieurs propriétés pour un certain nombre d'éléments en une seule fois.
- *Delete* - supprimer les éléments.

Pour utiliser ces options, cochez les cases devant les éléments concernés, puis cliquez sur le bouton requis.

Utilisation du filtre

Vous pouvez utiliser le filtre pour afficher uniquement les éléments qui vous intéressent. Pour de meilleures performances de recherche, les données sont recherchées avec les macros non résolues.

L'icône *Filter* est disponible dans le coin supérieur droit. En cliquant dessus, vous ouvrirez un filtre dans lequel vous pourrez spécifier les critères de filtrage souhaités.

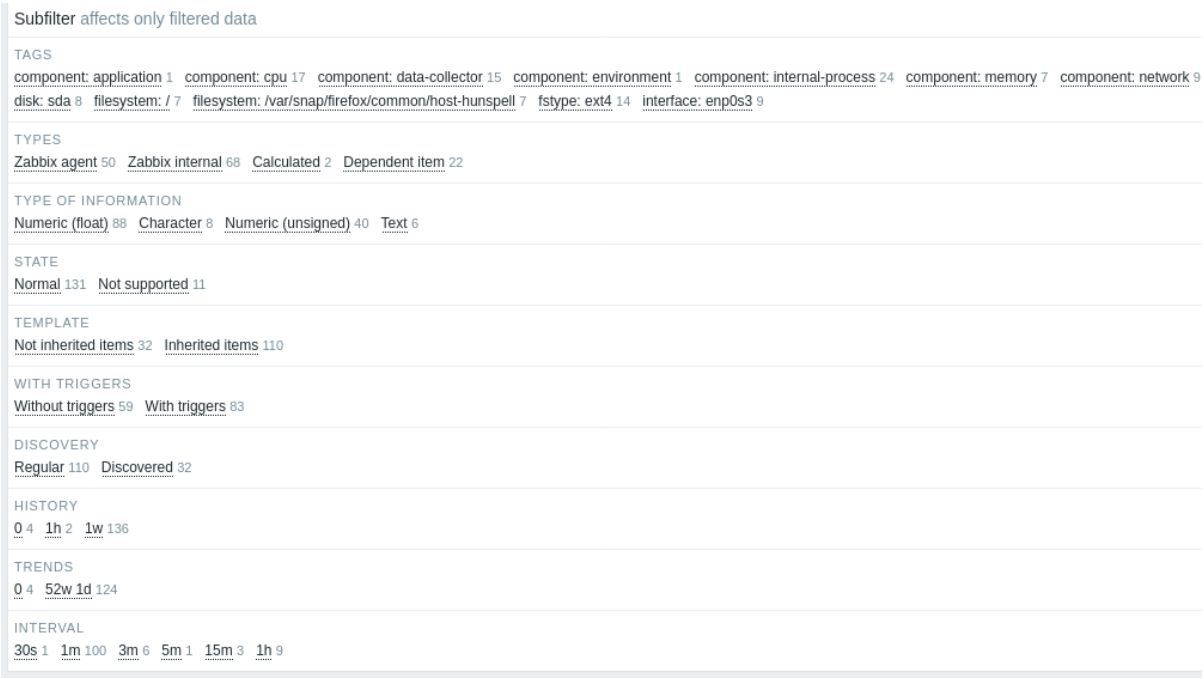


Parameter	Description
Host groups	Filtrer par un ou plusieurs groupes d'hôtes. La spécification d'un groupe d'hôtes parent sélectionne implicitement tous les groupes d'hôtes imbriqués. Les groupes d'hôtes contenant uniquement des modèles ne peuvent pas être sélectionnés.
Hosts	Filtrer par un ou plusieurs hôtes.
Name	Filtrer par le nom de l'élément.
Key	Filtrer par la clé de l'élément.
Value mapping	Filtrer par la correspondance de valeurs utilisée. Ce paramètre n'est pas affiché si l'option *Hosts* est vide.
Type	Filtrer par type d'élément (Zabbix agent, agent SNMP, etc.).
Type of information	Filtrer par type d'information (entier non signé, flottant, etc.).
History	Filtrer par la durée de conservation de l'historique de l'élément.
Trends	Filtrer par la durée de conservation des tendances de l'élément.
Update interval	Filtrer par l'intervalle de mise à jour de l'élément.
Tags	Spécifiez des tags pour limiter le nombre d'éléments affichés. Il est possible d'inclure ou d'exclure des tags et des valeurs de tags spécifiques. Plusieurs conditions peuvent être définies. La correspondance du nom du tag est toujours sensible à la casse. Plusieurs opérateurs sont disponibles pour chaque condition : **Exists** - inclure les noms de tags spécifiés **Equals** - inclure les noms et valeurs de tags spécifiés (sensible à la casse) **Contains** - inclure les noms de tags spécifiés lorsque les valeurs de tag contiennent la chaîne saisie (correspondance de sous-chaîne, insensible à la casse) **Does not exist** - exclure les noms de tags spécifiés **Does not equal** - exclure les noms et valeurs de tags spécifiés (sensible à la casse) **Does not contain** - exclure les noms de tags spécifiés lorsque les valeurs de tag contiennent la chaîne saisie (correspondance de sous-chaîne, insensible à la casse) Il existe deux types de calcul pour les conditions : **And/Or** - toutes les conditions doivent être remplies, les conditions ayant le même nom de tag seront regroupées par la condition Or **Or** - il suffit qu'une condition soit remplie
State	Filtrer par l'état de l'élément - *Normal* ou *Not supported*.
Status	Filtrer par le statut de l'élément - *Enabled* ou *Disabled*.
Triggers	Filtrer les éléments avec (ou sans) déclencheurs.
Inherited	Filtrer les éléments hérités (ou non hérités) d'un modèle.
Discovery	Filtrer les éléments découverts (ou non découverts) par la découverte de bas niveau.

Utilisation du sous-filtre

Le sous-filtre permet de modifier davantage le filtrage appliqué par le filtre principal.

Il contient des liens cliquables permettant d'accéder rapidement aux éléments associés. Les éléments sont associés par une entité commune : balise, type d'élément, état de l'élément, intervalle de mise à jour de l'élément, etc. Lorsqu'une entité est sélectionnée, elle est mise en évidence par un arrière-plan gris et les éléments sont immédiatement filtrés (il n'est pas nécessaire de cliquer sur *Appliquer* dans le filtre principal). La sélection d'une autre entité l'ajoute aux résultats filtrés. La sélection répétée de l'entité supprime le filtrage.



Les sous-filtres sont générés à partir des données filtrées, limitées à 1000 enregistrements. Pour afficher davantage d'enregistrements dans le sous-filtre, vous devez augmenter la valeur du paramètre *Limite des résultats de recherche et de filtrage* (dans *Administration* > *Général* > *Interface*).

Contrairement au filtre principal, le sous-filtre est mis à jour à chaque demande d'actualisation du tableau afin de toujours afficher des informations à jour sur les options de filtrage disponibles et leurs nombres.

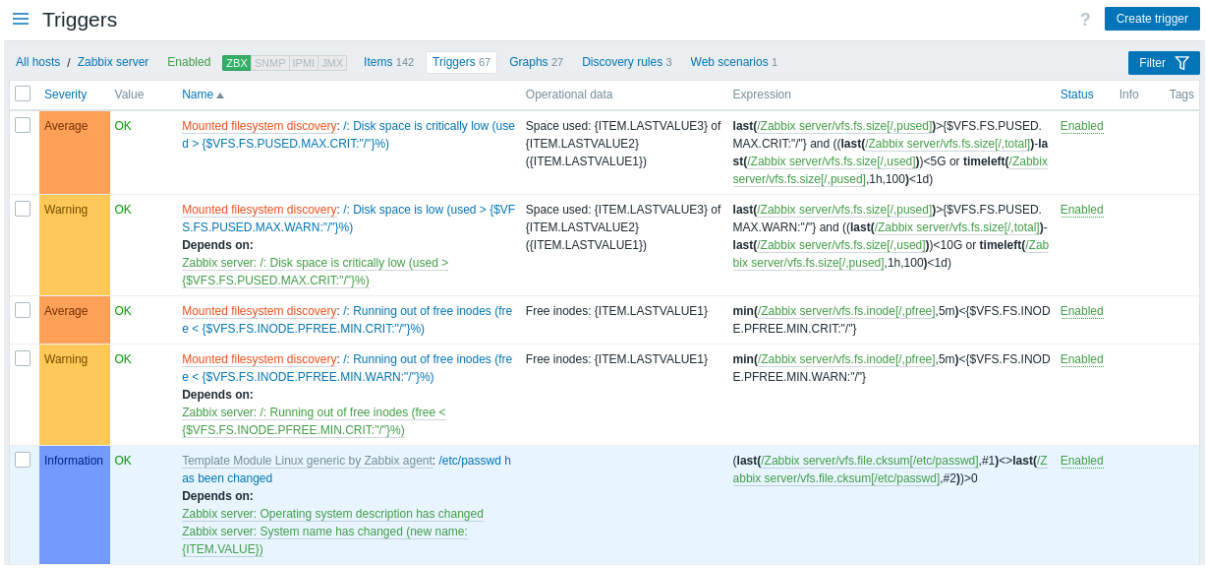
Le nombre d'entités affichées est limité à 100 horizontalement. S'il y en a davantage, une icône représentant trois points est affichée à la fin ; elle n'est pas cliquable.

Un nombre placé à côté de chaque entité cliquable indique le nombre d'éléments qui y sont regroupés (en fonction des résultats du filtre principal). Lorsqu'une entité est sélectionnée, les nombres associés aux autres entités disponibles sont affichés avec un signe plus indiquant combien d'éléments peuvent être ajoutés à la sélection actuelle. Les entités ne contenant aucun élément ne sont pas affichées, sauf si elles ont été sélectionnées auparavant dans le sous-filtre.

Déclencheurs

Présentation

La liste des déclencheurs d'un hôte est accessible depuis *Collecte de données* > *Hôtes* en cliquant sur *Déclencheurs* pour l'hôte concerné.



Données affichées :

Colonne	Description
Gravité	La gravité du déclencheur est affichée à la fois par son nom et par la couleur d'arrière-plan de la cellule.
Valeur	L'état du déclencheur est affiché : **OK** - état OK **PROBLEM** - état Problème
Hôte	Hôte du déclencheur. Cliquer sur le nom de l'hôte ouvre le formulaire de configuration de l'hôte. Cette colonne est affichée uniquement si plusieurs hôtes ou aucun hôte ne sont sélectionnés dans le filtre.
Nom	Nom du déclencheur, affiché sous forme de lien bleu vers les détails du déclencheur. Cliquer sur le lien du nom du déclencheur ouvre le formulaire de configuration du déclencheur. Si le déclencheur de l'hôte appartient à un modèle, le nom du modèle est affiché avant le nom du déclencheur sous forme de lien gris. Cliquer sur le lien du modèle ouvre la liste des déclencheurs au niveau du modèle. Si le déclencheur a été créé à partir d'un prototype de déclencheur, son nom est précédé du nom de la règle de découverte de bas niveau, en orange. Cliquer sur le nom de la règle de découverte ouvre la liste des prototypes de déclencheurs.
Données opérationnelles	Définition des données opérationnelles du déclencheur, contenant des chaînes arbitraires et des macros qui seront résolues dynamiquement dans *Supervision* > *Problèmes*.
Expression	L'expression du déclencheur est affichée. La partie hôte-élément de l'expression est affichée sous forme de lien menant au formulaire de configuration de l'élément.
État	L'état du déclencheur est affiché : *Activé*, *Désactivé* ou *Inconnu*. Cliquer sur l'état permet de le modifier manuellement : d'Activé à Désactivé (et inversement) ; d'Inconnu à Désactivé (et inversement). Les problèmes d'un déclencheur désactivé ne sont plus affichés dans l'interface, mais ne sont pas supprimés. Les déclencheurs découverts qui ont été perdus sont marqués d'une icône d'information. Le texte de l'infobulle fournit des détails sur leur état.
Informations	Si tout fonctionne correctement, aucune icône n'est affichée dans cette colonne. En cas d'erreur, une icône carrée portant la lettre « i » est affichée. Survolez l'icône pour afficher une infobulle contenant la description de l'erreur.
Étiquettes	Si un déclencheur contient des étiquettes, leur nom et leur valeur sont affichés dans cette colonne.

Pour configurer un nouveau déclencheur, cliquez sur le bouton *Créer un déclencheur* dans l'angle supérieur droit.

Options de modification en masse

Les boutons sous la liste offrent plusieurs options de modification en masse :

- *Activer* - changer l'état du déclencheur en *Activé*.
- *Désactiver* - changer l'état du déclencheur en *Désactivé*.

- *Copier* - copier les déclencheurs vers d'autres hôtes ou modèles.
- *Mise à jour en masse* - mettre à jour plusieurs propriétés pour un certain nombre de déclencheurs en une seule fois.
- *Supprimer* - supprimer les déclencheurs.

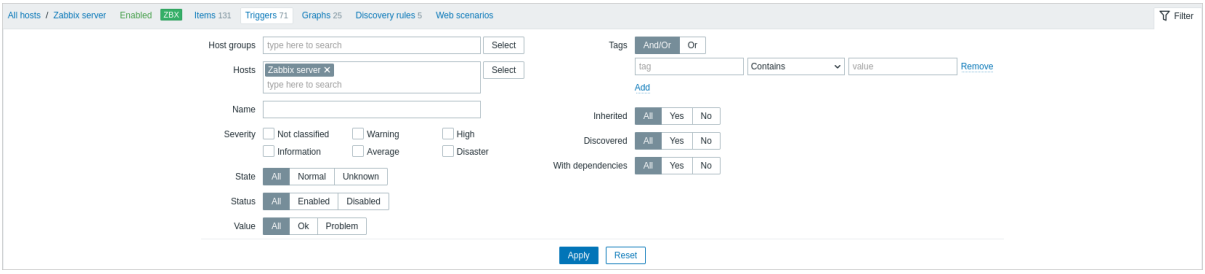
Pour utiliser ces options, cochez les cases devant les déclencheurs concernés, puis cliquez sur le bouton requis.

Utilisation du filtre

Vous pouvez utiliser le filtre pour afficher uniquement les déclencheurs qui vous intéressent.

Pour de meilleures performances de recherche, les données sont recherchées avec les macros non résolues.

L'icône *Filtre* est disponible dans le coin supérieur droit. En cliquant dessus, vous ouvrirez un filtre dans lequel vous pouvez spécifier les critères de filtrage souhaités.



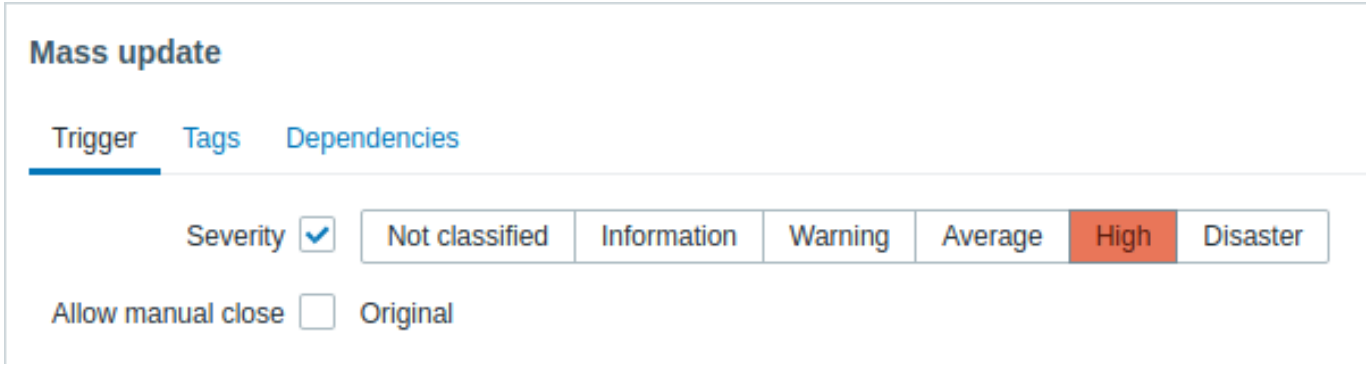
Parameter	Description
Host groups	Filtrer par un ou plusieurs groupes d'hôtes. La spécification d'un groupe d'hôtes parent sélectionne implicitement tous les groupes d'hôtes imbriqués. Les groupes d'hôtes contenant uniquement des modèles ne peuvent pas être sélectionnés.
Hosts	Filtrer par un ou plusieurs hôtes. Si des groupes d'hôtes sont déjà sélectionnés ci-dessus, la sélection des hôtes est limitée à ces groupes.
Name	Filtrer par nom de déclencheur.
Severity	Sélectionnez cette option pour filtrer par une ou plusieurs sévérités de déclencheur.
State	Filtrer par état d'expression du déclencheur (Normal/Inconnu).
Status	Filtrer par état d'activation du déclencheur (Activé/Désactivé).
Value	Filtrer par valeur du déclencheur.
Tags	Filtrer par nom et valeur de tag du déclencheur. Il est possible d'inclure ainsi que d'exclure des tags et des valeurs de tag spécifiques. Plusieurs conditions peuvent être définies. La correspondance du nom du tag est toujours sensible à la casse. Plusieurs opérateurs sont disponibles pour chaque condition : **Exists** - inclure les noms de tag spécifiés **Equals** - inclure les noms et valeurs de tag spécifiés (sensible à la casse) **Contains** - inclure les noms de tag spécifiés dont les valeurs de tag contiennent la chaîne saisie (correspondance de sous-chaîne, insensible à la casse) **Does not exist** - exclure les noms de tag spécifiés **Does not equal** - exclure les noms et valeurs de tag spécifiés (sensible à la casse) **Does not contain** - exclure les noms de tag spécifiés dont les valeurs de tag contiennent la chaîne saisie (correspondance de sous-chaîne, insensible à la casse) Il existe deux types de calcul pour les conditions : **And/Or** - toutes les conditions doivent être remplies, les conditions ayant le même nom de tag seront regroupées par la condition Or **Or** - une seule condition doit être remplie Les macros et les fonctions de macro sont prises en charge à la fois dans les champs du nom de tag et de la valeur de tag.
Inherited	Filtrer les déclencheurs hérités (ou non hérités) d'un modèle.
Discovered	Filtrer les déclencheurs découverts (ou non découverts) par la découverte de bas niveau.
With dependencies	Filtrer les déclencheurs avec dépendances (ou sans dépendances).

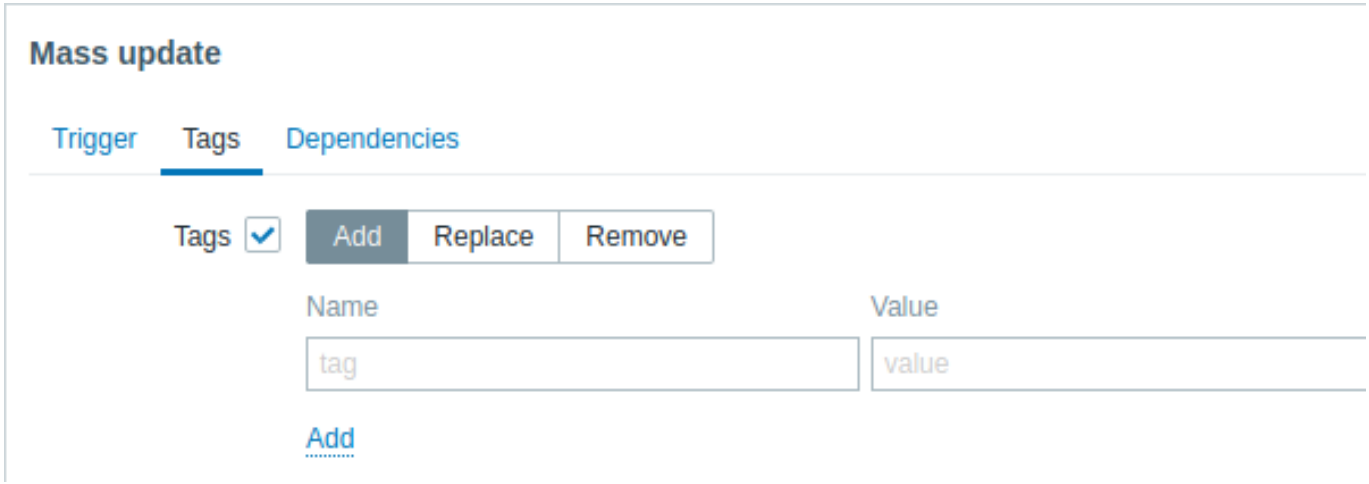
Utilisation de la mise à jour en masse

La mise à jour en masse vous permet de modifier certains attributs de plusieurs déclencheurs à la fois, sans avoir à ouvrir chaque déclencheur individuellement pour le modifier.

Pour mettre à jour plusieurs déclencheurs en masse, procédez comme suit :

- Cochez les cases des déclencheurs que vous souhaitez mettre à jour dans la liste
- Cliquez sur *Mise à jour en masse* sous la liste
- Accédez à l'onglet contenant les attributs requis (*Déclencheur*, *Étiquettes* ou *Dépendances*)
- Cochez les cases des attributs à mettre à jour

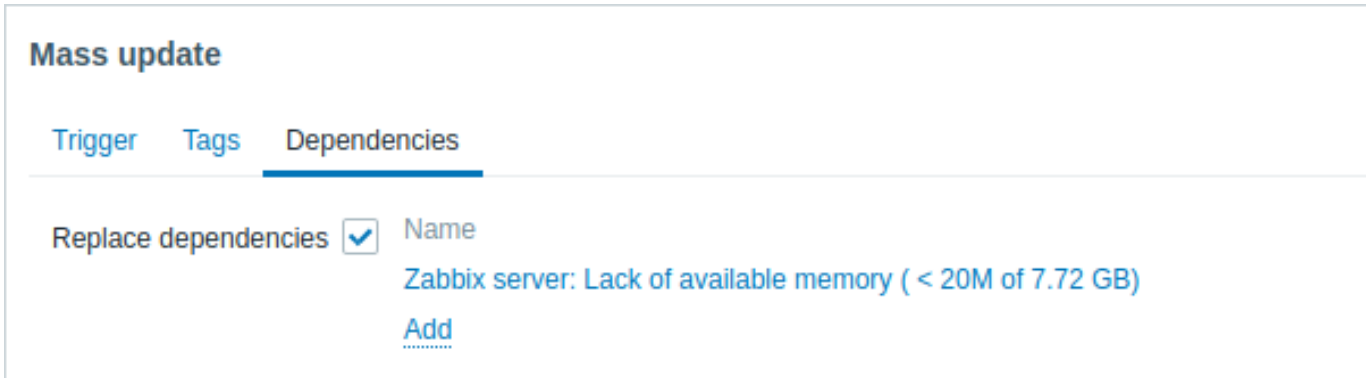




Les options suivantes sont disponibles lors de la sélection du bouton correspondant pour la mise à jour des étiquettes :

- *Ajouter* - ajouter de nouvelles étiquettes aux déclencheurs ;
- *Remplacer* - supprimer toutes les étiquettes existantes du déclencheur et les remplacer par celle(s) indiquée(s) ci-dessous ;
- *Supprimer* - supprimer les étiquettes indiquées des déclencheurs.

Notez que les étiquettes portant le même nom mais ayant des valeurs différentes ne sont pas considérées comme des « doublons » et peuvent être ajoutées au même déclencheur.



Remplacer les dépendances - supprime toutes les dépendances existantes du déclencheur et les remplace par celle(s) indiquée(s).

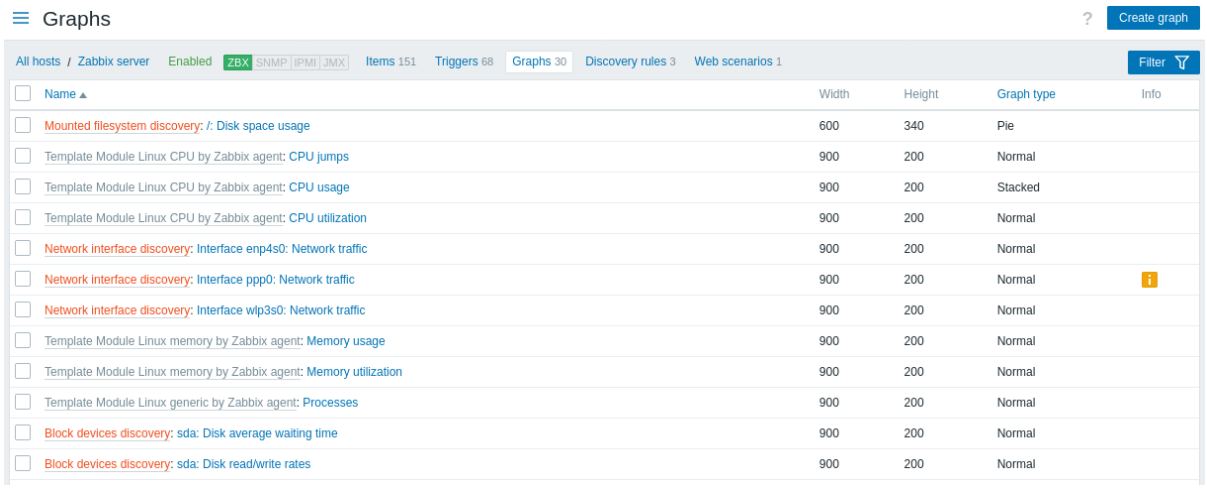
Cliquez sur *Mettre à jour* pour appliquer les modifications.

Graphiques

Vue d'ensemble

La liste des graphiques personnalisés d'un hôte est accessible depuis *Collecte de données* > *Hôtes* en cliquant sur *Graphiques* pour l'hôte concerné.

Une liste des graphiques existants s'affiche.



Données affichées :

Colonne	Description
Hôte	Hôte du graphique. Cliquer sur le nom de l'hôte ouvre le formulaire de configuration de l'hôte. Cette colonne s'affiche uniquement si plusieurs hôtes ou aucun hôte n'est sélectionné dans le filtre.
Nom	Nom du graphique personnalisé, affiché sous forme de lien bleu vers les détails du graphique. Cliquer sur le lien du nom du graphique ouvre le formulaire de configuration du graphique. Si le graphique de l'hôte appartient à un modèle, le nom du modèle est affiché avant le nom du graphique sous forme de lien gris. Cliquer sur le lien du modèle ouvre la liste des graphiques au niveau du modèle. Si le graphique a été créé à partir d'un prototype de graphique, son nom est précédé du nom de la règle de découverte de bas niveau, en orange. Cliquer sur le nom de la règle de découverte ouvre la liste des prototypes de graphiques.
Largeur	La largeur du graphique est affichée.
Hauteur	La hauteur du graphique est affichée.
Type de graphique	Le type de graphique est affiché : *Normal*, *Empilé*, *Camembert* ou *Explosé*.
Informations	Si le graphique fonctionne correctement, aucune icône ne s'affiche dans cette colonne. En cas d'erreur, une icône carrée contenant la lettre « i » s'affiche. Survolez l'icône pour afficher une infobulle contenant la description de l'erreur.

Pour configurer un nouveau graphique, cliquez sur le bouton *Créer un graphique* dans l'angle supérieur droit.

Options de modification en masse

Les boutons sous la liste offrent plusieurs options de modification en masse :

- *Copier* - copier les graphiques vers d'autres hôtes ou modèles
- *Supprimer* - supprimer les graphiques

Pour utiliser ces options, cochez les cases devant les graphiques concernés, puis cliquez sur le bouton souhaité.

Utilisation du filtre

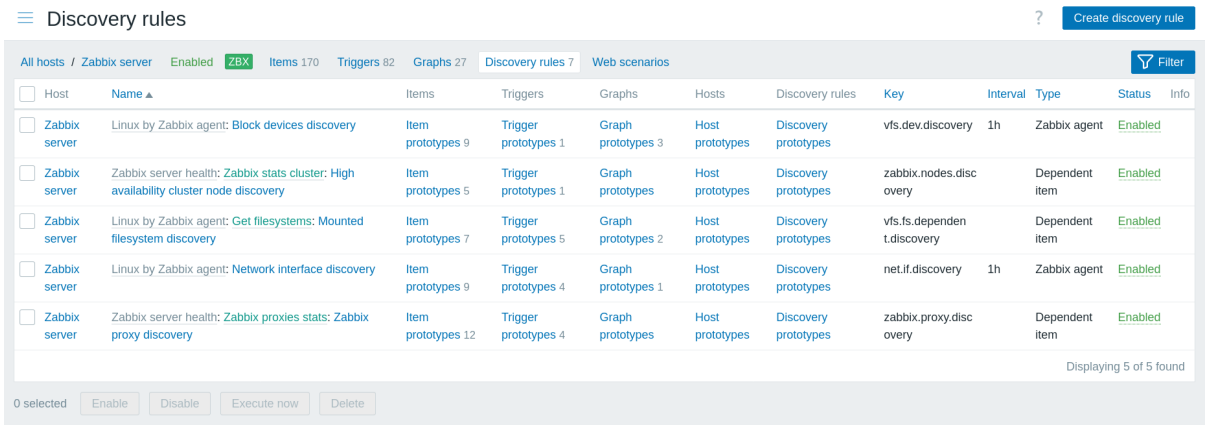
Vous pouvez filtrer les graphiques par groupe d'hôtes et par hôte. Pour de meilleures performances de recherche, les données sont recherchées avec les macros non résolues.

Règles de découverte

Vue d'ensemble

La liste des règles de découverte de bas niveau pour un hôte est accessible depuis *Collecte de données* > *Hôtes* en cliquant sur *Découverte* pour l'hôte concerné.

Une liste des règles de découverte de bas niveau existantes s'affiche. Il est également possible d'afficher toutes les règles de découverte indépendamment de l'hôte, ou toutes les règles de découverte d'un groupe d'hôtes spécifique, en modifiant les paramètres du filtre.



Données affichées :

Colonne	Description
Hôte	Le nom visible de l'hôte est affiché. Cliquer sur le nom de l'hôte ouvre le formulaire de configuration de l'hôte. En l'absence de nom d'hôte visible, le nom technique de l'hôte est affiché.
Nom	Nom de la règle, affiché sous forme de lien bleu. Cliquer sur le nom de la règle ouvre le formulaire de configuration de la règle de découverte de bas niveau. Si la règle de découverte appartient à un modèle, le nom du modèle est affiché avant le nom de la règle sous forme de lien gris. Cliquer sur le lien du modèle ouvre la liste des règles au niveau du modèle.
Éléments	Un lien vers la liste des prototypes d'éléments est affiché. Le nombre de prototypes d'éléments existants est affiché en gris.
Déclencheurs	Un lien vers la liste des prototypes de déclencheurs est affiché. Le nombre de prototypes de déclencheurs existants est affiché en gris.
Graphiques	Un lien vers la liste des prototypes de graphiques est affiché. Le nombre de prototypes de graphiques existants est affiché en gris.
Hôtes	Un lien vers la liste des prototypes d'hôtes est affiché. Le nombre de prototypes d'hôtes existants est affiché en gris.
Clé	La clé de l'élément utilisée pour la découverte est affichée.
Intervalle	La fréquence d'exécution de la découverte est affichée. *Remarque* : la découverte peut également être exécutée immédiatement en cliquant sur le bouton *Exécuter maintenant* situé sous la liste.
Type	Le type d'élément utilisé pour la découverte est affiché (agent Zabbix, agent SNMP, etc.).
État	L'état de la règle de découverte est affiché : *Activée*, *Désactivée* ou *Non prise en charge*. Cliquer sur l'état permet de le modifier : d'Activée à Désactivée (et inversement) ; de Non prise en charge à Désactivée (et inversement).
Informations	Si tout fonctionne correctement, aucune icône n'est affichée dans cette colonne. En cas d'erreur, une icône carrée portant la lettre « i » est affichée. Survolez l'icône pour afficher une infobulle contenant la description de l'erreur.

Pour configurer une nouvelle règle de découverte de bas niveau, cliquez sur le bouton *Créer une règle de découverte* dans l'angle supérieur droit.

Options de modification en masse

Les boutons sous la liste offrent plusieurs options de modification en masse :

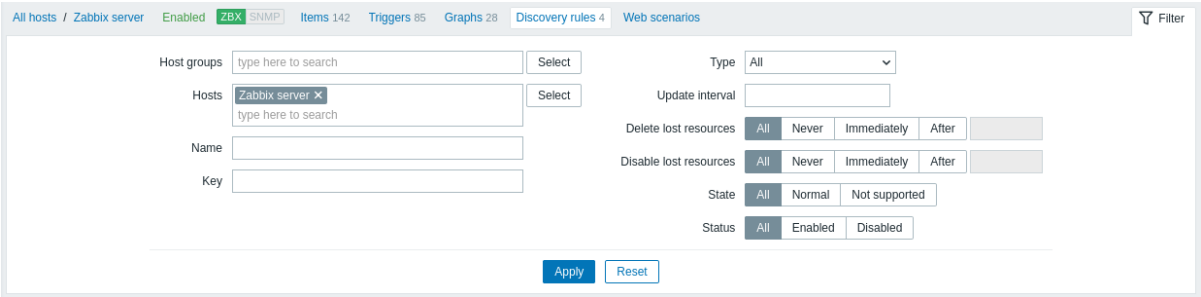
- *Enable* - changer l'état de la règle de découverte de bas niveau en *Enabled*.
- *Disable* - changer l'état de la règle de découverte de bas niveau en *Disabled*.
- *Execute now* - effectuer immédiatement la découverte basée sur les règles de découverte. Voir plus de détails. Notez que lors de l'exécution immédiate de la découverte, le cache de configuration n'est pas mis à jour, et le résultat ne reflétera donc pas les modifications très récentes de la configuration de la règle de découverte.
- *Delete* - supprimer les règles de découverte de bas niveau.

Pour utiliser ces options, cochez les cases devant les règles de découverte correspondantes, puis cliquez sur le bouton requis.

Utilisation du filtre

Vous pouvez utiliser le filtre pour afficher uniquement les règles de découverte qui vous intéressent. Pour de meilleures performances de recherche, les données sont recherchées avec les macros non résolues.

Le lien *Filter* est disponible au-dessus de la liste des règles de découverte. Si vous cliquez dessus, un filtre devient disponible et vous pouvez filtrer les règles de découverte par groupe d'hôtes, hôte, nom, clé d'élément, type d'élément et autres paramètres.

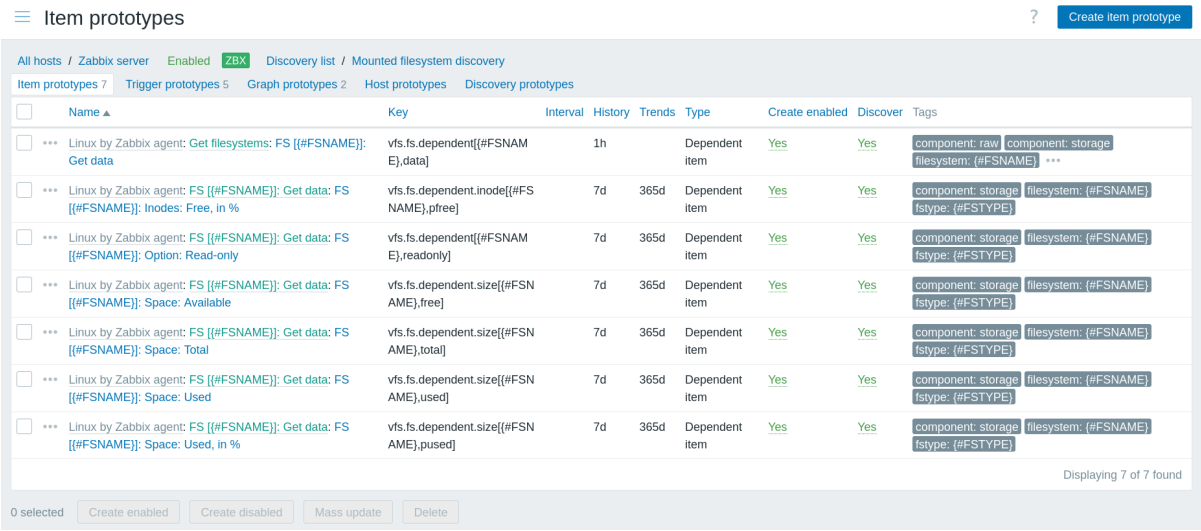


Parameter	Description
Host groups	Filtrer par un ou plusieurs groupes d'hôtes. La spécification d'un groupe d'hôtes parent sélectionne implicitement tous les groupes d'hôtes imbriqués.
Hosts	Filtrer par un ou plusieurs hôtes.
Name	Filtrer par le nom de la règle de découverte.
Key	Filtrer par la clé de l'élément de découverte.
Type	Filtrer par le type de l'élément de découverte.
Update interval	Filtrer par l'intervalle de mise à jour. Non disponible pour les éléments Zabbix trapper et dépendants.
Delete lost resources	Filtrer par la période *Delete lost resources*.
Disable lost resources	Filtrer par la période *Disable lost resources*.
SNMP OID	Filtrer par SNMP OID. Disponible uniquement si *SNMP agent* est sélectionné comme type.
State	Filtrer par l'état de la règle de découverte (All/Normal/Not supported).
Status	Filtrer par le statut de la règle de découverte (All/Enabled/Disabled).

Prototypes d'élément

Vue d'ensemble

Dans cette section, les prototypes d'éléments d'une règle de découverte de bas niveau sur le hôte sont affichés. Les prototypes d'éléments constituent la base des véritables éléments du hôte, créés lors de la découverte de bas niveau.



Données affichées :

Column	Description
Name	Nom du prototype d'élément, affiché sous forme de lien bleu. Cliquer sur le nom ouvre le formulaire de configuration du prototype d'élément. Si le prototype d'élément appartient à un modèle, le nom du modèle est affiché avant le nom de la règle sous forme de lien gris. Cliquer sur le lien du modèle ouvre la liste des prototypes d'éléments au niveau du modèle. Si le prototype d'élément est de type « Élément dépendant », le nom de l'élément maître est affiché en vert avant le nom du prototype d'élément.
Key	Clé du prototype d'élément.
Interval	Fréquence de la vérification.
History	Nombre de jours pendant lesquels conserver l'historique des données de l'élément.
Trends	Nombre de jours pendant lesquels conserver l'historique des tendances de l'élément.
Type	Type du prototype d'élément (agent Zabbix, agent SNMP, vérification simple, etc.).
Create enabled	Création de l'élément basé sur ce prototype : **Yes** - activée **No** - désactivée. Vous pouvez basculer entre « Yes » et « No » en cliquant dessus.
Discover	Découverte de l'élément basé sur ce prototype : **Yes** - découvrir **No** - ne pas découvrir. Vous pouvez basculer entre « Yes » et « No » en cliquant dessus.
Tags	Étiquettes du prototype d'élément.

Pour configurer un nouveau prototype d'élément, cliquez sur le bouton *Create item prototype* dans l'angle supérieur droit.

Options de modification en masse

Les boutons sous la liste offrent plusieurs options de modification en masse :

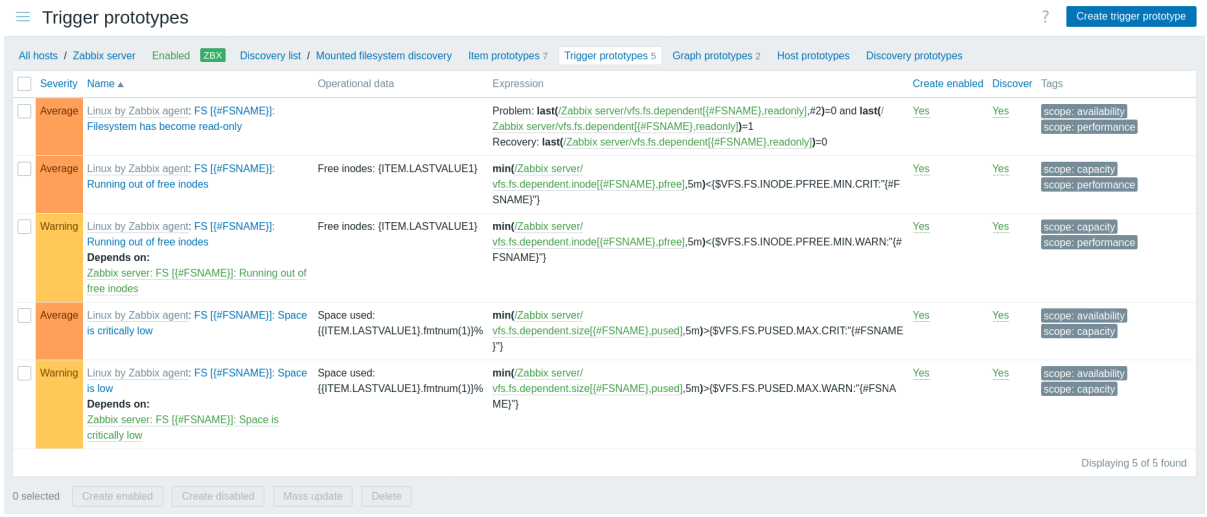
- *Créer activé* - créer ces éléments comme *Activé*
- *Créer désactivé* - créer ces éléments comme *Désactivé*
- *Mise à jour en masse* - mettre à jour en masse ces prototypes d'élément
- *Supprimer* - supprimer ces prototypes d'élément

Pour utiliser ces options, cochez les cases à cocher devant les prototypes d'élément concernés, puis cliquez sur le bouton requis.

Prototypes de déclencheur

Vue d'ensemble

Dans cette section, les prototypes de déclencheurs d'une règle de découverte de bas niveau sur le hôte sont affichés. Les prototypes de déclencheurs constituent la base des véritables déclencheurs du hôte, qui sont créés lors de la découverte de bas niveau.



Données affichées :

Column	Description
Nom	Nom du prototype de déclencheur, affiché sous forme de lien bleu. Cliquer sur le nom ouvre le formulaire de configuration du prototype de déclencheur. Si le prototype de déclencheur appartient à un modèle lié, le nom du modèle est affiché avant le nom du déclencheur sous forme de lien gris. Cliquer sur le lien du modèle ouvre la liste des prototypes de déclencheurs au niveau du modèle lié.
Données opérationnelles	Format des données opérationnelles du déclencheur, contenant des chaînes arbitraires et des macros qui seront résolues dynamiquement dans *Surveillance* > *Problèmes*.
Création activée	Créer le déclencheur sur la base de ce prototype comme suit : **Oui** - activé **Non** - désactivé. Vous pouvez basculer entre « Oui » et « Non » en cliquant dessus.
Découvrir	Découvrir le déclencheur sur la base de ce prototype : **Oui** - découvrir **Non** - ne pas découvrir. Vous pouvez basculer entre « Oui » et « Non » en cliquant dessus.
Étiquettes	Les étiquettes du prototype de déclencheur sont affichées.

Pour configurer un nouveau prototype de déclencheur, cliquez sur le bouton *Créer un prototype de déclencheur* dans l'angle supérieur droit.

Options de modification en masse

Les boutons sous la liste offrent plusieurs options de modification en masse :

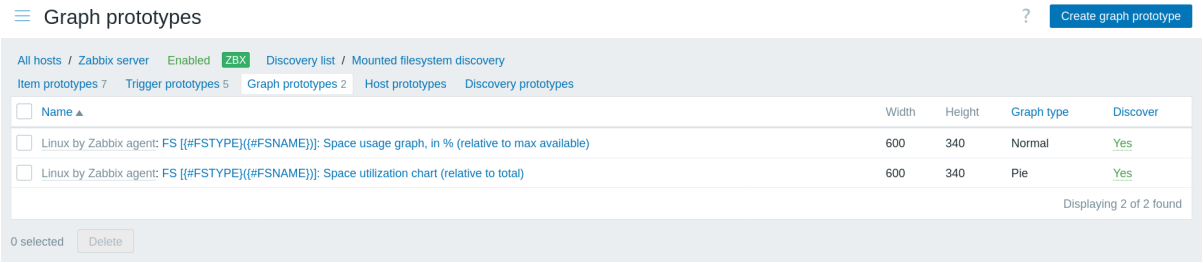
- *Créer activé* - créer ces déclencheurs comme *Activé*
- *Créer désactivé* - créer ces déclencheurs comme *Désactivé*
- *Mise à jour en masse* - mettre à jour en masse ces prototypes de déclencheur
- *Supprimer* - supprimer ces prototypes de déclencheur

Pour utiliser ces options, cochez les cases à cocher devant les prototypes de déclencheur correspondants, puis cliquez sur le bouton requis.

Prototypes de graphique

Vue d'ensemble

Dans cette section, les prototypes de graphes d'une règle de découverte de bas niveau sur le hôte sont affichés. Les prototypes de graphes constituent la base des graphes réels du hôte, qui sont créés lors de la découverte de bas niveau.



Données affichées :

Column	Description
Name	Nom du prototype de graphe, affiché sous forme de lien bleu. Cliquer sur le nom ouvre le formulaire de configuration du prototype de graphe. Si le prototype de graphe appartient à un modèle lié, le nom du modèle est affiché avant le nom du graphe sous forme de lien gris. Cliquer sur le lien du modèle ouvre la liste des prototypes de graphes au niveau du modèle lié.
Width	Largeur du prototype de graphe affichée.
Height	Hauteur du prototype de graphe affichée.
Type	Type du prototype de graphe affiché : *Normal*, *Stacked*, *Pie* ou *Exploded*.
Discover	Découvrir le graphe sur la base de ce prototype : **Yes** - découvrir **No** - ne pas découvrir. Vous pouvez basculer entre « Yes » et « No » en cliquant dessus.

Pour configurer un nouveau prototype de graphe, cliquez sur le bouton *Create graph prototype* dans l'angle supérieur droit.

Options de modification en masse

Les boutons sous la liste offrent plusieurs options de modification en masse :

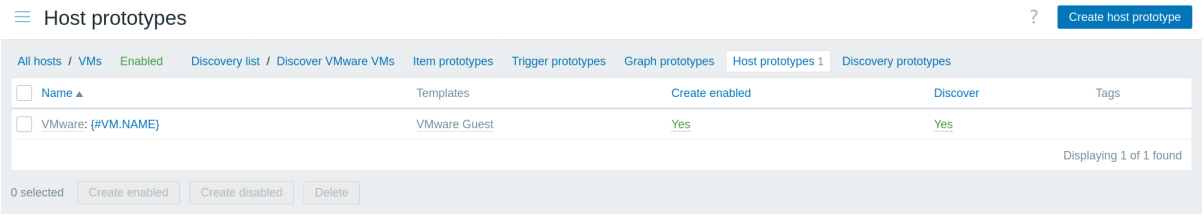
- *Supprimer* - supprimer ces prototypes de graphique

Pour utiliser ces options, cochez les cases devant les prototypes de graphique concernés, puis cliquez sur le bouton requis.

Modèles d'hôte

Vue d'ensemble

Dans cette section, les prototypes d'hôtes d'une règle de découverte de bas niveau sur l'hôte sont affichés. Les prototypes d'hôtes constituent la base des véritables hôtes créés lors de la découverte de bas niveau.



Données affichées :

Column	Description
Name	Nom du prototype d'hôte, affiché sous forme de lien bleu. Cliquer sur le nom ouvre le formulaire de configuration du prototype d'hôte. Si le prototype d'hôte appartient à un modèle lié, le nom du modèle est affiché avant le nom de l'hôte sous forme de lien gris. Cliquer sur le lien du modèle ouvre la liste des prototypes d'hôtes au niveau du modèle lié.
Templates	Les modèles du prototype d'hôte sont affichés.
Create enabled	Créer l'hôte sur la base de ce prototype comme suit : **Yes** - activé **No** - désactivé. Vous pouvez basculer entre « Yes » et « No » en cliquant dessus.
Discover	Découvrir l'hôte sur la base de ce prototype : **Yes** - découvrir **No** - ne pas découvrir. Vous pouvez basculer entre « Yes » et « No » en cliquant dessus.
Tags	Les tags du prototype d'hôte sont affichés.

Pour configurer un nouveau prototype d'hôte, cliquez sur le bouton *Create host prototype* dans l'angle supérieur droit.

Options de modification en masse

Les boutons situés sous la liste offrent quelques options de modification en masse :

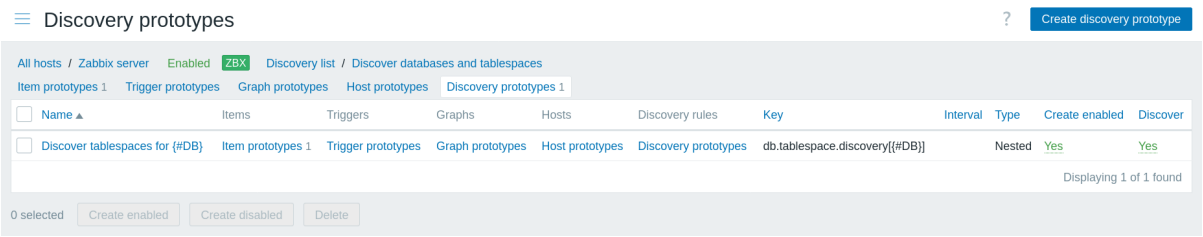
- *Créer activés* - créer ces hôtes comme *Activés*
- *Créer désactivés* - créer ces hôtes comme *Désactivés*
- *Supprimer* - supprimer ces prototypes d'hôte

Pour utiliser ces options, cochez les cases devant les prototypes d'hôte correspondants, puis cliquez sur le bouton requis.

Prototypes de découverte

Vue d'ensemble

Dans cette section, les prototypes de découverte d'une règle de découverte de bas niveau sur le hôte sont affichés. Les prototypes de découverte sont des règles de découverte imbriquées de la règle de découverte parente.



Données affichées :

Colonne	Description
Nom	Nom du prototype de découverte, affiché sous forme de lien bleu. Cliquer sur le nom ouvre le formulaire de configuration du prototype de découverte. Si le prototype de découverte appartient à un modèle lié, le nom du modèle est affiché avant le nom du hôte sous forme de lien gris. Cliquer sur le lien du modèle ouvre la liste des prototypes de découverte au niveau du modèle lié.
Éléments	Un lien vers la liste des prototypes d'éléments est affiché. Le nombre de prototypes d'éléments existants est affiché en gris.
Déclencheurs	Un lien vers la liste des prototypes de déclencheurs est affiché. Le nombre de prototypes de déclencheurs existants est affiché en gris.
Graphiques	Un lien vers la liste des prototypes de graphiques est affiché. Le nombre de prototypes de graphiques existants est affiché en gris.
Hôtes	Un lien vers la liste des prototypes de hôtes est affiché. Le nombre de prototypes de hôtes existants est affiché en gris.
Règles de découverte	Un lien vers la liste des prototypes de découverte est affiché. Le nombre de prototypes de découverte existants est affiché en gris.
Clé	La clé de l'élément utilisée pour la découverte est affichée.
Intervalle	La fréquence d'exécution de la découverte est affichée. *Remarque* : la découverte peut également être exécutée immédiatement en cliquant sur le bouton *Exécuter maintenant* situé sous la liste.
Type	Le type d'élément utilisé pour la découverte est affiché (agent Zabbix, agent SNMP, etc.).
Création activée	L'état de création du prototype de découverte est affiché : création activée (*Oui*) ou création désactivée (*Non*). En cliquant sur l'état, vous pouvez le modifier, de Oui à Non (et inversement).
Découvrir	L'état de découverte du prototype de découverte est affiché : découvrir (*Oui*) ou ne pas découvrir (*Non*). En cliquant sur l'état, vous pouvez le modifier, de Oui à Non (et inversement).

Pour configurer un nouveau prototype de découverte, cliquez sur le bouton *Créer un prototype de découverte* dans l'angle supérieur droit.

Options de modification en masse

Les boutons sous la liste offrent plusieurs options de modification en masse :

- *Créer activé* - créer ces prototypes de découverte comme *Activé*
- *Créer désactivé* - créer ces prototypes de découverte comme *Désactivé*
- *Supprimer* - supprimer ces prototypes de découverte

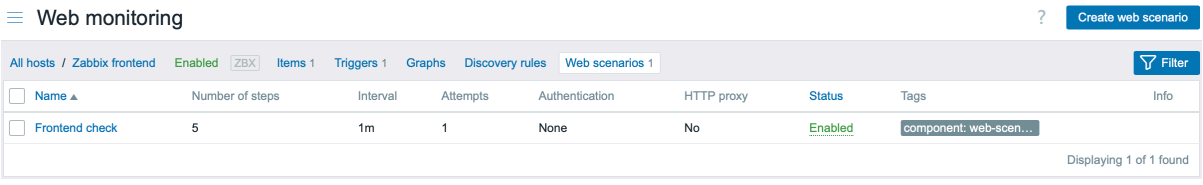
Pour utiliser ces options, cochez les cases devant les prototypes de découverte concernés, puis cliquez sur le bouton requis.

Scénarios web

Vue d'ensemble

La liste des scénarios web d'un hôte est accessible depuis *Collecte de données* > *Hôtes*, en cliquant sur *Web* pour l'hôte concerné.

Une liste des scénarios web existants s'affiche.



Données affichées :

Colonne	Description
Nom	Nom du scénario web. Cliquer sur le nom du scénario web ouvre le formulaire de configuration du scénario web. Si le scénario web appartient à un modèle, le nom du modèle est affiché avant le nom du scénario web sous forme de lien gris. Cliquer sur le lien du modèle ouvre la liste des scénarios web au niveau du modèle.
Nombre d'étapes	Nombre d'étapes contenues dans le scénario.
Intervalle de mise à jour	Fréquence d'exécution du scénario.
Tentatives	Nombre de tentatives d'exécution des étapes du scénario web.
Authentification	Méthode d'authentification utilisée : Basic, NTLM, Kerberos, Digest ou aucune.
Proxy HTTP	Affiche le proxy HTTP ou « Non » s'il n'est pas utilisé.
État	État du scénario web : *Activé* ou *Désactivé*. Cliquer sur l'état permet de le modifier.
Étiquettes	Les étiquettes du scénario web sont affichées. Jusqu'à trois étiquettes (paires nom:valeur) peuvent être affichées. S'il y en a davantage, un lien « ... » permet de les afficher toutes au survol de la souris.
Informations	Si tout fonctionne correctement, aucune icône n'est affichée dans cette colonne. En cas d'erreur, une icône carrée contenant la lettre « i » est affichée. Survolez l'icône pour afficher une infobulle contenant la description de l'erreur.

Pour configurer un nouveau scénario web, cliquez sur le bouton *Créer un scénario web* dans l'angle supérieur droit.

Options de modification en masse

Les boutons sous la liste offrent plusieurs options de modification en masse :

- *Activer* - changer l'état du scénario en *Activé*
- *Désactiver* - changer l'état du scénario en *Désactivé*
- *Effacer l'historique et les tendances* - effacer les données d'historique et de tendances des scénarios
- *Supprimer* - supprimer les scénarios web

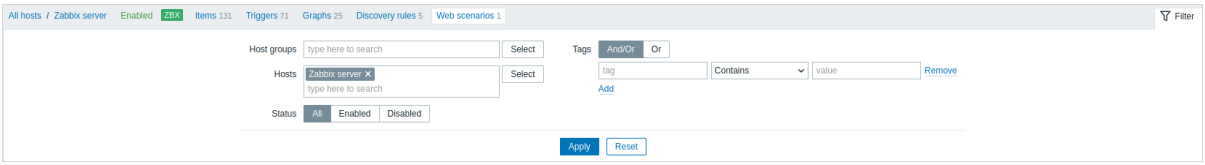
Pour utiliser ces options, cochez les cases devant les scénarios web concernés, puis cliquez sur le bouton requis.

Utilisation du filtre

Vous pouvez utiliser le filtre pour afficher uniquement les scénarios qui vous intéressent.

Pour de meilleures performances de recherche, la recherche des données s'effectue avec les macros non résolues.

Le lien *Filter* est disponible au-dessus de la liste des scénarios web. En cliquant dessus, vous ouvrirez un filtre dans lequel vous pourrez spécifier les critères de filtrage souhaités.



Parameter	Description
Host groups	Filtrer par un ou plusieurs groupes d'hôtes. La spécification d'un groupe d'hôtes parent sélectionne implicitement tous les groupes d'hôtes imbriqués.
Hosts	Filtrer par un ou plusieurs hôtes. Si des groupes d'hôtes sont déjà sélectionnés ci-dessus, la sélection des hôtes est limitée à ces groupes.
Status	Filtrer par statut de scénario web.

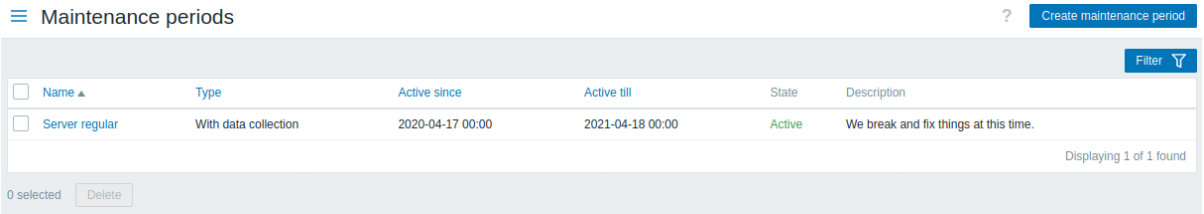
Parameter	Description
Tags	Filtrer par nom et valeur de tag de scénario web. Il est possible d'inclure ainsi que d'exclure des tags et des valeurs de tag spécifiques. Plusieurs conditions peuvent être définies. La correspondance du nom de tag est toujours sensible à la casse. Plusieurs opérateurs sont disponibles pour chaque condition : **Exists** - inclure les noms de tag spécifiés ; **Equals** - inclure les noms et valeurs de tag spécifiés (sensible à la casse) ; **Contains** - inclure les noms de tag spécifiés dont les valeurs de tag contiennent la chaîne saisie (correspondance de sous-chaîne, insensible à la casse) ; **Does not exist** - exclure les noms de tag spécifiés ; **Does not equal** - exclure les noms et valeurs de tag spécifiés (sensible à la casse) ; **Does not contain** - exclure les noms de tag spécifiés dont les valeurs de tag contiennent la chaîne saisie (correspondance de sous-chaîne, insensible à la casse). Il existe deux types de calcul pour les conditions : **And/Or** - toutes les conditions doivent être remplies, les conditions ayant le même nom de tag seront regroupées par la condition Or ; **Or** - une seule condition suffit.

Maintenance

Présentation

Dans la section *Collecte de données* > *Maintenance*, les utilisateurs peuvent configurer et gérer les périodes de maintenance des hôtes.

Une liste des périodes de maintenance existantes avec leurs détails est affichée.



Données affichées :

Colonne	Description
Nom	Nom de la période de maintenance. Cliquer sur le nom de la période de maintenance ouvre le formulaire de configuration de la période de maintenance.
Type	Le type de maintenance est affiché : *Avec collecte de données* ou *Sans collecte de données*
Active depuis	Date et heure auxquelles l'exécution des périodes de maintenance devient active. Remarque : cette heure n'active pas une période de maintenance ; les périodes de maintenance doivent être définies séparément.
Active jusqu'à	Date et heure auxquelles l'exécution des périodes de maintenance cesse d'être active.
État	État de la période de maintenance : **À venir** - deviendra bientôt active **Active** - est active **Expirée** - n'est plus active
Description	Description de la période de maintenance affichée.

Pour configurer une nouvelle période de maintenance, cliquez sur le bouton *Créer une période de maintenance* dans le coin supérieur droit.

Options de modification en masse

Un bouton sous la liste propose une option de modification en masse :

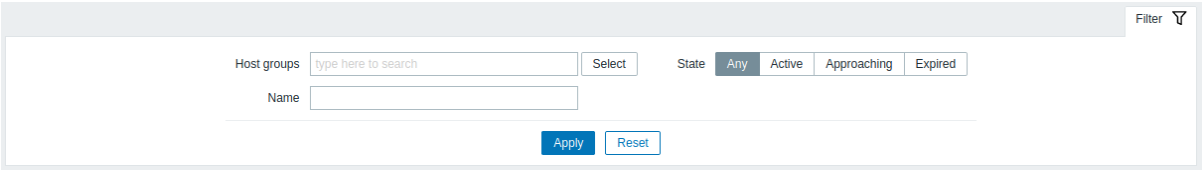
- *Supprimer* - supprimer les périodes de maintenance

Pour utiliser cette option, cochez les cases devant les périodes de maintenance concernées, puis cliquez sur *Supprimer*.

Utilisation du filtre

Vous pouvez utiliser le filtre pour afficher uniquement les périodes de maintenance qui vous intéressent. Pour de meilleures performances de recherche, les données sont recherchées avec les macros non résolues.

Le lien *Filtre* est disponible au-dessus de la liste des périodes de maintenance. Si vous cliquez dessus, un filtre devient disponible et vous pouvez filtrer les périodes de maintenance par groupe d'hôtes, nom et état.



Calcul des files d'attente pendant la maintenance

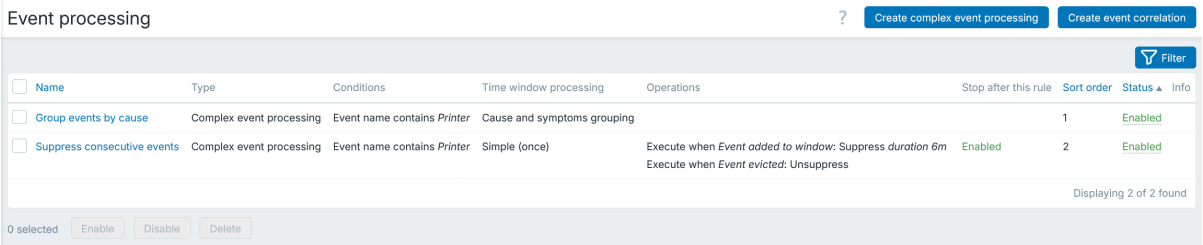
Attention:
Le proxy Zabbix n'a pas connaissance des périodes de maintenance ; voir Calcul des files d'attente pendant la maintenance pour plus de détails.

Event processing

Overview

In the *Data collection* > *Event processing* section users can configure and maintain:

- Complex event processing rules (CEP rules)
- Global event correlation rules (deprecated)



Displayed data:

Column	Description
Name	Name of the rule. Clicking the name opens the rule configuration form - for CEP rule or global correlation rule respectively.
Type	Type of the rule - *Complex event processing* or *Correlation*.
Conditions	Rule conditions are displayed.
Time window processing	Time window type is displayed.
Operations	Rule operations are displayed.
Stop after this rule	Stop execution after this rule - *Enabled* or *Disabled*.
Sort order	Sort order of the CEP rule.
Status	Rule status is displayed - *Enabled* or *Disabled*. By clicking the status you can change it.
Info	If everything is working correctly, no icon is displayed in this column. In case of errors, a square icon with the letter "i" is displayed. Hover over the icon to see a tooltip with the error description.

To configure a new CEP rule, click the *Create complex event processing* button in the upper-right corner.

To configure a new correlation rule, click the *Create event correlation* button in the upper-right corner.

Mass editing options

Buttons below the list offer some mass-editing options:

- *Enable* - change the rule status to *Enabled*
- *Disable* - change the rule status to *Disabled*

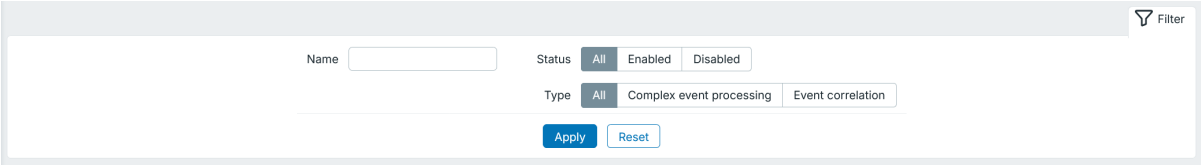
- *Delete* - delete the rules

To use these options, mark the checkboxes before the respective rules, then click the required button.

Using filter

You can use the filter to display only the rules you are interested in. For better search performance, data is searched with macros unresolved.

The *Filter* link is available above the list of CEP/correlation rules. If you click it, a filter becomes available where you can filter rules by name and status.

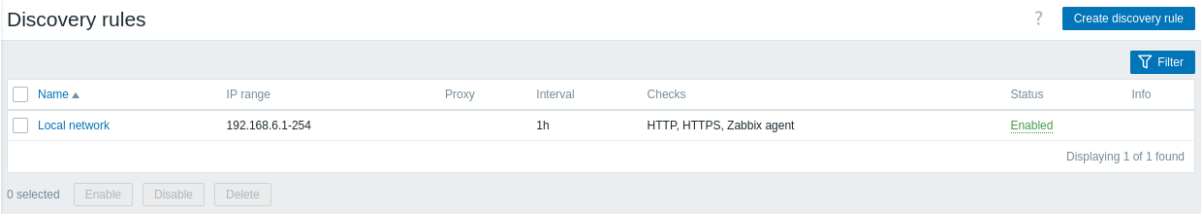


Découverte

Vue d'ensemble

Dans la section *Collecte de données* > *Découverte*, les utilisateurs peuvent configurer et gérer les règles de découverte.

Une liste des règles de découverte existantes avec leurs détails est affichée.



Données affichées :

Colonne	Description
Nom	Nom de la règle de découverte. Cliquer sur le nom de la règle de découverte ouvre le formulaire de configuration de la règle de découverte.
Plage IP	La plage d'adresses IP à utiliser pour l'analyse du réseau est affichée.
Proxy	Le nom du proxy est affiché si la découverte est effectuée par le proxy.
Intervalle	La fréquence d'exécution de la découverte est affichée.
Vérifications	Les types de vérifications utilisés pour la découverte sont affichés.
État	L'état de la règle de découverte est affiché : *Activée* ou *Désactivée*. Cliquer sur l'état permet de le modifier.
Informations	Si tout fonctionne correctement, rien n'est affiché dans cette colonne. En cas d'erreur, une icône d'information rouge portant la lettre « i » est affichée. Survolez l'icône pour afficher une infobulle contenant la description de l'erreur.

Pour configurer une nouvelle règle de découverte, cliquez sur le bouton *Créer une règle de découverte* dans le coin supérieur droit.

Options de modification en masse

Les boutons sous la liste offrent plusieurs options de modification en masse :

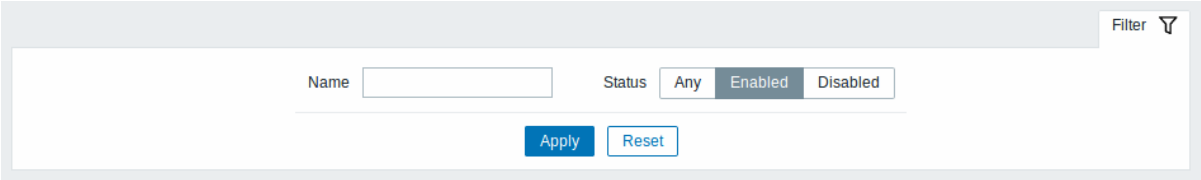
- *Activer* - modifier l'état de la règle de découverte en *Activé*
- *Désactiver* - modifier l'état de la règle de découverte en *Désactivé*
- *Supprimer* - supprimer les règles de découverte

Pour utiliser ces options, cochez les cases devant les règles de découverte concernées, puis cliquez sur le bouton requis.

Utilisation du filtre

Vous pouvez utiliser le filtre pour afficher uniquement les règles de découverte qui vous intéressent. Pour de meilleures performances de recherche, les données sont recherchées avec les macros non résolues.

Le lien *Filtre* est disponible au-dessus de la liste des règles de découverte. Si vous cliquez dessus, un filtre devient disponible et vous pouvez filtrer les règles de découverte par nom et par statut.



Alertes

Aperçu

Ce menu comprend des sections liées à la configuration des alertes dans Zabbix.

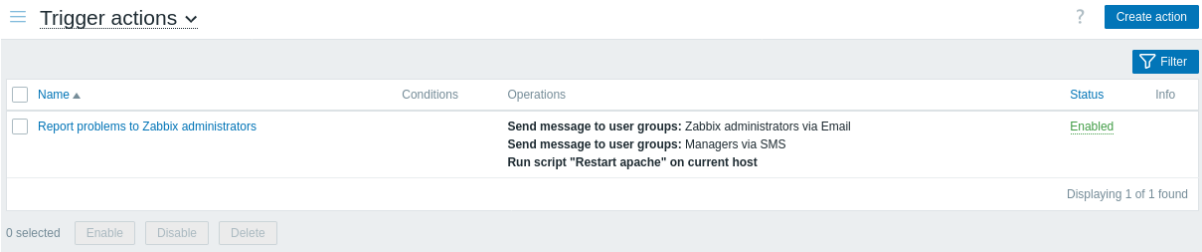
Actions

Vue d'ensemble

Dans la section *Alertes* > *Actions*, les utilisateurs peuvent configurer et gérer les actions.

Les actions affichées sont celles qui sont associées à la source d'événements sélectionnée (déclencheurs, services, découverte, enregistrement automatique, actions internes).

Pour passer à une autre source d'événements, cliquez sur *Actions* dans la section du menu *Alertes*. Il est également possible de passer d'une source à l'autre à l'aide de la liste déroulante du titre située dans le coin supérieur gauche.



Données affichées :

Colonne	Description
Nom	Nom de l'action. Cliquez sur le nom de l'action pour ouvrir le formulaire de configuration de l'action.
Conditions	Les conditions de l'action sont affichées.
Opérations	Les opérations de l'action sont affichées. La liste des opérations affiche également le type de média (e-mail, SMS ou script) utilisé pour la notification, ainsi que le nom et le prénom (entre parenthèses après le nom d'utilisateur) du destinataire de la notification. Une opération d'action peut être une notification ou une commande distante, selon le type d'opération sélectionné.
État	L'état de l'action est affiché : *Activée* ou *Désactivée*. Cliquez sur l'état pour le modifier. Consultez la section Escalades pour plus d'informations sur ce qui se produit lorsqu'une action est désactivée pendant une escalade en cours.
Informations	Si tout fonctionne correctement, aucune icône n'est affichée dans cette colonne. En cas d'erreurs, par exemple si des opérations ou des conditions d'action sont manquantes après une mise à niveau ou une découverte de bas niveau, une icône d'avertissement est affichée. Survolez l'icône pour afficher une infobulle contenant la description de l'erreur.

Pour configurer une nouvelle action, cliquez sur le bouton *Créer une action* dans le coin supérieur droit.

Pour les utilisateurs ne disposant pas de droits de super administrateur, les actions sont affichées selon les paramètres d'autorisation. Cela signifie que, dans certains cas, un utilisateur ne disposant pas de droits de super administrateur ne peut pas voir la liste complète des actions en raison de certaines restrictions d'autorisation. Une action est affichée à l'utilisateur ne disposant pas de droits de super administrateur si les conditions suivantes sont remplies :

- L'utilisateur dispose d'un accès en lecture-écriture aux groupes de hôtes, aux hôtes, aux modèles et aux déclencheurs dans les conditions de l'action

- L'utilisateur dispose d'un accès en lecture-écriture aux groupes de hôtes, aux hôtes et aux modèles dans les opérations d'action, les opérations de récupération et les opérations de mise à jour
- L'utilisateur dispose d'un accès en lecture aux groupes d'utilisateurs et aux utilisateurs dans les opérations d'action, les opérations de récupération et les opérations de mise à jour

Options de modification en masse

Les boutons sous la liste offrent plusieurs options de modification en masse :

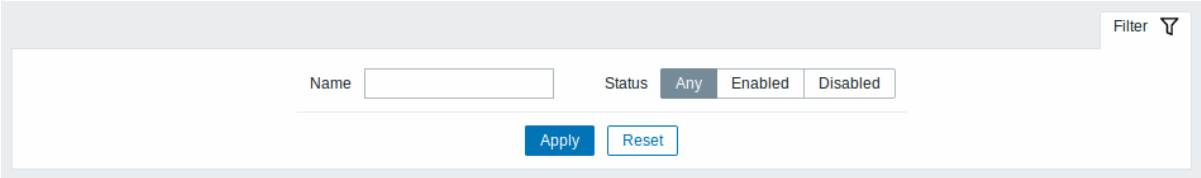
- *Activer* - changer l'état de l'action en *Activé*
- *Désactiver* - changer l'état de l'action en *Désactivé*
- *Supprimer* - supprimer les actions

Pour utiliser ces options, cochez les cases devant les actions correspondantes, puis cliquez sur le bouton requis.

Utilisation d'un filtre

Vous pouvez utiliser le filtre pour afficher uniquement les actions qui vous intéressent. Pour de meilleures performances de recherche, les données sont recherchées avec des macros non résolues.

Le lien *Filtre* est disponible au-dessus de la liste des actions. Si vous cliquez dessus, un filtre devient disponible et vous pouvez filtrer les actions par nom et statut.



Types de média

Vue d'ensemble

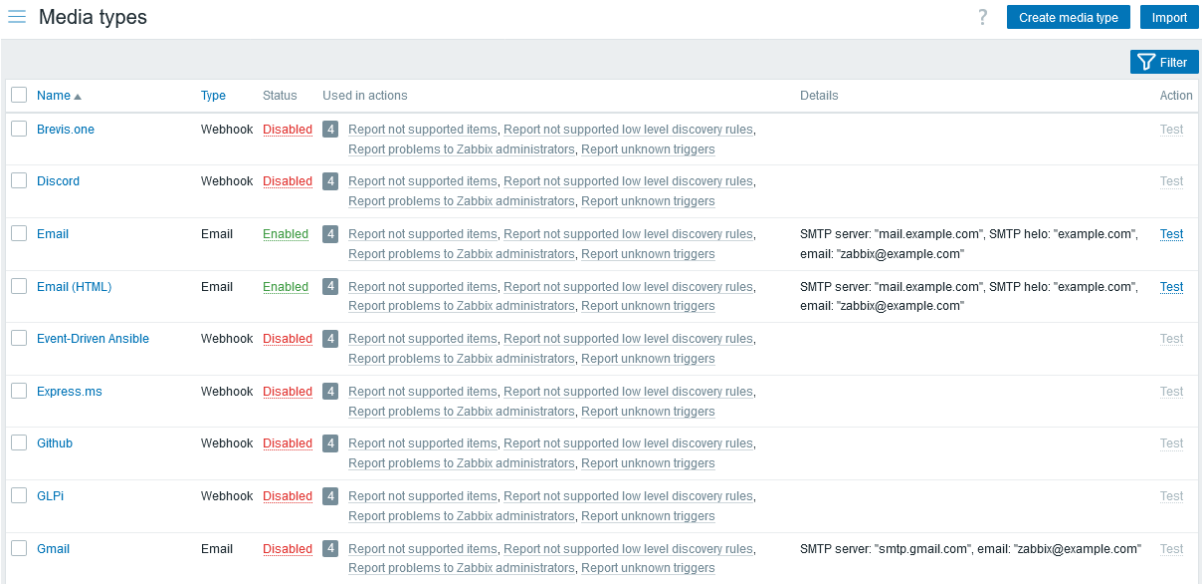
Dans la section *Alertes* > *Types de média*, les utilisateurs peuvent configurer et gérer les informations relatives aux types de média.

Ce menu est disponible uniquement pour les utilisateurs de type SuperAdmin.

Les informations relatives aux types de média contiennent des instructions générales sur l'utilisation d'un média comme canal de diffusion des notifications. Les détails spécifiques, tels que les adresses e-mail individuelles auxquelles envoyer une notification, sont associés aux utilisateurs concernés.

Une liste des types de média existants et de leurs détails est affichée.

Note:
Si des méthodes de diffusion (*E-mail*, *SMS*, *Webhook*, *Script* ou *Push*) sont spécifiées dans l'indicateur de fonctionnalité `$ZBX_FEATURE_FLAGS['media_type_denylist']` du fichier de configuration de l'interface, les types de média pré-configurés correspondants sont désactivés dans cette section.



Données affichées :

Colonne	Description
Nom	Nom du type de média. Cliquer sur le nom ouvre le formulaire de configuration du type de média.
Type	Le type de média (e-mail, SMS, etc.) est affiché.
État	L'état du type de média est affiché : *Activé* ou *Désactivé*. Cliquer sur l'état permet de le modifier.
Utilisé dans les actions	Les actions dans lesquelles le type de média est utilisé sont affichées, précédées du nombre total de ces actions. Cliquer sur le nom de l'action ouvre le formulaire de configuration de l'action. Si l'utilisateur ne dispose d'aucune permission sur l'action, le nom n'est pas cliquable.
Détails	Les informations détaillées du type de média sont affichées.
Actions	L'action suivante est disponible : **Tester** - cliquer pour ouvrir un formulaire de test dans lequel vous pouvez saisir les paramètres du type de média (par exemple, une adresse de destinataire avec un objet et un corps de test) et envoyer un message de test afin de vérifier que le type de média configuré fonctionne. Voir également : Test d'un type de média.

Pour configurer un nouveau type de média, cliquez sur le bouton *Créer un type de média* dans le coin supérieur droit.

Pour importer un type de média, cliquez sur le bouton *Importer* dans le coin supérieur droit.

Options de modification en masse

Les boutons sous la liste offrent plusieurs options de modification en masse :

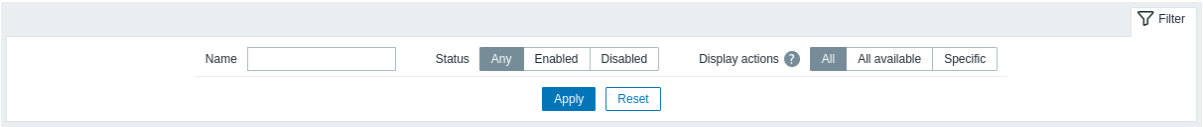
- *Activer* - change l'état du type de média en *Activé*
- *Désactiver* - change l'état du type de média en *Désactivé*
- *Exporter* - exporte les types de média vers un fichier YAML, XML ou JSON
- *Supprimer* - supprime les types de média

Pour utiliser ces options, cochez les cases à cocher devant les types de média concernés, puis cliquez sur le bouton requis.

Utilisation du filtre

Vous pouvez utiliser le filtre pour afficher uniquement les types de média qui vous intéressent. Pour de meilleures performances de recherche, les données sont recherchées avec les macros non résolues.

Le lien *Filter* est disponible au-dessus de la liste des types de média. Si vous cliquez dessus, un filtre devient disponible, où vous pouvez filtrer les types de média par nom et par statut. De plus, vous pouvez utiliser le filtre pour afficher les actions dans la colonne *Used in actions* en fonction de la portée de leur utilisation du type de média (définie par le paramètre *Send to media type* dans les détails de l'opération de l'action).



Scripts

Vue d'ensemble

Dans la section *Alertes* > *Scripts*, les scripts globaux définis par l'utilisateur peuvent être configurés et gérés.

Ce menu est disponible uniquement pour les utilisateurs de type SuperAdmin.

Les scripts globaux, selon la portée configurée et les permissions de l'utilisateur, peuvent être exécutés :

- Depuis le menu de l'hôte à différents endroits de l'interface (*Tableau de bord*, *Problèmes*, *Dernières données*, *Cartes*, etc.).
- Depuis le menu de l'événement.
- En tant qu'opération d'une action.

Les scripts sont exécutés par l'agent Zabbix, le serveur Zabbix (proxy) ou le serveur Zabbix uniquement. Voir également Exécution de commandes.

Les commandes distantes sont désactivées par défaut sur l'agent Zabbix et le proxy Zabbix. Elles peuvent être activées comme suit :

- Pour les commandes distantes exécutées par l'agent Zabbix, ajoutez un paramètre `AllowKey=system.run[<command>,*]` ou `AllowKeyRegexp=^system\.run\[<command>,.*\]$` pour chaque commande autorisée dans la configuration de l'agent, où * correspond à l'un ou l'autre des modes d'exécution (`wait` ou `nowait`).
- Pour les commandes distantes exécutées par le proxy Zabbix, définissez le paramètre `EnableRemoteCommands` sur 1 dans la configuration du proxy. Notez qu'il n'est pas nécessaire d'activer les commandes distantes sur le proxy Zabbix lorsqu'elles sont exécutées par l'agent Zabbix sur un hôte surveillé par le proxy.

Attention:
N'utilisez pas `AllowKey=system.run[*]` ou `AllowKeyRegexp=^system\.run\[.*\]$` sauf si cela est nécessaire, car ces paramètres permettent l'exécution de n'importe quelle commande sur l'hôte. Pour des raisons de sécurité, il est recommandé d'autoriser uniquement les commandes exactes, tandis que `system.run[*]` est refusé par défaut (`DenyKey=system.run[*]`).

L'exécution des scripts globaux sur le serveur Zabbix peut être désactivée en définissant `EnableGlobalScripts=0` dans la configuration du serveur. Pour les nouvelles installations à partir de Zabbix 7.0, l'exécution des scripts globaux sur le serveur Zabbix est désactivée par défaut.

Une liste des scripts existants avec leurs détails est affichée.



Données affichées :

Colonne	Description
Nom	Nom du script. Cliquer sur le nom du script ouvre le formulaire de configuration du script.
Portée	Portée du script : opération d'action, action manuelle sur l'hôte ou action manuelle sur l'événement. Ce paramètre détermine où le script est disponible : dans les commandes distantes des opérations d'une action, depuis le menu de l'hôte ou depuis le menu de l'événement, respectivement.
Utilisé dans les actions	Toutes les actions dans lesquelles le script est utilisé sont affichées, précédées du nombre total de ces actions. Cliquer sur le nom de l'action ouvre le formulaire de configuration de l'action. Si l'utilisateur ne dispose d'aucune permission sur l'action, le nom n'est pas cliquable.
Type	Le type de script est affiché : commande *URL*, *Webhook*, *Script*, *SSH*, *Telnet* ou *IPMI*.
Exécuter sur	Indique si le script sera exécuté sur l'agent Zabbix, le proxy Zabbix ou le serveur, ou uniquement sur le serveur Zabbix.
Commandes	Toutes les commandes à exécuter dans le script sont affichées. Rien n'est affiché ici pour les webhooks.
Groupe d'utilisateurs	Le groupe d'utilisateurs auquel le script est disponible est affiché (ou *Tous* pour tous les groupes d'utilisateurs).
Groupe d'hôtes	Le groupe d'hôtes pour lequel le script est disponible est affiché (ou *Tous* pour tous les groupes d'hôtes).
Accès à l'hôte	Le niveau de permission pour le groupe d'hôtes est affiché : *Lecture* ou *Écriture*. Seuls les utilisateurs disposant du niveau de permission requis pourront exécuter le script.

Pour configurer un nouveau script, cliquez sur le bouton *Créer un script* dans le coin supérieur droit.

Options de modification en masse

Un bouton sous la liste propose une option de modification en masse :

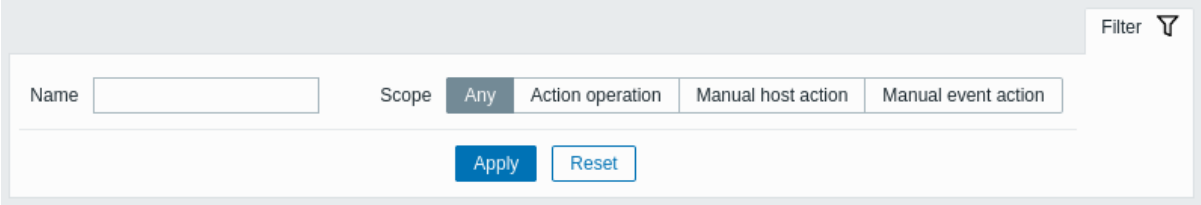
- *Supprimer* - supprimer les scripts

Pour utiliser cette option, cochez les cases devant les scripts concernés, puis cliquez sur *Supprimer*.

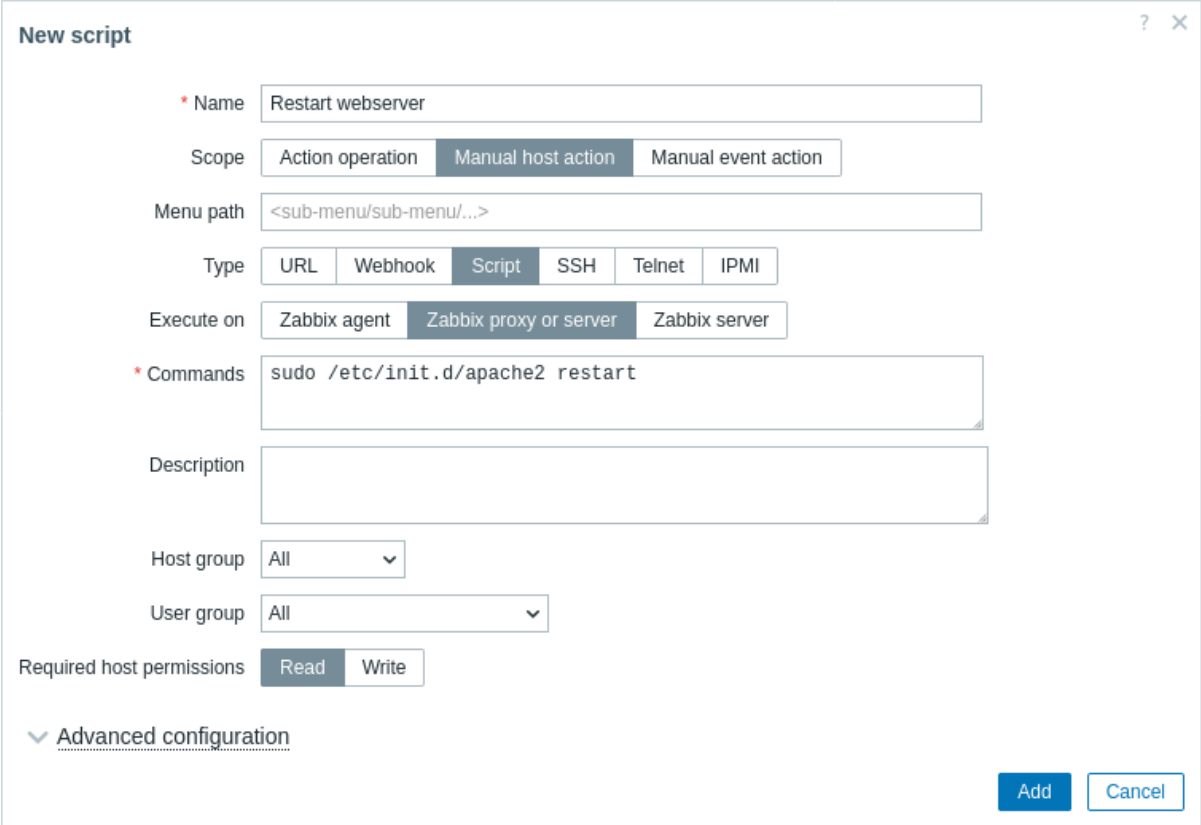
Utilisation du filtre

Vous pouvez utiliser le filtre pour afficher uniquement les scripts qui vous intéressent. Pour de meilleures performances de recherche, les données sont recherchées avec les macros non résolues.

Le lien *Filter* est disponible au-dessus de la liste des scripts. Si vous cliquez dessus, un filtre devient disponible, où vous pouvez filtrer les scripts par nom et par portée.



Configuration d'un script global



Note:
Il est recommandé d'utiliser les fonctions de macro plutôt que des macros simples dans les scripts globaux afin d'améliorer la sécurité, car les macros ne sont pas automatiquement échappées.

Attributs du script :

Paramètre	Description
Nom	Nom unique du script. Par exemple, `Clear /tmp filesystem`

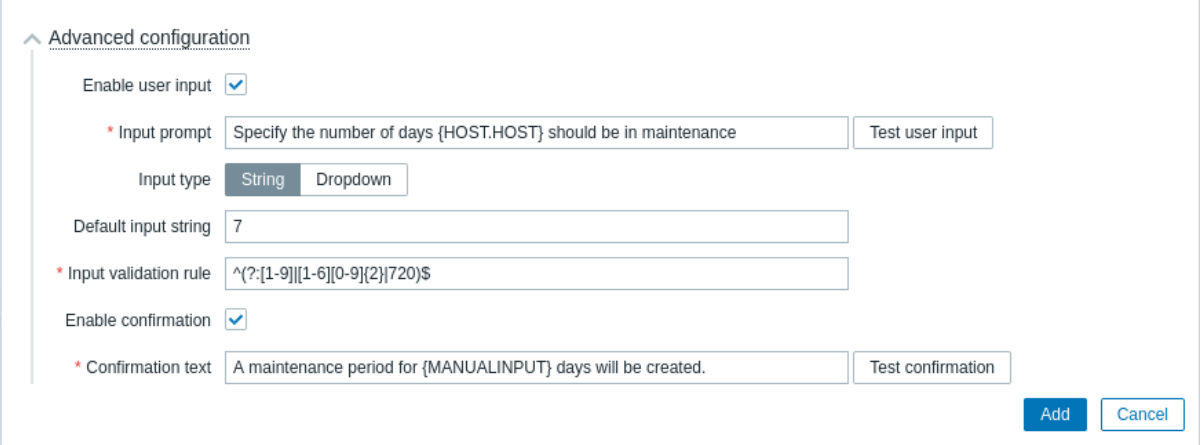
Paramètre	Description
Portée	Portée du script : opération d'action, action manuelle sur un hôte ou action manuelle sur un événement. Ce paramètre détermine où le script peut être utilisé : dans les commandes distantes des opérations d'action, depuis le menu de l'hôte ou depuis le menu de l'événement, respectivement. Définir la portée sur « Opération d'action » rend le script disponible pour tous les utilisateurs ayant accès à *Alertes* > *Actions*. Si un script est effectivement utilisé dans une action, sa portée ne peut pas être modifiée pour quitter « opération d'action ». **Prise en charge des macros** La portée influe sur l'ensemble des macros disponibles. Par exemple, les macros liées à l'utilisateur (`{USER.*}`) sont prises en charge dans les scripts afin de permettre la transmission d'informations sur l'utilisateur ayant lancé le script. Cependant, elles ne sont pas prises en charge si la portée du script est « opération d'action », car les opérations d'action sont exécutées automatiquement. Une macro `{MANUALINPUT}` permet de spécifier une saisie manuelle au moment de l'exécution du script. Elle est prise en charge pour les scripts d'action manuelle sur un hôte et d'action manuelle sur un événement. Pour connaître les autres macros prises en charge, recherchez « Notifications et commandes basées sur un déclencheur/Commandes basées sur un déclencheur », « Scripts d'action manuelle sur un hôte » et « Scripts d'action manuelle sur un événement » dans le tableau des macros prises en charge. Notez que si une macro peut être remplacée par une valeur contenant des espaces (par exemple, le nom d'un hôte), n'oubliez pas de la mettre entre guillemets si nécessaire.
Chemin du menu	Chemin de menu souhaité pour le script. Par exemple, `Default` ou `Default/` affichera le script dans le répertoire correspondant. Les menus peuvent être imbriqués, par exemple `Main menu/Sub menu1/Sub menu2`. Lors de l'accès aux scripts via le menu de l'hôte ou de l'événement dans les sections de supervision, ils seront organisés selon les répertoires indiqués. Ce champ s'affiche uniquement si « Action manuelle sur un hôte » ou « Action manuelle sur un événement » est sélectionné comme *Portée*.
Type	Cliquez sur le bouton correspondant pour sélectionner le type de script : commande **URL**, **Webhook**, **Script**, **SSH**, **Telnet** ou **IPMI**. Le type **URL** est disponible uniquement lorsque « Action manuelle sur un hôte » ou « Action manuelle sur un événement » est sélectionné comme *Portée*.
Type de script : **URL**	
URL	Indiquez l'URL pour y accéder rapidement depuis le menu de l'hôte ou le menu de l'événement. Les macros et les macros utilisateur personnalisées sont prises en charge. La prise en charge des macros dépend de la portée du script (voir *Portée* ci-dessus). Utilisez la macro {MANUALINPUT} dans ce champ pour pouvoir spécifier une saisie manuelle au moment de l'exécution du script, par exemple : `http://{MANUALINPUT}/zabbix/zabbix.php?action=dashboard.view` Les valeurs des macros ne doivent pas être encodées dans l'URL.
Ouvrir dans une nouvelle fenêtre	Détermine si l'URL doit être ouverte dans un nouvel onglet du navigateur ou dans l'onglet actuel.
Type de script : **Webhook**	
Paramètres	Spécifiez les variables du webhook sous forme de paires attribut-valeur. Voir également : configuration du média Webhook. Les macros et les macros utilisateur personnalisées sont prises en charge dans les valeurs des paramètres. La prise en charge des macros dépend de la portée du script (voir *Portée* ci-dessus).
Script	Saisissez le code JavaScript dans l'éditeur modal qui s'ouvre lorsque vous cliquez dans le champ du paramètre ou sur l'icône en forme de crayon située à côté. La prise en charge des macros dépend de la portée du script (voir *Portée* ci-dessus). Voir également : configuration du média Webhook, objets JavaScript supplémentaires.
Délai d'expiration	Délai d'expiration de l'exécution JavaScript (1 à 60 s, 30 s par défaut). Les suffixes de temps sont pris en charge (par exemple, `30s`, `1m`).
Type de script : **Script**	

Paramètre	Description
Exécuter sur	Cliquez sur le bouton correspondant pour exécuter le script shell sur : **Zabbix agent** - le script sera exécuté par Zabbix agent (si l'élément system.run est autorisé) sur l'hôte **Zabbix proxy ou serveur** - le script sera exécuté par Zabbix proxy ou serveur, selon que l'hôte est supervisé par un proxy ou un serveur. Il sera exécuté sur le proxy si cette fonction est activée par EnableRemoteCommands. Il sera exécuté sur le serveur si les scripts globaux sont activés par le paramètre du serveur EnableGlobalScripts. **Zabbix serveur** - le script sera exécuté uniquement par Zabbix serveur. Cette option ne sera pas disponible si les scripts globaux sont désactivés par le paramètre du serveur EnableGlobalScripts.
Commandes	Saisissez le chemin complet des commandes à exécuter dans le script. La prise en charge des macros dépend de la portée du script (voir *Portée* ci-dessus). Les macros utilisateur personnalisées sont prises en charge.
Type de script : **SSH**	
Méthode d'authentification	Sélectionnez la méthode d'authentification : mot de passe ou clé publique.
Nom d'utilisateur	Saisissez le nom d'utilisateur.
Mot de passe	Saisissez le mot de passe. Ce champ est disponible si « Mot de passe » est sélectionné comme méthode d'authentification.
Fichier de clé publique	Saisissez le chemin du fichier de clé publique. Ce champ est disponible si « Clé publique » est sélectionné comme méthode d'authentification.
Fichier de clé privée	Saisissez le chemin du fichier de clé privée. Ce champ est disponible si « Clé publique » est sélectionné comme méthode d'authentification.
Phrase secrète	Saisissez la phrase secrète. Ce champ est disponible si « Clé publique » est sélectionné comme méthode d'authentification.
Port	Saisissez le port du service SSH distant sur l'hôte cible auquel Zabbix se connectera.
Commandes	Saisissez les commandes. La prise en charge des macros dépend de la portée du script (voir *Portée* ci-dessus). Les macros utilisateur personnalisées sont prises en charge.
Type de script : **Telnet**	
Nom d'utilisateur	Saisissez le nom d'utilisateur.
Mot de passe	Saisissez le mot de passe.
Port	Saisissez le port du service Telnet distant sur l'hôte cible auquel Zabbix se connectera.
Commandes	Saisissez les commandes. La prise en charge des macros dépend de la portée du script (voir *Portée* ci-dessus). Les macros utilisateur personnalisées sont prises en charge.
Type de script : **IPMI**	
Commande	Saisissez la commande IPMI. La prise en charge des macros dépend de la portée du script (voir *Portée* ci-dessus). Les macros utilisateur personnalisées sont prises en charge.
Description	Saisissez une description du script.
Groupe d'hôtes	Sélectionnez le groupe d'hôtes pour lequel le script sera disponible (ou *Tous* pour tous les groupes d'hôtes).
Groupe d'utilisateurs	Sélectionnez le groupe d'utilisateurs auquel le script sera disponible (ou *Tous* pour tous les groupes d'utilisateurs). Ce champ s'affiche uniquement si « Action manuelle sur un hôte » ou « Action manuelle sur un événement » est sélectionné comme *Portée*.
Autorisations requises sur l'hôte	Sélectionnez le niveau d'autorisation pour le groupe d'hôtes : *Lecture* ou *Écriture*. Seuls les utilisateurs disposant du niveau d'autorisation requis pourront exécuter le script. Ce champ s'affiche uniquement si « Action manuelle sur un hôte » ou « Action manuelle sur un événement » est sélectionné comme *Portée*.

Paramètre	Description
Configuration avancée	Cliquez sur l'en-tête *Configuration avancée* pour afficher les options de configuration avancée. Ce champ s'affiche uniquement si « Action manuelle sur un hôte » ou « Action manuelle sur un événement » est sélectionné comme *Portée*.

Configuration avancée

Les options de configuration avancée sont disponibles dans une section réductible *Configuration avancée* :



Paramètre	Description
Activer la saisie utilisateur	Cochez la case pour activer la saisie manuelle de l'utilisateur avant l'exécution du script. La saisie manuelle de l'utilisateur remplacera la valeur de la macro {MANUALINPUT} dans le script. Voir également : Saisie manuelle de l'utilisateur.
Invite de saisie	Saisissez un texte personnalisé invitant l'utilisateur à effectuer une saisie. Ce texte sera affiché au-dessus du champ de saisie dans la fenêtre contextuelle *Saisie manuelle*. Pour afficher un aperçu de la fenêtre contextuelle *Saisie manuelle*, cliquez sur *Tester la saisie utilisateur*. L'aperçu vous permet également de vérifier si la chaîne saisie respecte la règle de validation (voir les paramètres ci-dessous). La prise en charge des macros et des macros utilisateur dépend de la portée du script (voir *Portée* dans les paramètres généraux de configuration du script).
Type de saisie	Sélectionnez le type de saisie manuelle : **Chaîne** - chaîne unique ; **Liste déroulante** - la valeur est sélectionnée parmi plusieurs options de liste déroulante.
Options de la liste déroulante	Saisissez les valeurs uniques de la liste déroulante de saisie utilisateur dans une liste séparée par des virgules. Pour inclure une option vide dans la liste déroulante, ajoutez une virgule supplémentaire au début, au milieu ou à la fin de la liste. Ce champ s'affiche uniquement si *Liste déroulante* est sélectionné comme *Type de saisie*.
Chaîne de saisie par défaut	Saisissez la chaîne par défaut pour la saisie utilisateur (ou aucune). Ce champ sera validé par rapport à l'expression régulière fournie dans le champ *Règle de validation de la saisie*. La valeur saisie ici sera affichée par défaut dans la fenêtre contextuelle *Saisie manuelle*. Ce champ s'affiche uniquement si *Chaîne* est sélectionné comme *Type de saisie*.
Règle de validation de la saisie	Saisissez une expression régulière pour valider la chaîne saisie par l'utilisateur. Les expressions régulières globales sont prises en charge. Ce champ s'affiche uniquement si *Chaîne* est sélectionné comme *Type de saisie*.
Activer la confirmation	Cochez la case pour afficher un message de confirmation avant l'exécution du script. Cette fonctionnalité peut être particulièrement utile pour les opérations potentiellement dangereuses (comme un script de redémarrage) ou celles qui peuvent prendre beaucoup de temps.
Texte de confirmation	Saisissez un texte de confirmation personnalisé pour la fenêtre contextuelle de confirmation activée avec la case ci-dessus (par exemple, *Le système distant va être redémarré. Êtes-vous sûr ?*). Pour voir à quoi ressemblera le texte, cliquez sur *Tester la confirmation* à côté du champ. Les macros et les macros utilisateur personnalisées sont prises en charge. *Remarque :* les macros ne seront pas développées lors du test du message de confirmation.

Si la saisie manuelle de l'utilisateur et un message de confirmation sont tous deux configurés, ils seront affichés dans des fenêtres contextuelles consécutives.

Saisie manuelle par l'utilisateur

La saisie manuelle par l'utilisateur permet de fournir un paramètre personnalisé à chaque exécution du script. Cela évite de devoir créer plusieurs scripts utilisateur similaires ne différant que par un seul paramètre.

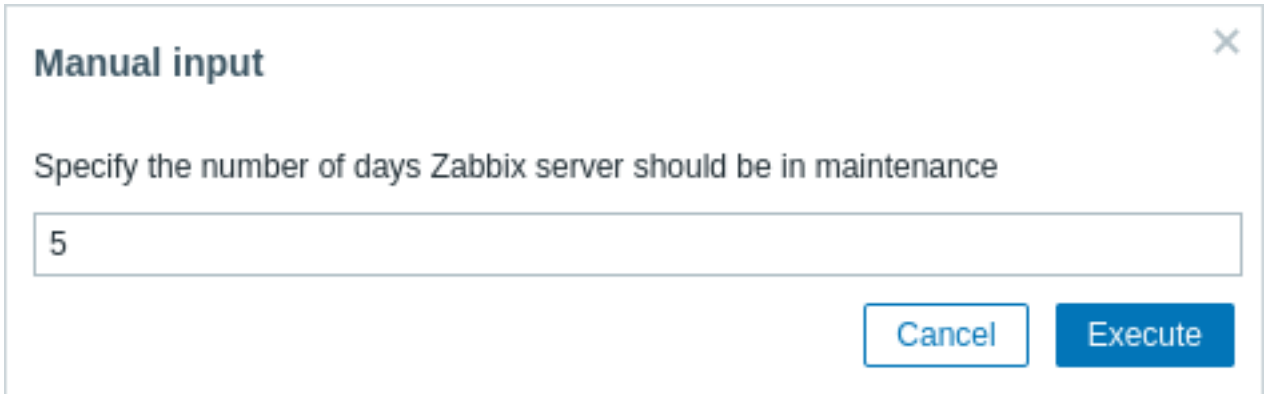
Par exemple, vous pouvez vouloir fournir un entier différent ou une autre adresse URL au script lors de son exécution.

Pour activer la saisie manuelle par l'utilisateur :

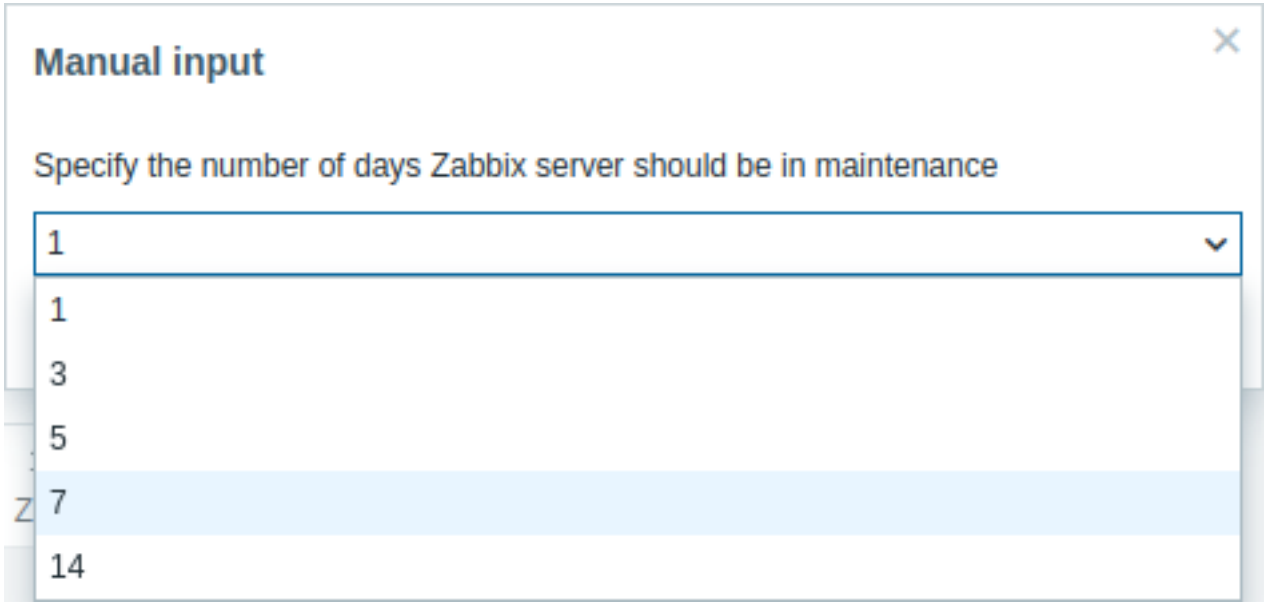
- utilisez la macro {MANUALINPUT} dans le script (commandes, script, paramètre de script) là où cela est nécessaire ; ou dans le champ URL des scripts URL ;
- dans la configuration avancée du script, activez la saisie manuelle par l'utilisateur et configurez les options de saisie.

Lorsque la saisie utilisateur est activée, une fenêtre contextuelle *Saisie manuelle* s'affiche avant l'exécution du script et demande à l'utilisateur de fournir une valeur personnalisée. La valeur fournie remplace {MANUALINPUT} dans le script.

Selon la configuration, l'utilisateur devra saisir une valeur de type chaîne :



Ou sélectionner la valeur dans une liste déroulante d'options prédéfinies :



La saisie manuelle par l'utilisateur est disponible uniquement pour les scripts dont la portée est « Action manuelle sur l'hôte » ou « Action manuelle sur l'événement ».

Exécution du script et résultat

Les scripts exécutés par le serveur Zabbix sont exécutés dans l'ordre décrit sur la page d'exécution des commandes.

Le résultat du script s'affiche dans une fenêtre contextuelle qui apparaît après l'exécution du script. La valeur de retour du script correspond à la sortie standard :

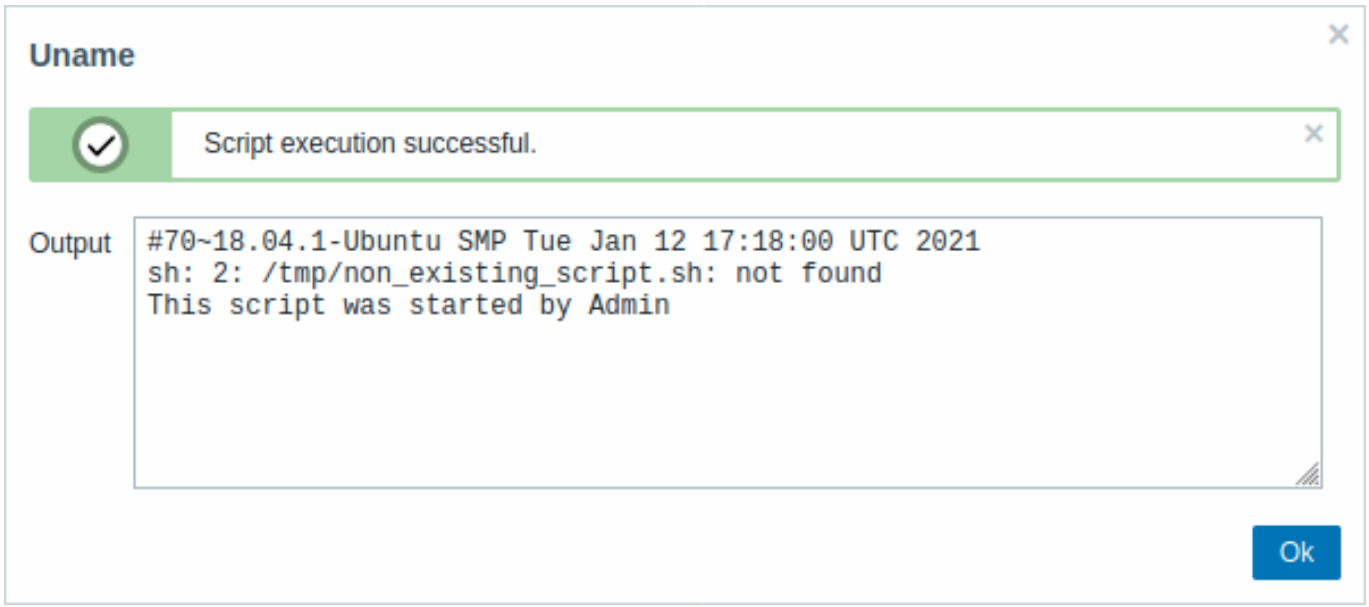
- Si le script se termine avec succès (code de sortie 0), la valeur de retour est limitée à 16 Mo (y compris les espaces de fin qui sont tronqués).
- Si le script se termine avec une erreur (code de sortie non nul), la valeur de retour correspond à la sortie d'erreur standard, limitée à 2 Ko.

Par défaut, Zabbix ne stocke pas la sortie étendue des scripts. Pour conserver les détails complets de la sortie, vous pouvez implémenter une journalisation dans le script lui-même (par exemple, rediriger la sortie vers un fichier journal local).

Notez que pour les scripts exécutés soit sur le serveur Zabbix, soit sur le proxy Zabbix, les limites de la base de données s'appliquent également.

Vous trouverez ci-dessous un exemple de script et la fenêtre de résultat :

```
uname -v
/tmp/non_existing_script.sh
echo "This script was started by {USER.USERNAME}"
```



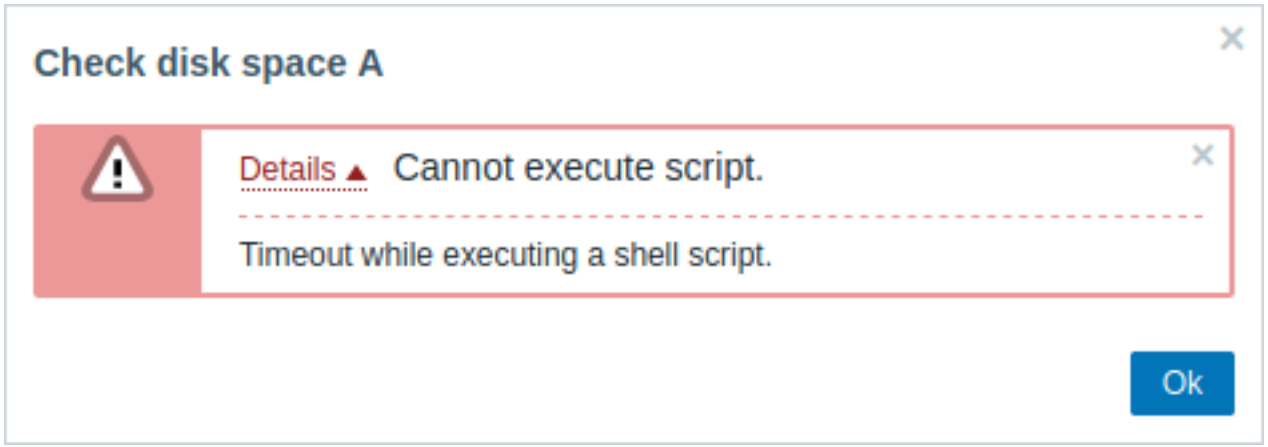
Le résultat du script n'affiche pas le script lui-même.

Délai d'expiration du script sur l'agent Zabbix

Vous pouvez rencontrer une situation dans laquelle un délai d'expiration survient lors de l'exécution d'un script.

Voici un exemple de script exécuté sur l'agent Zabbix, ainsi que la fenêtre de résultat correspondante :

```
sleep 5
df -h
```



Dans ce cas, le message d'erreur est le suivant :

```
Timeout while executing a shell script.
```

Pour éviter ce type de situation, il est recommandé d'optimiser le script lui-même (dans l'exemple ci-dessus, « 5 ») plutôt que d'ajuster le paramètre `Timeout` dans la configuration de l'agent Zabbix et la configuration du serveur Zabbix.

Si le paramètre `Timeout` a été modifié dans la configuration de l'agent Zabbix, le message d'erreur suivant s'affiche :

```
Get value from agent failed: ZBX_TCP_READ() timed out.
```

Cela signifie que la modification a été effectuée dans la configuration de l'agent Zabbix, mais qu'il est également nécessaire de modifier le paramètre `Timeout` dans la configuration du serveur Zabbix.

Délai d'exécution du script sur les agents actifs

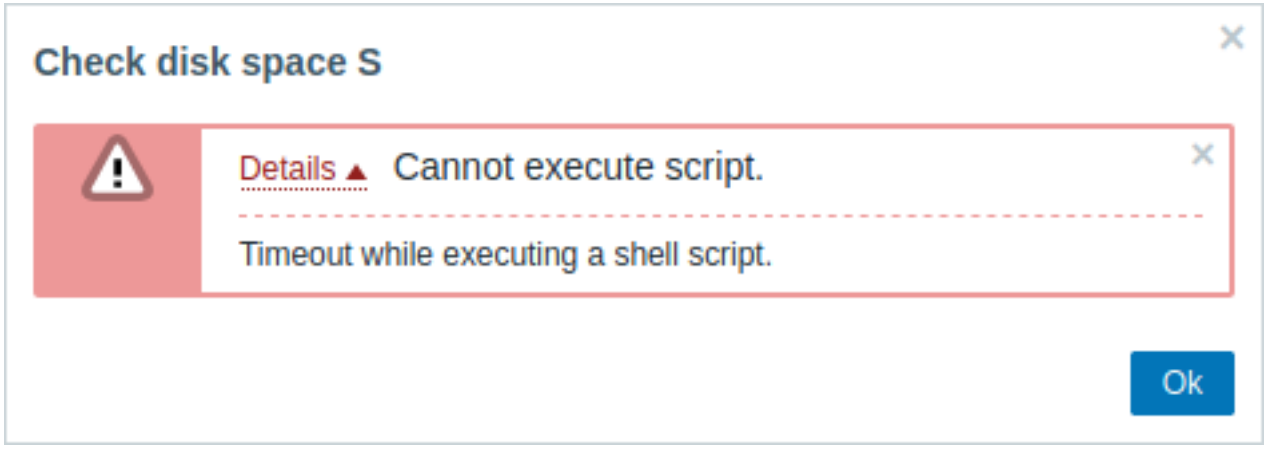
Il est possible d'exécuter des scripts sur des agents fonctionnant uniquement en mode actif. Une fois l'exécution du script déclenchée par une opération d'action ou par une exécution manuelle du script, la commande est incluse dans la configuration de vérification active et exécutée dès que l'agent actif la reçoit.

Les scripts manuels sont envoyés à l'agent actif avec le délai d'attente du serveur/proxy pour l'exécution du script. Veuillez augmenter le délai d'attente par défaut du serveur/proxy pour l'exécution du script. Le délai d'attente doit être supérieur à la fréquence d'actualisation des vérifications actives, sinon le délai est dépassé avant que l'agent actif ne reçoive le script et ne puisse renvoyer le résultat.

Délai d'expiration du script sur le serveur/proxy Zabbix

Voici un exemple de script exécuté sur le serveur Zabbix, ainsi que la fenêtre de résultats ci-dessous :

```
sleep 11
df -h
```



Il est également recommandé d'optimiser le script lui-même (plutôt que d'ajuster le paramètre `TrapperTimeout` à une valeur correspondante (dans notre cas, > 11) en modifiant la configuration du serveur Zabbix).

Utilisateurs

Aperçu

Ce menu comprend des sections liées à la configuration des utilisateurs dans Zabbix. Ce menu est disponible uniquement pour les utilisateurs du type SuperAdmin.

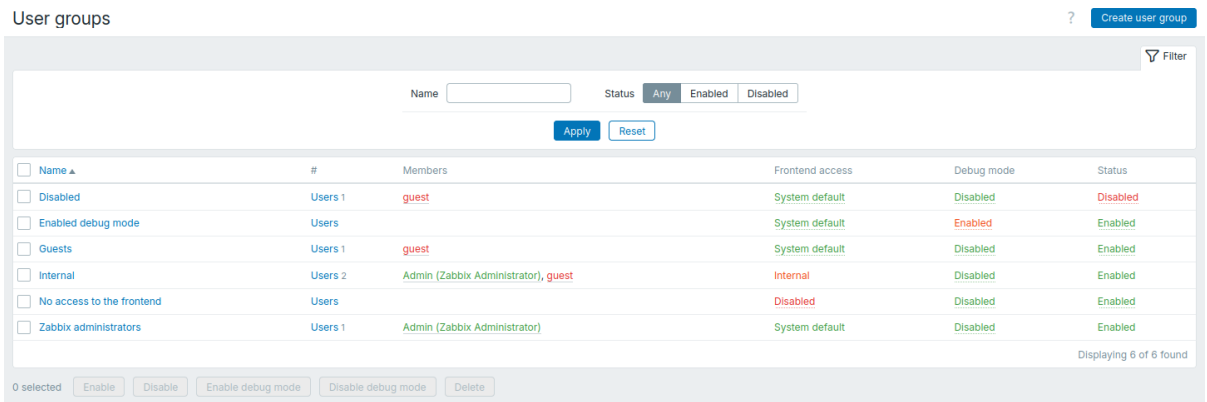
Groupes d'utilisateurs

Vue d'ensemble

Dans la section *Users* > *User groups*, les groupes d'utilisateurs du système sont gérés.

Groupes d'utilisateurs

La liste des groupes d'utilisateurs existants avec leurs détails s'affiche.



Données affichées :

Colonne	Description
Nom	Nom du groupe d'utilisateurs. Cliquez sur le nom du groupe d'utilisateurs pour ouvrir le formulaire de configuration du groupe d'utilisateurs.
#	Nombre d'utilisateurs dans le groupe. Cliquez sur *Utilisateurs* pour afficher les utilisateurs correspondants filtrés dans la liste des utilisateurs.
Membres	Noms d'utilisateur des utilisateurs individuels du groupe d'utilisateurs (avec le nom et le prénom entre parenthèses). Cliquez sur le nom d'utilisateur pour ouvrir le formulaire de configuration de l'utilisateur. Les utilisateurs appartenant à des groupes désactivés s'affichent en rouge.
Accès à l'interface	Le niveau d'accès à l'interface s'affiche : **Valeur par défaut du système** - les utilisateurs sont authentifiés par Zabbix, LDAP ou HTTP (selon la méthode d'authentification définie globalement) ; **Interne** - les utilisateurs sont authentifiés par Zabbix ; ignoré si l'authentification HTTP est la valeur par défaut globale ; **LDAP** - les utilisateurs sont authentifiés par LDAP ; ignoré si l'authentification HTTP est la valeur par défaut globale ; **Désactivé** - l'accès à l'interface Zabbix est interdit pour ce groupe. Vous pouvez modifier le niveau en cliquant sur le niveau actuel.
Mode débogage	L'état du mode débogage s'affiche - *Activé* ou *Désactivé*. Vous pouvez le modifier en cliquant sur l'état.
État	L'état du groupe d'utilisateurs s'affiche - *Activé* ou *Désactivé*. Vous pouvez le modifier en cliquant sur l'état.

Pour configurer un nouveau groupe d'utilisateurs, cliquez sur le bouton *Créer un groupe d'utilisateurs* dans le coin supérieur droit.

Options de modification en masse

Les boutons sous la liste offrent plusieurs options de modification en masse :

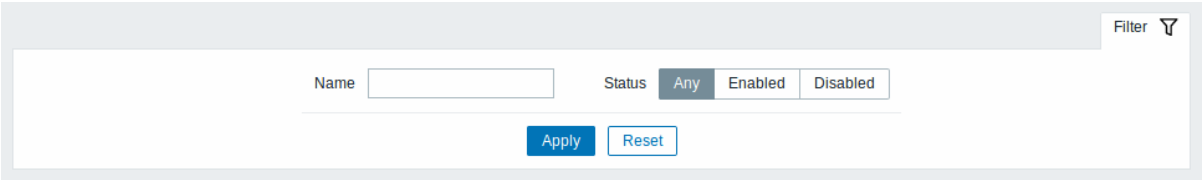
- *Activer* - changer l'état du groupe d'utilisateurs en *Activé*
- *Désactiver* - changer l'état du groupe d'utilisateurs en *Désactivé*
- *Activer le mode de débogage* - activer le mode de débogage pour les groupes d'utilisateurs
- *Désactiver le mode de débogage* - désactiver le mode de débogage pour les groupes d'utilisateurs
- *Supprimer* - supprimer les groupes d'utilisateurs

Pour utiliser ces options, cochez les cases devant les groupes d'utilisateurs concernés, puis cliquez sur le bouton requis.

Utilisation du filtre

Vous pouvez utiliser le filtre pour afficher uniquement les groupes d'utilisateurs qui vous intéressent. Pour de meilleures performances de recherche, les données sont recherchées avec les macros non résolues.

Le lien *Filter* est disponible au-dessus de la liste des groupes d'utilisateurs. Si vous cliquez dessus, un filtre devient disponible et vous pouvez filtrer les groupes d'utilisateurs par nom et par statut.



Rôles d'utilisateur

Présentation

Dans la section *Utilisateurs* > *Rôles utilisateur*, vous pouvez créer des rôles utilisateur.

Les rôles utilisateur vous permettent de créer des autorisations précises en fonction du type d'utilisateur sélectionné initialement (*Utilisateur*, *Administrateur*, *Super administrateur*).

Après la sélection d'un type d'utilisateur, toutes les autorisations disponibles pour ce type d'utilisateur sont accordées (cochées par défaut).

Les autorisations peuvent uniquement être révoquées parmi celles disponibles pour le type d'utilisateur ; elles ne peuvent pas être étendues au-delà de ce qui est disponible pour ce type d'utilisateur.

Les cases à cocher correspondant aux autorisations indisponibles sont grisées ; les utilisateurs ne pourront pas accéder à l'élément, même en saisissant directement dans le navigateur l'URL de cet élément.

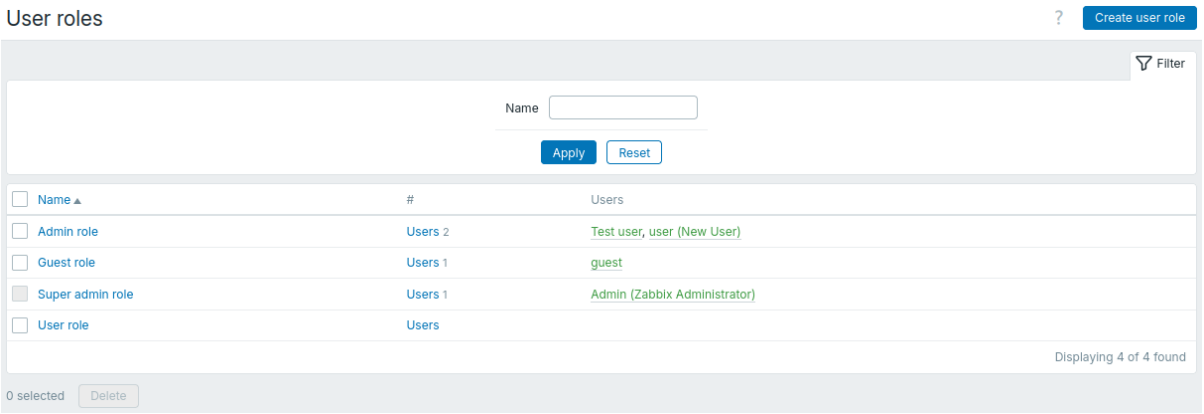
Note:
La restriction de l'accès à certains éléments de l'interface empêche uniquement l'ouverture de cette page ; elle ne supprime pas la possibilité d'accéder aux données sous-jacentes depuis d'autres parties de l'interface.

Les rôles utilisateur peuvent être attribués aux utilisateurs du système. Chaque utilisateur ne peut avoir qu'un seul rôle attribué.

Rôles utilisateur par défaut

Par défaut, Zabbix est configuré avec quatre rôles utilisateur, qui disposent d'un ensemble prédéfini d'autorisations :

- Rôle Invité
- Rôle Utilisateur
- Rôle Administrateur
- Rôle Super administrateur



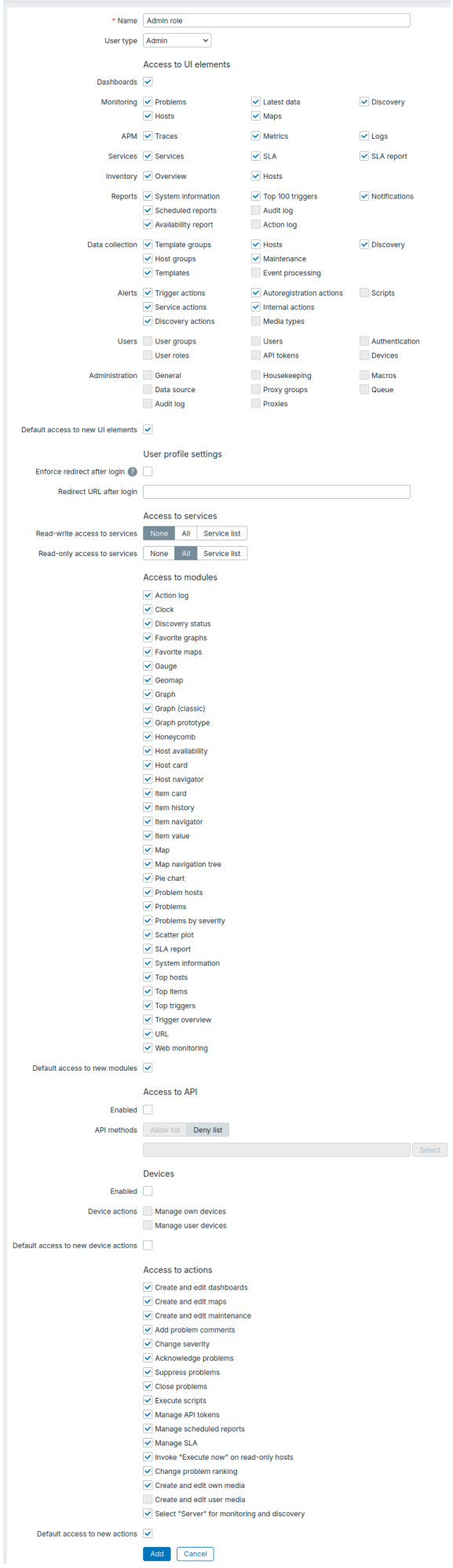
Ces rôles sont basés sur les principaux types d'utilisateurs dans Zabbix. La liste de tous les utilisateurs auxquels le rôle correspondant est attribué est affichée. Les utilisateurs inclus dans des groupes désactivés sont indiqués en rouge. Le *rôle Invité* est un rôle de type utilisateur disposant uniquement des autorisations nécessaires pour afficher certaines sections de l'interface.

Note:
Le *rôle Super administrateur* par défaut ne peut pas être modifié ni supprimé, car au moins un utilisateur Super administrateur disposant de privilèges illimités doit exister dans Zabbix. Les utilisateurs de type *Super administrateur* peuvent modifier les paramètres de leur propre rôle, mais pas le type d'utilisateur.

Configuration

Pour créer un nouveau rôle, cliquez sur le bouton *Create user role* dans le coin supérieur droit. Pour mettre à jour un rôle existant, cliquez sur son nom pour ouvrir le formulaire de configuration.

User roles


* Name
Admin role
User type
Admin
Access to UI elements
Dashboards
Monitoring
Problems
Latest data
Discovery
Hosts
Maps
APM
Traces
Metrics
Logs
Services
Services
SLA
SLA report
Inventory
Overview
Hosts
Reports
System information
Top 100 triggers
Notifications
Scheduled reports
Audit log
Availability report
Action log
Data collection
Template groups
Hosts
Discovery
Host groups
Maintenance
Templates
Event processing
Alerts
Trigger actions
Autoregistration actions
Scripts
Service actions
Internal actions
Discovery actions
Media types
Users
User groups
Users
Authentication
User roles
API tokens
Devices
Administration
General
Housekeeping
Macros
Data source
Proxy groups
Queue
Audit log
Proxies
Default access to new UI elements
User profile settings
Enforce redirect after login
Redirect URL after login
Access to services
Read-write access to services
None
All
Service list
Read-only access to services
None
All
Service list
Access to modules
Action log
Clock
Discovery status
Favorite graphs
Favorite maps
Gauge
Geomap
Graph
Graph (classic)
Graph prototype
Honeycomb
Host availability
Host card
Host navigator
Item card
Item history
Item navigator
Item value
Map
Map navigation tree
Pie chart
Problem hosts
Problems
Problems by severity
Scatter plot
SLA report
System information
Top hosts
Top items
Top triggers
Trigger overview
URL
Web monitoring
Default access to new modules
Access to API
Enabled
API methods
Allow list
Deny list
Select
Devices
Enabled
Device actions
Manage own devices
Manage user devices
Default access to new device actions
Access to actions
Create and edit dashboards
Create and edit maps
Create and edit maintenance
Add problem comments
Change severity
Acknowledge problems
Suppress problems
Close problems
Execute scripts
Manage API tokens
Manage scheduled reports
Manage SLA
Invoke "Execute now" on read-only hosts
Change problem ranking
Create and edit own media
Create and edit user media
Select "Server" for monitoring and discovery
Default access to new actions
Add
Cancel

Les autorisations disponibles sont affichées. Pour révoquer une autorisation donnée, décochez sa case.

Les autorisations disponibles ainsi que les valeurs par défaut de chaque rôle utilisateur préconfiguré dans Zabbix sont décrites ci-dessous.

Autorisations par défaut

Accès aux éléments de l'interface

L'accès par défaut aux sections de menu dépend du type d'utilisateur. Consultez la page Autorisations pour plus de détails.

Accès aux autres options

Certaines sections décrites dans le tableau ci-dessous peuvent être masquées :

- *Accès aux modules* est masqué si `$ZBX_FEATURE_FLAGS['modules_config_enabled']` est défini sur `false` dans le fichier de configuration de l'interface.
- *Périphériques* est masqué si `EnableMobileDevices` est désactivé.

Paramètre	Description	Rôles utilisateur par défaut *Rôle de super administrateur*	*Rôle d'administrateur*	*Rôle d'utilisateur*	*Rôle invité*
Accès par défaut aux nouveaux éléments de l'interface	Cette option indique comment les nouvelles sections de menu seront accessibles après une mise à niveau de Zabbix. Les sections de menu existantes des modules ne sont pas affectées.	Oui	Oui	Oui	Oui
Paramètres du profil utilisateur					
Forcer la redirection après la connexion	Cochez cette case pour limiter les utilisateurs à l'URL de redirection après la connexion. Si elle n'est pas cochée, les utilisateurs peuvent définir leur URL personnelle après la connexion dans leur profil ; l'URL de redirection après la connexion est utilisée uniquement comme solution de secours.	Oui	Non	Non	Non
URL de redirection après la connexion	Indiquez l'URL de redirection après la connexion.				
Accès aux services					

Accès en lecture-écriture aux services	Sélectionnez l'accès en lecture-écriture aux services : **Aucun** - aucun accès **Tous** - accès en lecture-écriture à tous les services **Liste de services** - sélectionnez les services auxquels accorder un accès en lecture-écriture L'accès en lecture-écriture, s'il est accordé, est prioritaire sur les paramètres d'accès en lecture seule et est hérité dynamiquement par les services enfants.	Tous	Tous	Aucun	Aucun
Accès en lecture-écriture aux services avec une étiquette	Indiquez le nom de l'étiquette et, éventuellement, sa valeur pour accorder en plus un accès en lecture-écriture aux services correspondant à l'étiquette. Cette option est disponible si « Liste de services » est sélectionné dans le paramètre *Accès en lecture-écriture aux services*. L'accès en lecture-écriture, s'il est accordé, est prioritaire sur les paramètres d'accès en lecture seule et est hérité dynamiquement par les services enfants.				
Accès en lecture seule aux services	Sélectionnez l'accès en lecture seule aux services : **Aucun** - aucun accès **Tous** - accès en lecture seule à tous les services **Liste de services** - sélectionnez les services auxquels accorder un accès en lecture seule L'accès en lecture seule n'est pas prioritaire sur l'accès en lecture-écriture et est hérité dynamiquement par les services enfants.			Tous	Tous
Accès en lecture seule aux services avec une étiquette	Indiquez le nom de l'étiquette et, éventuellement, sa valeur pour accorder en plus un accès en lecture seule aux services correspondant à l'étiquette. Cette option est disponible si « Liste de services » est sélectionné dans le paramètre *Accès en lecture seule aux services*. L'accès en lecture seule n'est pas prioritaire sur l'accès en lecture-écriture et est hérité dynamiquement par les services enfants.				
Accès aux modules					
<Nom du module>	Autoriser ou refuser l'accès à un module spécifique. Seuls les modules activés sont affichés dans cette section. Il n'est pas possible d'accorder ou de restreindre l'accès à un module actuellement désactivé.	Oui	Oui	Oui	Oui
Accès par défaut aux nouveaux modules	Cette option indique comment les nouveaux modules et widgets seront accessibles après une mise à niveau de Zabbix. Elle s'applique également aux modules et widgets ajoutés dans la section *Administration > Général > Modules*.				
Accès à l'API					
Activé	Activer/désactiver l'accès à l'API.	Oui	Non	Non	Non

Méthodes d'API	Sélectionnez *Liste d'autorisation* pour autoriser, ou *Liste de refus* pour refuser, les méthodes d'API indiquées dans le champ de recherche. Notez qu'il n'est pas possible d'autoriser certaines méthodes d'API et d'en refuser d'autres. Dans le champ de recherche, commencez à saisir le nom de la méthode, puis sélectionnez-la dans la liste de saisie semi-automatique. Vous pouvez également cliquer sur le bouton *Sélectionner* et sélectionner des méthodes dans la liste complète disponible pour ce type d'utilisateur. Notez que si une action donnée du bloc Accès aux actions n'est pas cochée, les utilisateurs ne pourront pas utiliser les méthodes d'API associées à cette action. Les caractères génériques sont pris en charge. Exemples : `dashboard.*` (toutes les méthodes du service d'API « dashboard. »), `*` (toute méthode), `*.export` (méthodes dont le nom contient « .export » dans tous les services d'API). Si aucune méthode n'est indiquée, la *Liste d'autorisation* refuse l'accès à toutes les méthodes d'API, tandis que la *Liste de refus* autorise l'accès à toutes les méthodes d'API disponibles pour le type d'utilisateur. Pour autoriser toutes les méthodes avec la *Liste d'autorisation*, indiquez `*`. Pour bloquer complètement l'accès à l'API, décochez la case *Activé* au lieu de laisser la *Liste d'autorisation* vide. La liste des méthodes d'API ne s'applique pas aux méthodes qui ne nécessitent pas d'authentification, telles que `apiinfo.version` et `user.login`.				
Accès aux actions					
Créer et modifier des tableaux de bord	Décocher cette case révoque également les droits d'utiliser les méthodes d'API `.create`, `.update` et `.delete` pour les éléments correspondants.	Oui	Oui	Oui	Non
Créer et modifier des cartes					
Créer et modifier la maintenance				Non	
Ajouter des commentaires aux problèmes	Décocher cette case révoque également les droits d'effectuer l'action correspondante via la méthode d'API `event.acknowledge`.			Oui	
Modifier la gravité					
Accuser réception des problèmes					
Supprimer les problèmes					
Fermer les problèmes					
Exécuter des scripts	Décocher cette case révoque également les droits d'utiliser la méthode d'API `script.execute`.				
Gérer les jetons d'API	Décocher cette case révoque également les droits d'utiliser toutes les méthodes d'API `token.*`.				
Gérer les rapports planifiés	Décocher cette case révoque également les droits d'utiliser toutes les méthodes d'API `report.*`.			Non	

Gérer les SLA	Activer/désactiver les droits de gérer les SLA.				
Appeler « Exécuter maintenant » sur les hôtes en lecture seule	Autoriser les utilisateurs à utiliser l'option « Exécuter maintenant » dans Dernières données pour les éléments des hôtes en lecture seule.			Oui	
Modifier le classement des problèmes	Autoriser les utilisateurs à modifier le classement d'un problème, de cause à symptôme et inversement.				
Créer et modifier ses propres médias	Autoriser les utilisateurs à créer et modifier leurs propres médias.				
Créer et modifier les médias des utilisateurs	Autoriser les utilisateurs à créer et modifier les médias d'autres utilisateurs. Cette option est disponible uniquement pour les utilisateurs super administrateurs.		Non	Non	
Sélectionner « Serveur » pour la supervision et la découverte	Décocher cette case restreint la création d'hôtes supervisés par le serveur et de règles de découverte réseau exécutées par le serveur, ainsi que le test d'éléments avec le serveur. L'option *Serveur* sera désactivée dans la configuration de l'hôte pour le paramètre *Supervisé par*, dans la configuration de la règle de découverte réseau pour le paramètre *Découverte par*, et dans le test d'éléments pour le paramètre *Tester avec*.		Oui	Non	
Accès par défaut aux nouvelles actions	Cette option indique comment les nouvelles actions seront accessibles après une mise à niveau de Zabbix.		Oui	Oui	
Périphériques					
Activé	Activer/désactiver l'accès à *Paramètres utilisateur* > *Périphériques* (accessible à tous les utilisateurs) et *Utilisateurs* > *Périphériques* (accessible uniquement aux super administrateurs).	Oui	Non	Non	Non
Gérer ses propres périphériques	Autoriser les utilisateurs à gérer leurs propres périphériques dans *Paramètres utilisateur* > *Périphériques*.				
Gérer les périphériques des utilisateurs	Autoriser les utilisateurs à gérer les périphériques appartenant à d'autres utilisateurs dans *Utilisateurs* > *Périphériques*.				
Accès par défaut aux nouvelles actions	Cette option indique comment les nouvelles actions seront accessibles après une mise à niveau de Zabbix.				

Voir également :

- Configurer un utilisateur

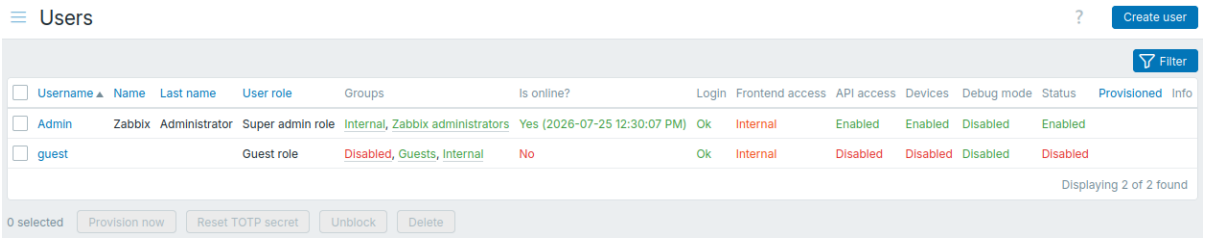
Utilisateurs

Vue d'ensemble

La section *Utilisateurs* > *Utilisateurs* permet de gérer les utilisateurs du système.

Utilisateurs

Une liste des utilisateurs existants avec leurs détails s'affiche.



Données affichées :

Colonne	Description
Nom d'utilisateur	Nom d'utilisateur utilisé pour se connecter à Zabbix. Cliquer sur le nom d'utilisateur ouvre le formulaire de configuration de l'utilisateur.
Prénom	Prénom de l'utilisateur.
Nom	Nom de famille de l'utilisateur.
Rôle utilisateur	Le rôle utilisateur est affiché.
Groupes	Les groupes dont l'utilisateur est membre sont répertoriés. Cliquer sur le nom du groupe d'utilisateurs ouvre le formulaire de configuration du groupe d'utilisateurs. Les groupes désactivés sont affichés en rouge.
En ligne ?	L'état en ligne de l'utilisateur est affiché - *Oui* ou *Non*. L'heure de la dernière activité de l'utilisateur est affichée entre parenthèses.
Connexion	L'état de connexion de l'utilisateur est affiché - *OK* ou *Bloqué*. Un utilisateur peut être temporairement bloqué après avoir dépassé le nombre de tentatives de connexion infructueuses défini dans la section *Administration > Général > Autres* (cinq par défaut). Cliquer sur *Bloqué* permet de débloquer l'utilisateur.
Accès à l'interface	Le niveau d'accès à l'interface est affiché - *Par défaut du système*, *Interne*, *LDAP* ou *Désactivé*, selon celui défini pour l'ensemble du groupe d'utilisateurs.
Accès à l'API	L'état d'accès à l'API est affiché - *Activé* ou *Désactivé*, selon celui défini pour le rôle utilisateur. Si l'accès à l'API est activé, le survol de *Activé* avec la souris affiche les méthodes API du rôle utilisateur sous *Méthodes autorisées* ou *Méthodes refusées*, selon le mode des méthodes API. *Aucune* s'affiche si aucune méthode n'est spécifiée.
Appareils	L'état d'accès aux appareils est affiché - *Activé* ou *Désactivé*, selon celui défini pour le rôle utilisateur. Cette colonne s'affiche si `EnableMobileDevices` est activé.
Mode débogage	L'état du mode débogage est affiché - *Activé* ou *Désactivé*, selon celui défini pour l'ensemble du groupe d'utilisateurs.
État	L'état de l'utilisateur est affiché - *Activé* ou *Désactivé*, selon celui défini pour l'ensemble du groupe d'utilisateurs.
Provisionné	La date à laquelle l'utilisateur a été provisionné pour la dernière fois est affichée. Utilisé pour les utilisateurs créés par le provisionnement JIT depuis LDAP/SAML.
Informations	Les informations sur les erreurs sont affichées. Un avertissement jaune s'affiche pour les utilisateurs sans groupes d'utilisateurs. Un avertissement rouge s'affiche pour les utilisateurs sans rôles, ainsi que pour les utilisateurs sans rôles ni groupes d'utilisateurs.

Pour configurer un nouvel utilisateur, cliquez sur le bouton *Créer un utilisateur* dans le coin supérieur droit.

Options de modification en masse

Les boutons situés sous la liste proposent plusieurs options de modification en masse :

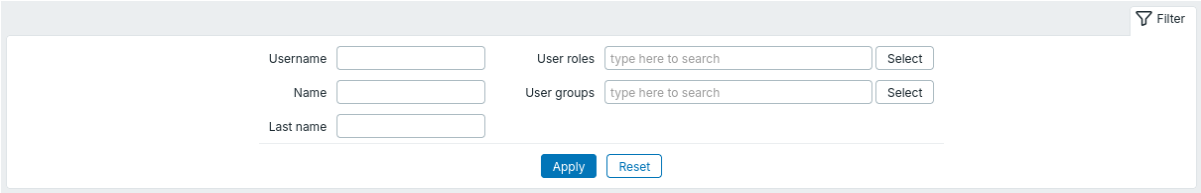
- *Provisionner maintenant* - mettre à jour les informations utilisateur depuis LDAP (cette option est activée uniquement si un utilisateur LDAP est sélectionné)
- *Réinitialiser le secret TOTP* - réinitialiser les secrets TOTP des utilisateurs pour toutes les méthodes TOTP et supprimer la session utilisateur (cette option est activée uniquement si MFA est activée ; pour les utilisateurs sans secret TOTP, leur session ne sera pas supprimée)
- *Débloquer* - réactiver l'accès au système pour les utilisateurs bloqués
- *Supprimer* - supprimer les utilisateurs

Pour utiliser ces options, cochez les cases devant les utilisateurs concernés, puis cliquez sur le bouton requis.

Utilisation du filtre

Vous pouvez utiliser le filtre pour afficher uniquement les utilisateurs qui vous intéressent. Pour de meilleures performances de recherche, les données sont recherchées avec les macros non résolues.

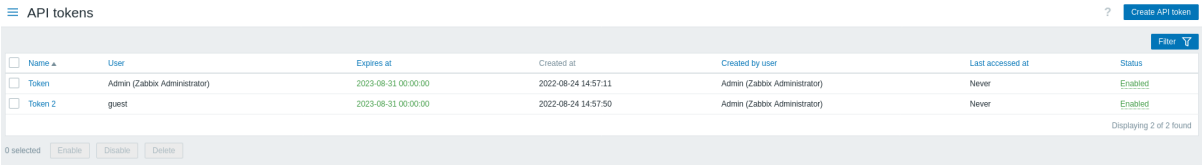
Le lien *Filtre* est disponible au-dessus de la liste des utilisateurs. Si vous cliquez dessus, un filtre devient disponible et vous pouvez filtrer les utilisateurs par nom d'utilisateur, prénom, nom, rôle utilisateur et groupe d'utilisateurs.



Jetons API

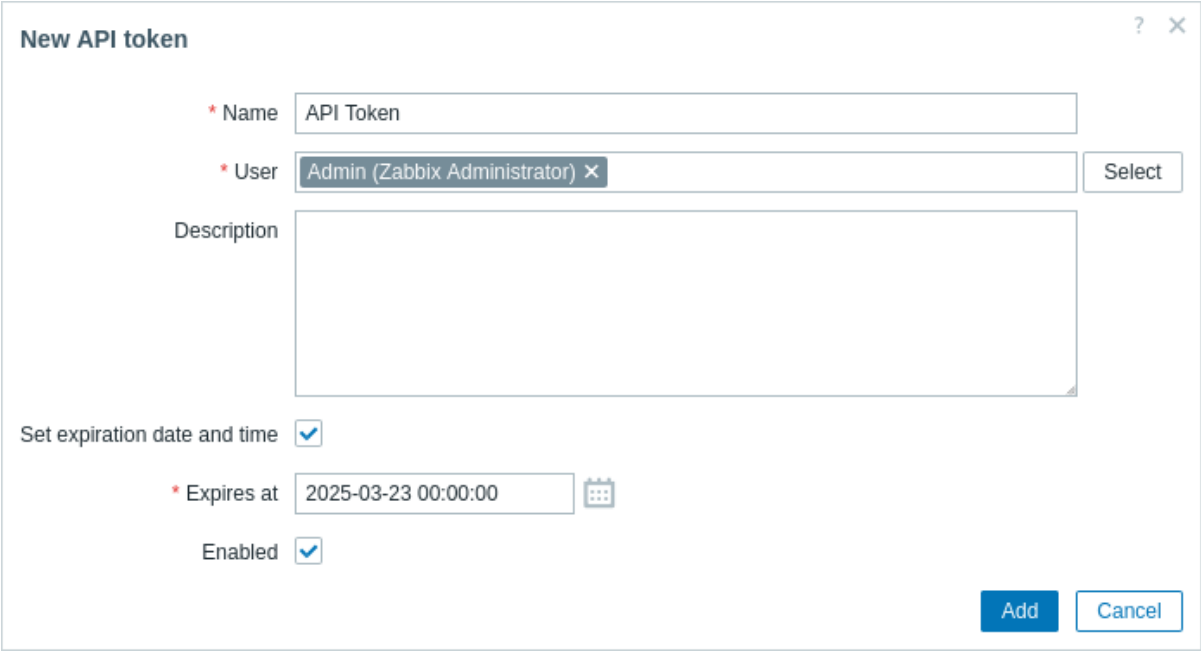
Présentation générale

Cette section vous permet de créer et de gérer des jetons d'API.
 Les jetons d'API sont utilisés pour l'authentification dans l'API Zabbix.
 Chaque jeton d'API est attribué à un seul utilisateur et hérite des permissions de cet utilisateur. Les requêtes authentifiées avec un jeton d'API sont autorisées selon le rôle et les permissions de l'utilisateur associé.



Vous pouvez filtrer les jetons d'API par nom, par utilisateurs auxquels les jetons sont attribués, par date d'expiration, par utilisateurs ayant créé les jetons ou par statut (activé/désactivé). Cliquez sur le statut du jeton dans la liste pour activer/désactiver rapidement un jeton. Vous pouvez également activer/désactiver plusieurs jetons à la fois en les sélectionnant dans la liste, puis en cliquant sur les boutons *Activer/Désactiver* situés sous la liste.

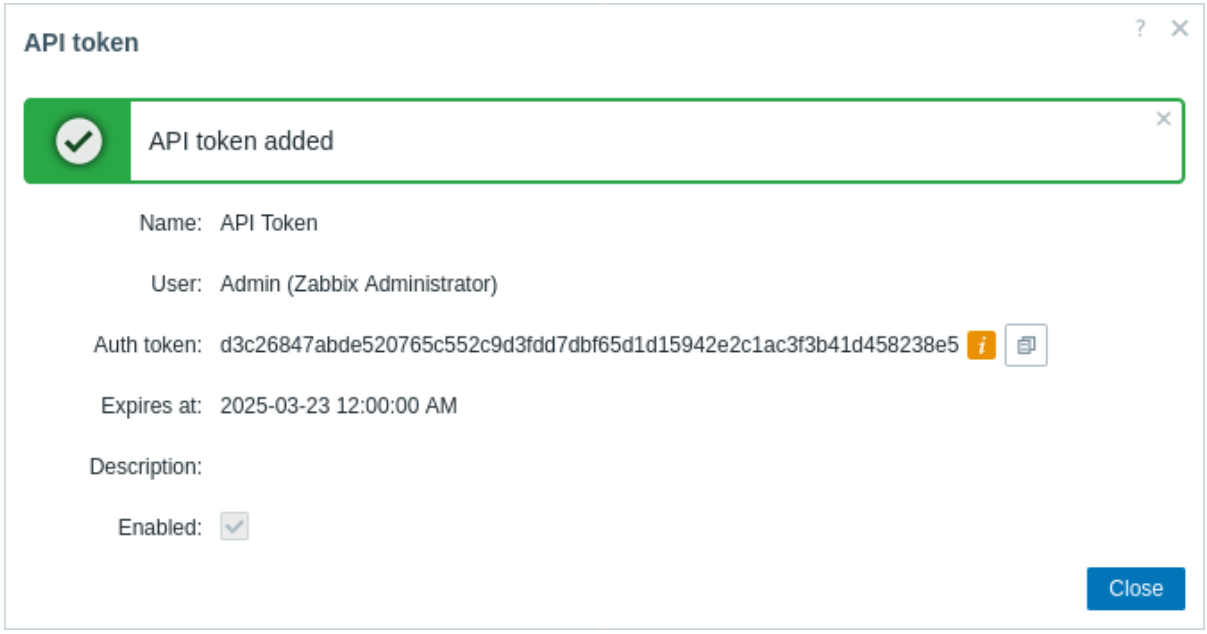
Pour créer un nouveau jeton, cliquez sur le bouton *Créer un jeton d'API* dans le coin supérieur droit, puis renseignez les champs obligatoires dans l'écran de configuration du jeton :



Paramètre	Description
Nom	Nom visible du jeton.
Utilisateur	Utilisateur auquel le jeton doit être attribué. Pour sélectionner rapidement un utilisateur, commencez à saisir son nom d'utilisateur, son prénom ou son nom, puis sélectionnez l'utilisateur requis dans la liste de saisie semi-automatique. Vous pouvez également cliquer sur le bouton *Sélectionner* et sélectionner un utilisateur dans la liste complète des utilisateurs. Un jeton ne peut être attribué qu'à un seul utilisateur.
Description	Description facultative du jeton.
Définir la date et l'heure d'expiration	Décochez cette case si le jeton ne doit pas avoir de date d'expiration.

Paramètre	Description
Date d'expiration	Cliquez sur l'icône du calendrier pour sélectionner la date d'expiration du jeton ou saisissez-la manuellement au format YYYY-MM-DD hh:mm:ss
Activé	Décochez cette case si vous devez créer un jeton désactivé.

Cliquez sur *Ajouter* pour créer un jeton.



Copiez la valeur du *jeton d'authentification* et enregistrez-la dans un emplacement sûr **avant de fermer la page**, puis cliquez sur *Fermer*. Le jeton apparaîtra dans la liste.

Warning:
La valeur du *jeton d'authentification* ne pourra plus être consultée ultérieurement. Elle n'est disponible qu'immédiatement après la création d'un jeton. Si vous perdez un jeton enregistré, vous devrez le régénérer, ce qui créera une nouvelle chaîne d'autorisation.

Cliquez sur le nom du jeton pour modifier le nom, la description, les paramètres de date d'expiration ou le statut du jeton. Notez qu'il n'est pas possible de modifier l'utilisateur auquel le jeton est attribué. Cliquez sur le bouton *Mettre à jour* pour enregistrer les modifications. Si un jeton a été perdu ou exposé, vous pouvez cliquer sur le bouton *Régénérer* pour générer une nouvelle valeur de jeton. Une boîte de dialogue de confirmation apparaîtra et vous demandera de confirmer cette opération, car après sa validation, le jeton précédemment généré deviendra invalide.

Les utilisateurs n'ayant pas accès à la section de menu *Administration* peuvent consulter et modifier les détails des jetons qui leur sont attribués dans la section *Profil utilisateur > Jetons d'API* uniquement si l'option *Gérer les jetons d'API* est autorisée dans les permissions de leur rôle utilisateur.

Authentification

Vue d'ensemble

La section *Utilisateurs > Authentification* vous permet de spécifier la méthode d'authentification des utilisateurs pour Zabbix ainsi que les exigences relatives aux mots de passe internes.

Les méthodes d'authentification disponibles sont l'authentification interne, HTTP, LDAP, SAML et MFA.

Authentification par défaut

Par défaut, Zabbix utilise l'authentification **interne** de Zabbix pour tous les utilisateurs.

Il est possible de modifier la méthode d'authentification par défaut pour utiliser le système LDAP à l'échelle du système. Pour ce faire, accédez à l'onglet *LDAP* et configurez les paramètres LDAP, puis revenez à l'onglet *Authentication* et sélectionnez LDAP dans le sélecteur *Default authentication*.

Notez que la méthode d'authentification peut être affinée au niveau du groupe d'utilisateurs. Même si l'authentification LDAP est définie globalement, certains groupes d'utilisateurs peuvent toujours être authentifiés par Zabbix. L'accès à l'interface de ces groupes doit être défini sur Internal.

Il est également possible d'activer l'authentification LDAP uniquement pour certains groupes d'utilisateurs si l'authentification interne est utilisée globalement. Dans ce cas, les détails de l'authentification LDAP peuvent être spécifiés et utilisés pour certains groupes d'utilisateurs, dont l'accès à l'interface doit alors être défini sur LDAP. Si un utilisateur appartient à au moins un groupe d'utilisateurs utilisant l'authentification LDAP, il ne pourra pas utiliser la méthode d'authentification interne.

Les méthodes d'authentification HTTP, SAML 2.0 et MFA peuvent être utilisées en complément de la méthode d'authentification par défaut.

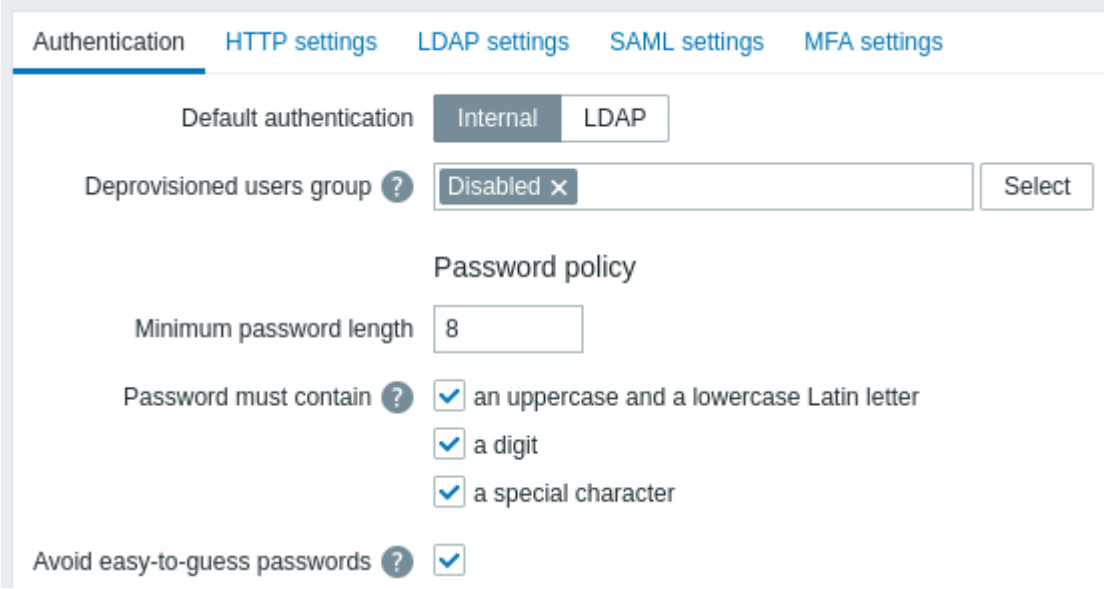
Zabbix prend en charge le provisionnement juste-à-temps (JIT), qui permet de créer des comptes utilisateur dans Zabbix lors de la première authentification d'un utilisateur externe et de provisionner ces comptes utilisateur. Le provisionnement JIT est pris en charge pour LDAP et SAML.

Voir également :

- Authentification HTTP
- Authentification LDAP
- Authentification SAML
- Authentification MFA

Configuration

L'onglet *Authentification* permet de définir la méthode d'authentification par défaut, de spécifier un groupe pour les utilisateurs déprovisionnés et de définir les exigences de complexité des mots de passe pour les utilisateurs Zabbix.



Paramètres de configuration :

Paramètre	Description
Authentification par défaut	Sélectionnez la méthode d'authentification par défaut pour Zabbix : *Interne* ou *LDAP*.
Groupe des utilisateurs déprovisionnés	Spécifiez un groupe d'utilisateurs pour les utilisateurs déprovisionnés. Ce paramètre est requis uniquement pour le provisionnement JIT, concernant les utilisateurs qui ont été créés dans Zabbix à partir de systèmes LDAP ou SAML, mais qui ne doivent plus être provisionnés. Un groupe d'utilisateurs désactivé doit être spécifié.
Longueur minimale du mot de passe	Par défaut, la longueur minimale du mot de passe est définie sur 8. Plage prise en charge : 1-70. Notez que les mots de passe de plus de 72 caractères seront tronqués.
Le mot de passe doit contenir	Cochez une ou plusieurs cases pour exiger l'utilisation des caractères spécifiés dans un mot de passe : - une lettre latine majuscule et une lettre latine minuscule - un chiffre - un caractère spécial Survolez le point d'interrogation pour afficher une indication contenant la liste des caractères correspondant à chaque option.

Paramètre	Description
Éviter les mots de passe faciles à deviner	Si cette option est cochée, un mot de passe sera vérifié selon les exigences suivantes : - ne doit pas contenir le nom, le prénom ou le nom d'utilisateur de l'utilisateur - ne doit pas faire partie des mots de passe courants ou spécifiques au contexte. La liste des mots de passe courants et spécifiques au contexte est générée automatiquement à partir de la liste « Top 100k passwords » du NCSC, de la liste « Top 1M passwords » de SecLists et de la liste des mots de passe spécifiques au contexte de Zabbix. Les utilisateurs internes ne pourront pas définir de mots de passe inclus dans cette liste, car ces mots de passe sont considérés comme faibles en raison de leur utilisation courante.

Les modifications des exigences de complexité des mots de passe n'affecteront pas les mots de passe existants des utilisateurs, mais si un utilisateur existant choisit de modifier son mot de passe, le nouveau mot de passe devra respecter les exigences actuelles. Une indication contenant la liste des exigences sera affichée à côté du champ *Mot de passe* dans le profil utilisateur et dans le formulaire de configuration utilisateur accessible depuis le menu *Utilisateurs* > *Utilisateurs*.

HTTP

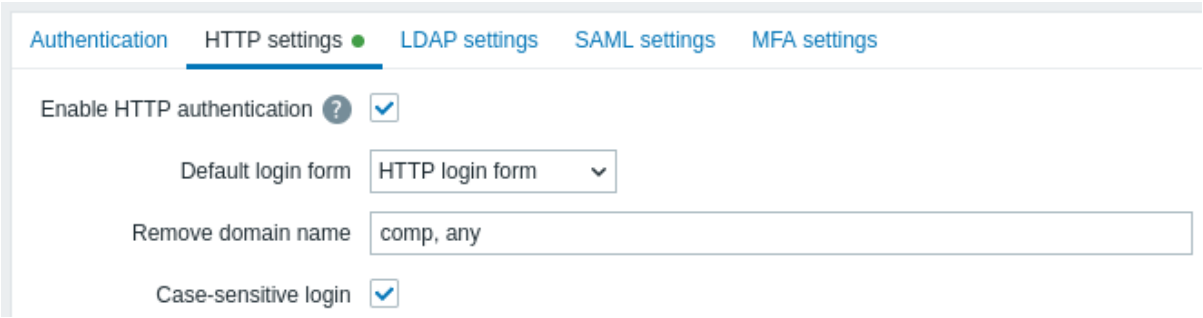
Présentation

L'authentification basée sur HTTP ou sur un serveur web (par exemple : BasicAuthentication, NTLM/Kerberos) peut être utilisée pour vérifier les noms d'utilisateur et les mots de passe. Notez qu'un utilisateur doit également exister dans Zabbix, mais son mot de passe Zabbix ne sera pas utilisé.

Attention:
Attention ! Assurez-vous que l'authentification du serveur web est configurée et fonctionne correctement avant de l'activer.

Note:
La configuration de l'authentification HTTP peut être désactivée en définissant `$ZBX_FEATURE_FLAGS['http_auth_enabled'] = false` dans le fichier de configuration de l'interface (`zabbix.conf.php`). Dans ce cas, l'onglet *Paramètres HTTP* contenant les options d'authentification HTTP sera masqué.

Configuration



Paramètres de configuration :

Paramètre	Description
Activer l'authentification HTTP	Cochez la case pour activer l'authentification HTTP. En plaçant le curseur sur ?, une infobulle s'affiche pour avertir que, dans le cas d'une authentification par le serveur web, tous les utilisateurs (même ceux dont l'accès à l'interface est défini sur LDAP/Interne) seront authentifiés par le serveur web et non par Zabbix.

Paramètre	Description
Formulaire de connexion par défaut	Indiquez si les utilisateurs non authentifiés doivent être redirigés vers : **Formulaire de connexion Zabbix** - page de connexion Zabbix standard. **Formulaire de connexion HTTP** - page de connexion HTTP. Il est recommandé d'activer l'authentification basée sur le serveur web uniquement pour la page `index_http.php`. Si *Formulaire de connexion par défaut* est défini sur « page de connexion HTTP », l'utilisateur sera connecté automatiquement si le module d'authentification du serveur web définit un nom d'utilisateur valide dans la variable `$_SERVER`. Les clés `$_SERVER` prises en charge sont `PHP_AUTH_USER`, `REMOTE_USER`, `AUTH_USER`.
Supprimer le nom de domaine	Liste de noms de domaine, séparés par des virgules, qui doivent être supprimés du nom d'utilisateur. Par exemple, `comp,any` : si le nom d'utilisateur est « Admin@any » ou « comp\Admin », l'utilisateur sera connecté en tant que « Admin » ; si le nom d'utilisateur est « notacompany\Admin », la connexion sera refusée.
Connexion sensible à la casse	Décochez la case pour désactiver la distinction entre majuscules et minuscules dans les noms d'utilisateur (activée par défaut). La désactivation de la distinction entre majuscules et minuscules permet, par exemple, de se connecter en tant que « admin » même si l'utilisateur Zabbix est « Admin » ou « ADMIN ». Notez que si la distinction entre majuscules et minuscules est désactivée et que plusieurs utilisateurs Zabbix ont des noms d'utilisateur similaires (par exemple, Admin et admin), la connexion de ces utilisateurs sera toujours refusée avec le message d'erreur suivant : « Échec de l'authentification : les identifiants fournis ne sont pas uniques. »

Note:
Pour les utilisateurs internes qui ne parviennent pas à se connecter avec des identifiants HTTP (lorsque le formulaire de connexion HTTP est défini par défaut), ce qui entraîne une erreur 401, vous pouvez ajouter une ligne `ErrorDocument 401 /index.php?form=default` aux directives d'authentification de base afin de rediriger vers le formulaire de connexion Zabbix standard.

LDAP

Aperçu

L'authentification LDAP externe peut être utilisée pour vérifier les noms d'utilisateur et les mots de passe.

L'authentification LDAP de Zabbix fonctionne au moins avec Microsoft Active Directory et OpenLDAP.

Si seule la connexion LDAP est configurée, l'utilisateur doit également exister dans Zabbix, toutefois son mot de passe Zabbix ne sera pas utilisé. Si l'authentification réussit, Zabbix fera correspondre un nom d'utilisateur local avec l'attribut de nom d'utilisateur renvoyé par LDAP.

Provisionnement des utilisateurs

Il est possible de configurer le **provisionnement JIT (juste à temps) des utilisateurs** LDAP. Dans ce cas, il n'est pas nécessaire qu'un utilisateur existe déjà dans Zabbix. Le compte utilisateur peut être créé lorsque l'utilisateur se connecte à Zabbix pour la première fois.

Lorsqu'un utilisateur LDAP saisit son identifiant et son mot de passe LDAP, Zabbix vérifie sur le serveur LDAP *par défaut* si cet utilisateur existe. Si l'utilisateur existe et ne possède pas encore de compte dans Zabbix, un nouvel utilisateur est créé dans Zabbix et l'utilisateur peut se connecter.

Un utilisateur créé via le provisionnement JIT est associé au serveur LDAP (annuaire) défini comme serveur par défaut au moment de sa création. La modification ultérieure du serveur LDAP par défaut ne modifie pas et ne met pas à jour le serveur LDAP associé aux utilisateurs déjà provisionnés.

Attention:
Si le provisionnement JIT est activé, un groupe d'utilisateurs pour les utilisateurs déprovisionnés doit être spécifié dans l'onglet *Authentification*.

Le provisionnement JIT permet également de mettre à jour les comptes utilisateurs provisionnés en fonction des modifications apportées dans LDAP. Par exemple, si un utilisateur est déplacé d'un groupe LDAP vers un autre, il est également déplacé d'un groupe vers un autre dans Zabbix ; si un utilisateur est supprimé d'un groupe LDAP, il est également supprimé du groupe dans

Zabbix et, s'il n'appartient à aucun autre groupe, il est ajouté au groupe d'utilisateurs pour les utilisateurs déprovisionnés. Les comptes utilisateurs provisionnés sont mis à jour selon la période de provisionnement configurée ou lorsque l'utilisateur se connecte à Zabbix.

Notez que le provisionnement en arrière-plan est effectué par l'interface Zabbix lorsque l'utilisateur interagit avec celle-ci ou possède au moins une page de l'interface ouverte dans le navigateur. Aucun processus d'arrière-plan dédié n'est utilisé pour provisionner les utilisateurs.

LDAP prend en charge trois méthodes de liaison à l'annuaire pour l'authentification et les recherches :

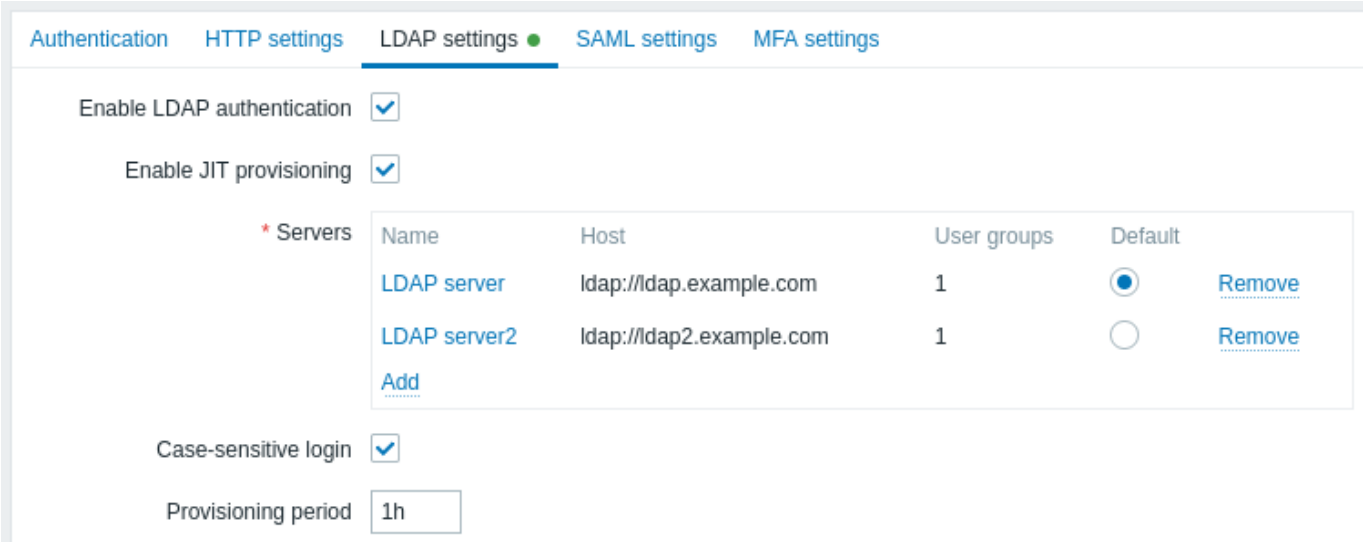
- Liaison anonyme - aucun *Bind DN* / *Bind password* n'est fourni et le serveur LDAP autorise les requêtes anonymes.
- Utilisateur de liaison dédié (compte de service) - un compte LDAP spécifique est défini dans *Bind DN* / *Bind password* et utilisé par Zabbix pour effectuer les recherches et le provisionnement. Cette option est recommandée et offre la plus grande flexibilité, car Zabbix peut effectuer des recherches et un provisionnement en arrière-plan sans les identifiants de l'utilisateur final.
- Liaison directe de l'utilisateur - Zabbix effectue la liaison à l'aide des identifiants saisis par l'utilisateur lors de la connexion (aucun *Bind DN* / *Bind password* n'est configuré) ; cette méthode est configurée en incluant un espace réservé tel que `uid=%{user}` dans le Base DN. Dans ce mode, Zabbix n'a accès au mot de passe de l'utilisateur que pendant la connexion interactive. Par conséquent, les actions de provisionnement qui nécessitent une authentification auprès de LDAP en dehors de la session de connexion de l'utilisateur (par exemple, l'utilisation du bouton *Provision now* ou l'exécution d'un provisionnement en arrière-plan lorsque l'utilisateur n'est pas connecté activement) ne peuvent pas s'authentifier et ne fonctionneront donc pas. Le provisionnement et les mises à jour pour la liaison directe de l'utilisateur ont lieu uniquement au moment où l'utilisateur se connecte.

Plusieurs serveurs

Plusieurs serveurs LDAP peuvent être définis, si nécessaire. Par exemple, un serveur différent peut être utilisé pour authentifier un autre groupe d'utilisateurs. Une fois les serveurs LDAP configurés, dans la configuration des groupes d'utilisateurs, il devient possible de sélectionner le serveur LDAP requis pour le groupe d'utilisateurs concerné.

Si un utilisateur appartient à plusieurs groupes d'utilisateurs et à plusieurs serveurs LDAP, le premier serveur de la liste des serveurs LDAP triés par nom par ordre croissant sera utilisé pour l'authentification.

Configuration

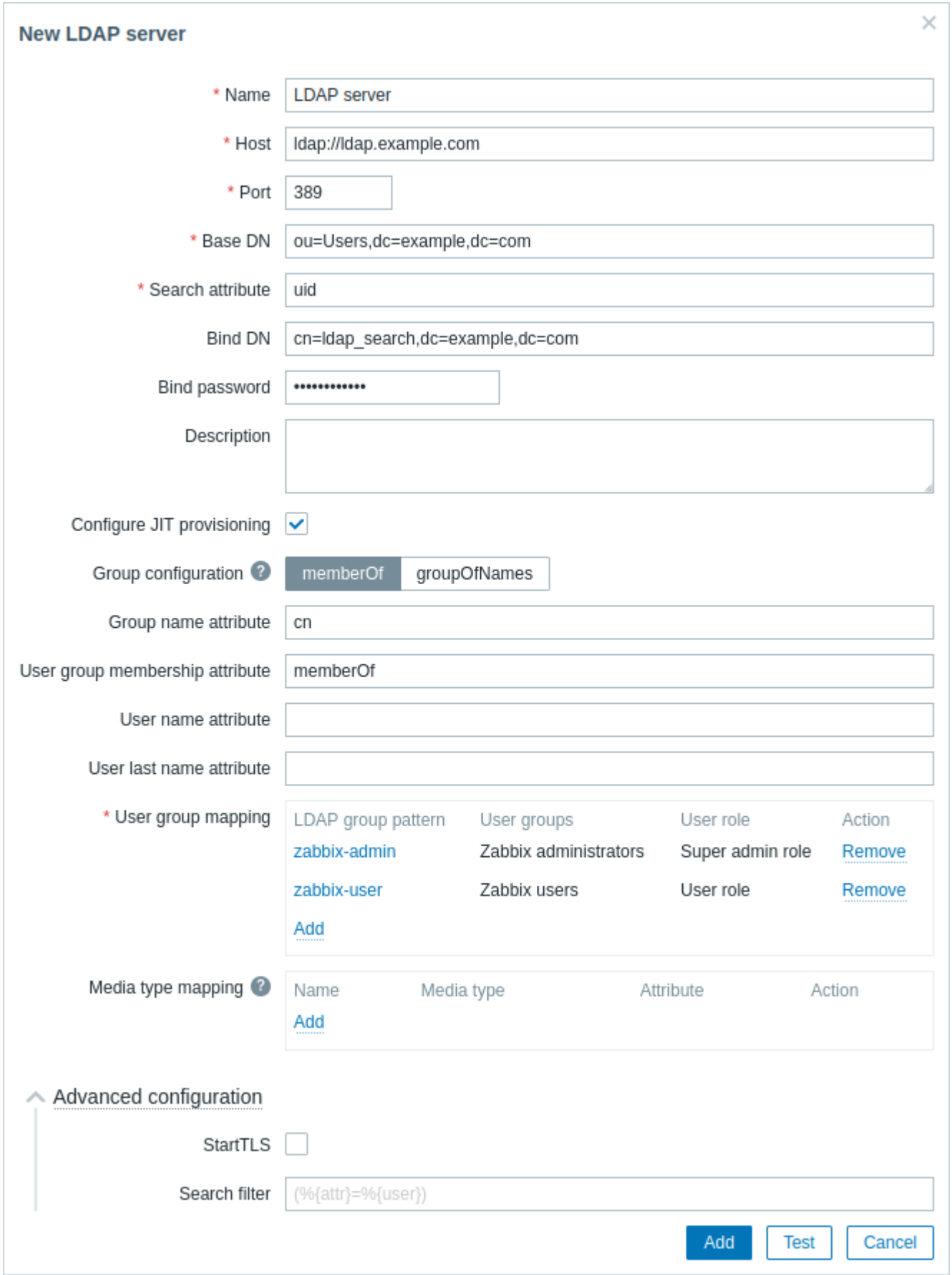


Paramètres de configuration :

Paramètre	Description
Activer l'authentification LDAP	Cochez la case pour activer l'authentification LDAP.
Activer le provisionnement JIT	Cochez la case pour activer le provisionnement JIT.
Serveurs	Cliquez sur *Ajouter* pour configurer un serveur LDAP (voir Configuration du serveur LDAP ci-dessous).

Paramètre	Description
Connexion sensible à la casse	Décochez la case pour désactiver la sensibilité à la casse lors de la connexion avec les noms d'utilisateur (activée par défaut). La désactivation de la sensibilité à la casse permet, par exemple, de se connecter en tant que « admin » même si l'utilisateur Zabbix est « Admin » ou « ADMIN ». Veuillez noter que si la sensibilité à la casse est désactivée et que plusieurs utilisateurs Zabbix ont des noms d'utilisateur similaires (par exemple, Admin et admin), la connexion de ces utilisateurs sera toujours refusée avec le message d'erreur suivant : « Authentication failed: supplied credentials are not unique. »
Période de provisionnement	Définissez la période de provisionnement, c'est-à-dire la fréquence à laquelle l'utilisateur connecté sera provisionné lors de l'utilisation de l'interface.

Configuration du serveur LDAP



Paramètres de configuration du serveur LDAP :

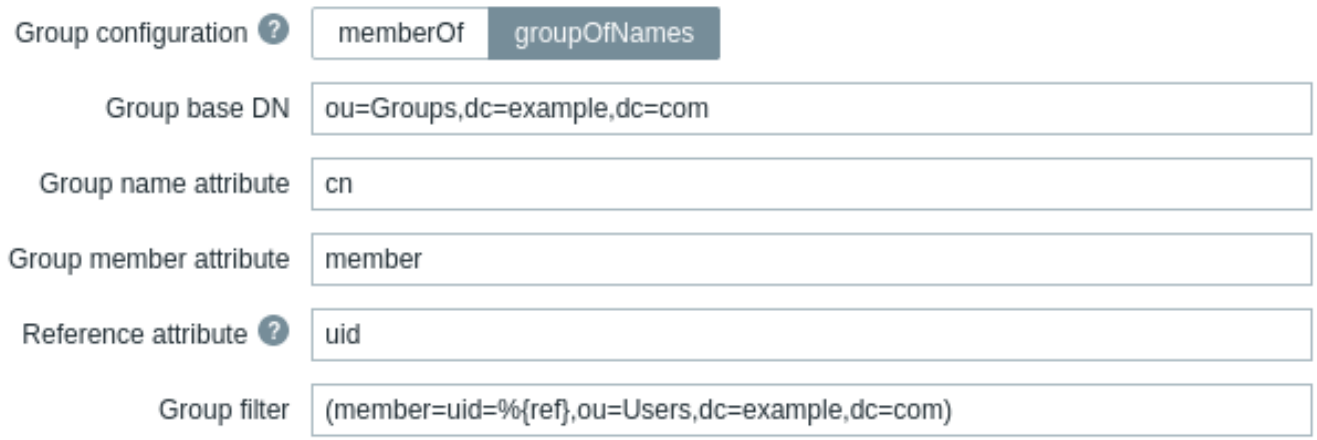
Parameter	Description
Name	Nom du serveur LDAP dans la configuration de Zabbix.

Parameter	Description
Host	Nom d'hôte, IP ou URI du serveur LDAP. Exemples : ldap.example.com, 127.0.0.1, ldap://ldap.example.com Pour un serveur LDAP sécurisé, utilisez le protocole *ldaps* et un nom d'hôte. Exemple : ldaps://ldap.example.com Avec OpenLDAP 2.x.x et versions ultérieures, un URI LDAP complet de la forme ldap://hostname:port ou ldaps://hostname:port peut être utilisé.
Port	Port du serveur LDAP. La valeur par défaut est 389. Pour une connexion LDAP sécurisée, le numéro de port est normalement 636. Non utilisé lors de l'utilisation d'URI LDAP complets.
Base DN	Chemin de base vers les comptes utilisateur sur le serveur LDAP : ou=Users,ou=system (pour OpenLDAP), DC=company,DC=com (pour Microsoft Active Directory) uid=%{user},dc=example,dc=com (pour une liaison directe de l'utilisateur, voir la note ci-dessous)
Search attribute	Attribut du compte LDAP utilisé pour la recherche : uid (pour OpenLDAP), sAMAccountName (pour Microsoft Active Directory)
Bind DN	Compte LDAP utilisé pour la liaison et la recherche sur le serveur LDAP, exemples : uid=ldap_search,ou=system (pour OpenLDAP), CN=ldap_search,OU=user_group,DC=company,DC=com (pour Microsoft Active Directory) La liaison anonyme est également prise en charge. Notez que la liaison anonyme ouvre potentiellement la configuration du domaine à des utilisateurs non autorisés (informations sur les utilisateurs, ordinateurs, serveurs, groupes, services, etc.). Pour des raisons de sécurité, désactivez les liaisons anonymes sur les hôtes LDAP et utilisez plutôt un accès authentifié.
Bind password	Mot de passe LDAP du compte utilisé pour la liaison et la recherche sur le serveur LDAP.
Description	Description du serveur LDAP.
Configure JIT provisioning	Cochez cette case pour afficher les options liées au provisionnement JIT.
Group configuration	Sélectionnez la méthode de configuration des groupes : **memberOf** - en recherchant les utilisateurs et leur attribut d'appartenance aux groupes **groupOfNames** - en recherchant les groupes via l'attribut member Notez que memberOf est préférable car il est plus rapide ; utilisez groupOfNames si votre serveur LDAP ne prend pas en charge `memberOf` ou si un filtrage des groupes est requis.
Group name attribute	Spécifiez l'attribut permettant d'obtenir le nom du groupe à partir de tous les objets de l'attribut `memberOf` (voir le champ *User group membership attribute*) Le nom du groupe est nécessaire pour le mappage des groupes d'utilisateurs.
User group membership attribute	Spécifiez l'attribut qui contient des informations sur les groupes auxquels l'utilisateur appartient (par exemple, `memberOf`). Par exemple, l'attribut memberOf peut contenir des informations comme ceci : `memberOf=cn=zabbix-admin,ou=Groups,dc=example,dc=com` Ce champ est disponible uniquement pour la méthode memberOf.
User name attribute	Spécifiez l'attribut qui contient le prénom de l'utilisateur.
User last name attribute	Spécifiez l'attribut qui contient le nom de famille de l'utilisateur.
User group mapping	Associez un modèle de groupe d'utilisateurs LDAP à un groupe d'utilisateurs Zabbix et à un rôle utilisateur. Ceci est nécessaire pour déterminer quel groupe/rôle utilisateur l'utilisateur provisionné recevra dans Zabbix. Cliquez sur *Add* pour ajouter un mappage. Le champ *LDAP group pattern* prend en charge les caractères génériques. Le nom du groupe doit correspondre à un groupe existant. Si un utilisateur LDAP correspond à plusieurs groupes d'utilisateurs Zabbix, l'utilisateur devient membre de tous ces groupes. Si un utilisateur correspond à plusieurs rôles utilisateur Zabbix, il obtiendra celui ayant le niveau d'autorisation le plus élevé parmi eux.
Media type mapping	Associez les attributs LDAP media de l'utilisateur (par exemple, l'e-mail) aux médias utilisateur Zabbix pour l'envoi de notifications (la valeur de l'attribut est utilisée comme champ média *Send to*).
Advanced configuration	Cliquez sur l'en-tête *Advanced configuration* pour afficher les options de configuration avancées (voir ci-dessous).
StartTLS	Cochez la case pour utiliser l'opération StartTLS lors de la connexion au serveur LDAP. La connexion échouera si le serveur ne prend pas en charge StartTLS. StartTLS ne peut pas être utilisé avec des serveurs qui utilisent le protocole *ldaps*.

Parameter	Description
Search filter	Définissez une chaîne personnalisée lors de l'authentification d'un utilisateur dans LDAP. Les espaces réservés suivants sont pris en charge : `%{attr}` - nom de l'attribut de recherche (uid, sAMAccountName) `%{user}` - valeur du nom d'utilisateur à authentifier Par exemple, pour effectuer une recherche sensible à la casse dans un environnement LDAP ou Microsoft Active Directory insensible à la casse, la chaîne peut être définie comme suit : `(%{attr}:caseExactMatch:=%{user})`. Si le filtre n'est pas personnalisé, LDAP utilisera la valeur par défaut : `(%{attr}=%{user})`.

Note:
Pour configurer un serveur LDAP pour une **liaison directe de l'utilisateur**, ajoutez un attribut uid=%{user} au paramètre *Base DN* (par exemple, *uid=%{user},dc=example,dc=com*) et laissez les paramètres *BindDN* et *Bind password* vides. Lors de l'authentification, l'espace réservé %{user} sera remplacé par le nom d'utilisateur saisi lors de la connexion.
 Avec la liaison directe de l'utilisateur, Zabbix n'a accès aux identifiants de l'utilisateur que pendant la connexion interactive. Par conséquent, les tâches de provisionnement qui s'exécutent sans connexion interactive (par exemple, le bouton *Provision now* ou la méthode API `user.provision`) ignoreront les utilisateurs qui s'authentifient via la liaison directe de l'utilisateur, car Zabbix ne peut pas effectuer la liaison en leur nom. Utilisez une liaison anonyme ou un utilisateur de liaison dédié (compte de service) si vous avez besoin que le provisionnement et les recherches LDAP fonctionnent depuis l'interface sans que l'utilisateur final ait à être connecté.

Les champs suivants sont spécifiques à "groupOfNames" comme méthode *Group configuration* :



Parameter	Description
Group base DN	Chemin de base des groupes sur le serveur LDAP.
Group name attribute	Spécifiez l'attribut permettant d'obtenir le nom du groupe dans le chemin de base des groupes indiqué. Le nom du groupe est nécessaire pour le mappage des groupes d'utilisateurs.
Group member attribute	Spécifiez l'attribut qui contient des informations sur les membres du groupe dans LDAP (par exemple, `member`).
Reference attribute	Spécifiez l'attribut de référence pour le filtre de groupe (voir le champ *Group filter*). Utilisez ensuite `%{ref}` dans le filtre de groupe pour obtenir les valeurs de l'attribut spécifié ici.
Group filter	Spécifiez le filtre permettant de récupérer le groupe dont l'utilisateur est membre. Par exemple, `(member=uid=%{ref},ou=Users,dc=example,dc=com)` correspondra à "User1" si l'attribut member du groupe est `uid=User1,ou=Users,dc=example,dc=com` et renverra le groupe dont "User1" est membre.

Warning:
En cas de problème avec les certificats, pour faire fonctionner une connexion LDAP sécurisée (ldaps), vous devrez peut-être ajouter une ligne `TLS_REQCERT allow` au fichier de configuration `/etc/openldap/ldap.conf`. Cela peut réduire la sécurité de la connexion au catalogue LDAP.

Note:
Il est recommandé de créer un compte LDAP distinct (*Bind DN*) pour effectuer la liaison et la recherche sur le serveur LDAP avec des privilèges minimaux dans LDAP, au lieu d'utiliser de vrais comptes utilisateur (utilisés pour la connexion à l'interface Zabbix).
 Une telle approche offre davantage de sécurité et ne nécessite pas de modifier le *Bind password* lorsque l'utilisateur change son propre mot de passe sur le serveur LDAP.
 Dans le tableau ci-dessus, il s'agit du nom de compte *ldap_search*.

Test de l'accès

Le bouton *Tester* vous permet de tester l'accès utilisateur :

Paramètre	Description
Identifiant	Nom d'utilisateur LDAP à tester (prérempli avec le nom d'utilisateur actuel de l'interface Zabbix). Ce nom d'utilisateur doit exister sur le serveur LDAP. Zabbix n'activera pas l'authentification LDAP s'il ne parvient pas à authentifier l'utilisateur de test.
Mot de passe utilisateur	Mot de passe de l'utilisateur LDAP à tester.

SAML

Vue d'ensemble

L'authentification SAML 2.0 peut être utilisée pour se connecter à Zabbix.

Si la connexion SAML est configurée sans provisionnement JIT des utilisateurs, l'utilisateur doit déjà exister dans Zabbix ; le mot de passe Zabbix de l'utilisateur n'est pas utilisé pour l'authentification. Après une authentification SAML réussie, Zabbix associe l'attribut de nom d'utilisateur reçu de SAML à un nom d'utilisateur Zabbix local existant.

Provisionnement des utilisateurs

Il est possible de configurer le **provisionnement** JIT (just-in-time) des utilisateurs SAML. Dans ce cas, il n'est pas nécessaire qu'un utilisateur existe déjà dans Zabbix. Le compte utilisateur peut être créé lorsque l'utilisateur se connecte à Zabbix pour la première fois.

Attention:
Si le provisionnement JIT est activé, un groupe d'utilisateurs pour les utilisateurs déprovisionnés doit être spécifié dans l'onglet *Authentication*.

En plus du provisionnement JIT, il est également possible d'activer et de configurer le provisionnement SCIM (System for Cross-domain Identity Management) - une gestion *continue* des comptes utilisateurs pour les utilisateurs créés par le provisionnement des utilisateurs. Le provisionnement SCIM nécessite un jeton d'API Zabbix (avec des autorisations Super admin) pour l'authentification dans Zabbix.

Par exemple, si un utilisateur est déplacé d'un groupe SAML à un autre, l'utilisateur sera également déplacé d'un groupe à un autre dans Zabbix ; si un utilisateur est supprimé d'un groupe SAML, l'utilisateur sera également supprimé du groupe dans Zabbix et, s'il n'appartient à aucun autre groupe, ajouté au groupe d'utilisateurs pour les utilisateurs déprovisionnés.

Si SCIM est activé et configuré, un utilisateur SAML sera provisionné au moment où il se connecte à Zabbix et sera mis à jour en continu en fonction des changements dans SAML. Les utilisateurs SAML déjà existants ne seront pas provisionnés, et seuls les utilisateurs provisionnés seront mis à jour. Notez que seuls les médias valides seront ajoutés à un utilisateur lorsque celui-ci est provisionné ou mis à jour.

Si SCIM n'est pas activé, un utilisateur SAML sera provisionné (puis mis à jour ultérieurement) au moment où il se connecte à Zabbix.

Note:
Si l'authentification SAML est activée, les utilisateurs pourront choisir entre une connexion locale ou via l'authentification unique SAML. Si le provisionnement JIT est utilisé, seule l'authentification unique est possible.

Configuration du fournisseur d'identité

Afin de fonctionner avec Zabbix, un fournisseur d'identité SAML (onelogin.com, auth0.com, okta.com, etc.) doit être configuré de la manière suivante :

- *Assertion Consumer URL* doit être définie sur `<path_to_zabbix_ui>/index_sso.php?acs`
- *Single Logout URL* doit être définie sur `<path_to_zabbix_ui>/index_sso.php?sls`

Exemples de `<path_to_zabbix_ui>` : `https://example.com/zabbix/ui`, `http://another.example.com/zabbix`, `http://<any_public_ip_address>/zabbix`

Configuration de Zabbix

> **Attention:**
> Il est nécessaire d'installer php-openssl si vous souhaitez utiliser l'authentification SAML dans l'interface.

Pour utiliser l'authentification SAML, Zabbix doit être configuré comme suit :

1. Fournissez la clé privée et les certificats du SP. Un certificat IdP doit être fourni (collez-le ou sélectionnez un fichier) pour activer SAML via l'interface. Un certificat SP et une clé privée SP doivent être fournis si les options *Signer* ou *Chiffrer* sont sélectionnées.

Si `$SSO['CERT_STORAGE'] = 'database'` est défini dans le fichier de configuration de l'interface (`zabbix.conf.php`), vous pouvez coller le texte du certificat ou téléverser le fichier du certificat dans l'interface pendant la configuration de SAML : aucun fichier n'est requis sur le système de fichiers.

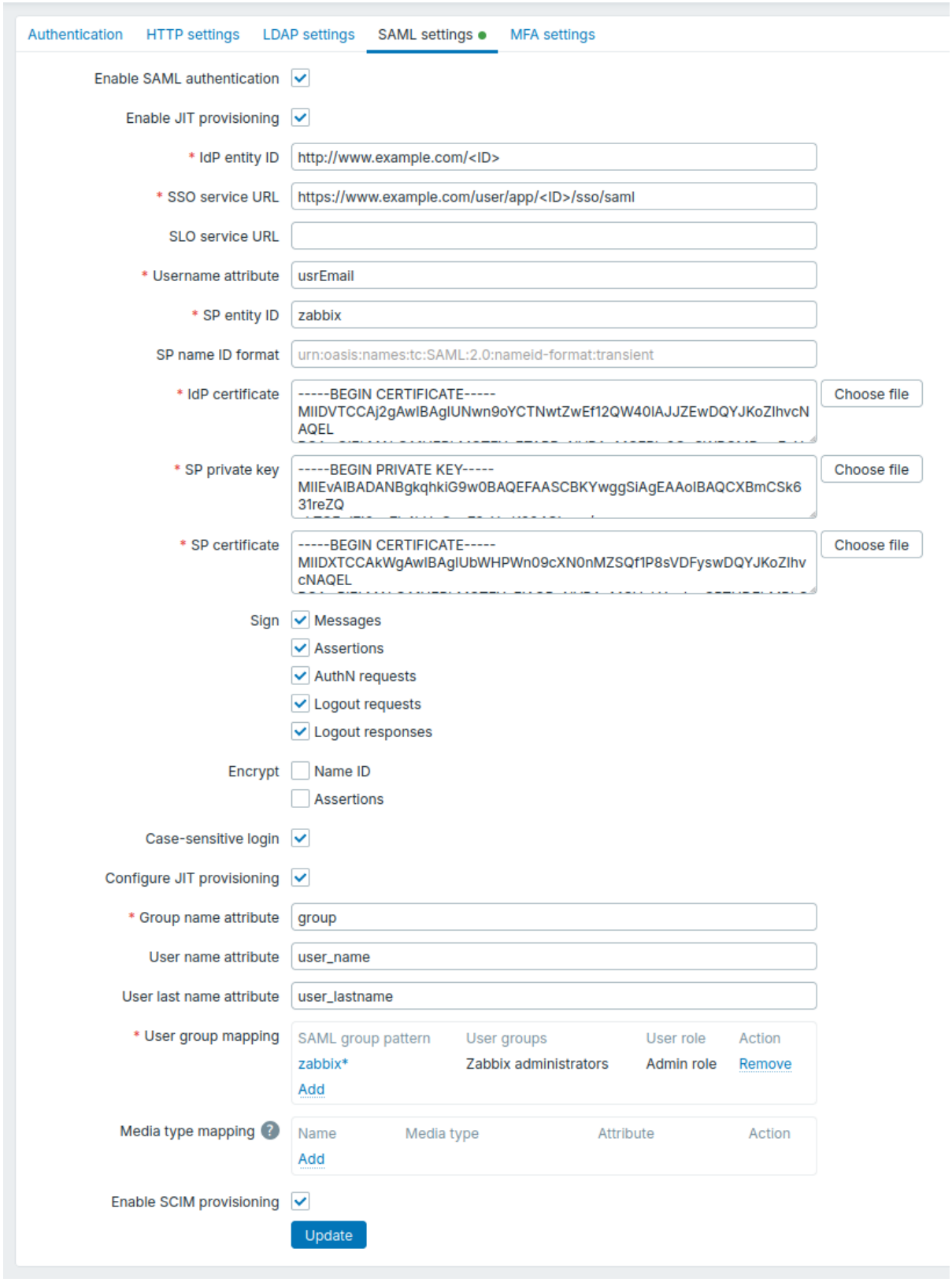
Si `$SSO['CERT_STORAGE'] = 'file'` est défini dans le fichier de configuration, le certificat doit être disponible sur le système de fichiers (par défaut dans `ui/conf/certs` ou dans le chemin configuré dans le fichier de configuration) et l'interface ne stockera pas les certificats dans la base de données. Notez que si `$SSO['CERT_STORAGE']` n'est pas défini ou est commenté, le stockage dans des fichiers est utilisé.

Par défaut, Zabbix recherchera les fichiers aux emplacements suivants :

- ui/conf/certs/sp.key - fichier de clé privée SP
- ui/conf/certs/sp.crt - fichier de certificat SP
- ui/conf/certs/idp.crt - fichier de certificat IdP

> **Attention:**
> Les certificats et les clés peuvent être importés via l'interface en collant leur texte ou en téléversant des fichiers. Bien que les importations non chiffrées soient autorisées par commodité dans certains environnements, l'utilisation d'un transport chiffré est fortement recommandée (par exemple, activez HTTPS/TLS pour l'interface Zabbix). Lorsque les certificats et les clés SAML sont stockés dans la base de données, activez TLS/SSL pour les connexions à la base de données afin de protéger le trafic entre l'application et la base de données.
 Chiffrez les sauvegardes de la base de données susceptibles de contenir des certificats SAML ou des clés privées, et limitez l'accès aux fichiers de sauvegarde ainsi qu'aux tables de la base de données qui stockent les éléments SAML. Limitez les privilèges de l'utilisateur de la base de données au minimum nécessaire.

2. Tous les paramètres les plus importants peuvent être configurés dans l'interface Zabbix. Cependant, il est possible de spécifier des paramètres supplémentaires dans le fichier de configuration.



Paramètres de configuration disponibles dans l'interface Zabbix :

Paramètre	Description
Activer l'authentification SAML	Cochez la case pour activer l'authentification SAML.
Activer le provisionnement JIT	Cochez la case pour activer le provisionnement JIT des utilisateurs.
ID d'entité de l'IDP	Identifiant d'entité unique au sein du fournisseur d'identité SAML.
URL du service SSO	URL vers laquelle les utilisateurs seront redirigés lors de la connexion.
URL du service SLO	URL vers laquelle les utilisateurs seront redirigés lors de la déconnexion. Si elle est laissée vide, le service SLO ne sera pas utilisé.

Paramètre	Description
Attribut du nom d'utilisateur	Attribut SAML à utiliser comme nom d'utilisateur lors de la connexion à Zabbix. La liste des valeurs prises en charge est déterminée par le fournisseur d'identité. Exemples : uid userprincipalname samaccountname username userusername urn:oid:0.9.2342.19200300.100.1.1 urn:oid:1.3.6.1.4.1.5923.1.1.1.13 urn:oid:0.9.2342.19200300.100.1.44
ID d'entité du SP	Identifiant unique du fournisseur de services (si la correspondance n'est pas établie, l'opération sera rejetée). Il est possible de spécifier une URL ou toute chaîne de données.
Format de l'ID de nom du SP	Demande un format d'ID de nom particulier dans la réponse. Exemples : urn:oasis:names:tc:SAML:2.0:nameid-format:persistent urn:oasis:names:tc:SAML:2.0:nameid-format:unspecified urn:oasis:names:tc:SAML:2.0:nameid-format:transient
Certificat IdP	Contenu du certificat du fournisseur de services (SP) pour la configuration du service d'authentification unique (SSO) SAML.
Clé privée SP	Contenu de la clé privée du fournisseur de services (SP) pour la configuration du service d'authentification unique (SSO) SAML. Elle fournit une authentification sécurisée et un échange de données avec le fournisseur d'identité (IdP).
Certificat SP	Contenu du certificat du fournisseur de services (SP) pour la configuration du service d'authentification unique (SSO) SAML.
Signer	Décochez les cases pour désactiver les signatures SAML pour les entités correspondantes : *Messages* *Assertions* *Requêtes AuthN* *Requêtes de déconnexion* *Réponses de déconnexion*
Chiffrer	Cochez les cases pour sélectionner les entités pour lesquelles le chiffrement SAML doit être activé : *ID de nom* *Assertions*
Connexion sensible à la casse	Décochez la case pour désactiver la sensibilité à la casse lors de la connexion avec les noms d'utilisateur (activée par défaut). La désactivation de la sensibilité à la casse permet, par exemple, de se connecter en tant que « admin » même si l'utilisateur Zabbix est « Admin » ou « ADMIN ». Notez que si la sensibilité à la casse est désactivée et que plusieurs utilisateurs Zabbix ont des noms d'utilisateur similaires (par exemple, Admin et admin), la connexion de ces utilisateurs sera toujours refusée avec le message d'erreur suivant : « Échec de l'authentification : les identifiants fournis ne sont pas uniques. »
Configurer le provisionnement JIT	Cochez cette case pour afficher les options liées au provisionnement JIT des utilisateurs.
Attribut du nom du groupe	Spécifiez l'attribut du nom du groupe pour le provisionnement JIT des utilisateurs.
Attribut du nom de l'utilisateur	Spécifiez l'attribut du nom de l'utilisateur pour le provisionnement JIT des utilisateurs.
Attribut du nom de famille de l'utilisateur	Spécifiez l'attribut du nom de famille de l'utilisateur pour le provisionnement JIT des utilisateurs.

Paramètre	Description
Mappage des groupes d'utilisateurs	Mappez un modèle de groupe d'utilisateurs SAML à un groupe d'utilisateurs et à un rôle d'utilisateur Zabbix. Cela est nécessaire pour déterminer le groupe et le rôle d'utilisateur qui seront attribués à l'utilisateur provisionné dans Zabbix. Cliquez sur *Ajouter* pour ajouter un mappage. Le champ *Modèle de groupe SAML* prend en charge les caractères génériques. Le nom du groupe doit correspondre à un groupe existant. Si un utilisateur SAML correspond à plusieurs groupes d'utilisateurs Zabbix, il devient membre de chacun d'eux. Si un utilisateur correspond à plusieurs rôles d'utilisateur Zabbix, il obtient le niveau d'autorisation le plus élevé parmi ces rôles.
Mappage des types de média	Mappez les attributs de média SAML de l'utilisateur (par exemple, l'adresse e-mail) aux médias de l'utilisateur Zabbix pour l'envoi de notifications.
Activer le provisionnement SCIM	Cochez cette case pour activer le provisionnement SCIM 2.0.

Consultez les exemples de configuration des fournisseurs d'identité SAML pour la connexion et le provisionnement des utilisateurs dans Zabbix avec :

- Microsoft Azure AD
- Okta
- Onelogin

Remarques sur le provisionnement SCIM

Pour le provisionnement SCIM, indiquez le chemin vers l'interface Zabbix et ajoutez `api_scim.php` à celui-ci, du côté du fournisseur d'identité, c'est-à-dire :

`https://<path-to-zabbix-ui>/api_scim.php`

Les attributs utilisateur utilisés dans Zabbix (nom d'utilisateur, prénom, nom de famille et attributs de média) doivent être ajoutés en tant qu'attributs personnalisés et, si nécessaire, l'espace de noms externe doit être identique au schéma utilisateur : `urn:ietf:params:scim:schemas:core:2.0:User`.

Paramètres avancés

Des paramètres SAML supplémentaires peuvent être configurés dans le fichier de configuration de l'interface Zabbix (`zabbix.conf.php`) :

- `$SSO['SP_KEY']`
- `$SSO['SP_CERT']`
- `$SSO['IDP_CERT']`
- `$SSO['SETTINGS']`

Exemples de configuration

Exemple 1 : Connexion à une instance Zabbix avec un équilibreur de charge utilisant TLS/SSL

Si un équilibreur de charge est utilisé pour se connecter à une instance Zabbix et que l'équilibreur de charge utilise TLS/SSL alors que Zabbix ne l'utilise pas, spécifiez les paramètres `baseurl`, `strict` et `use_proxy_headers` comme suit :

```
$SSO['SETTINGS'] = [
    'strict' => false,
    'baseurl' => 'https://zabbix.example.com/zabbix/',
    'use_proxy_headers' => true
];
```

Exemple 2 : Configuration d'algorithmes personnalisés de signature et de condensat SAML

```
$SSO['SETTINGS'] = [
    'security' => [
        'signatureAlgorithm' => 'http://www.w3.org/2001/04/xmldsig-more#rsa-sha384'
        'digestAlgorithm' => 'http://www.w3.org/2001/04/xmldsig-more#sha384',
        // ...
    ],
    // ...
];
```

Exemple 3 : Configuration du SSO avec authentification Kerberos à l'aide d'ADFS

```
$SSO['SETTINGS'] = [
    'security' => [
        'requestedAuthnContext' => [
            'urn:oasis:names:tc:SAML:2.0:ac:classes:Kerberos',
        ],
        'requestedAuthnContextComparison' => 'exact'
    ]
];
```

Dans ce cas, dans le champ *SP name ID* de la configuration SAML, définissez :

```
urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified
```

MFA

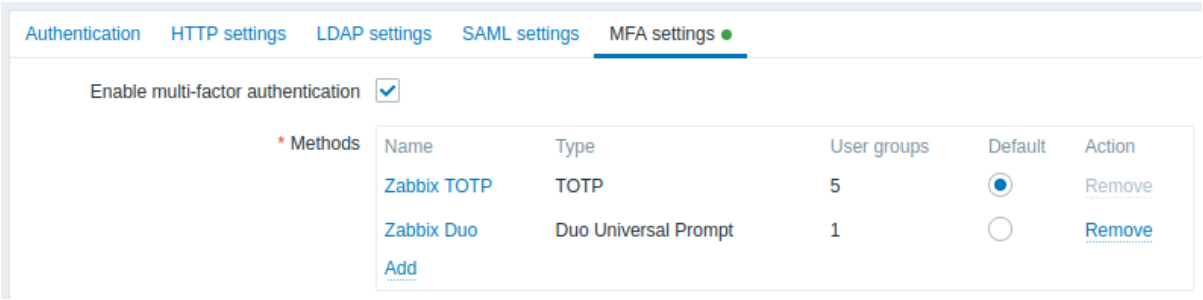
Aperçu

L'authentification multifacteur (MFA) peut être utilisée pour se connecter à Zabbix, en fournissant une couche de sécurité supplémentaire au-delà d'un simple nom d'utilisateur et mot de passe.

Avec la MFA, l'utilisateur doit exister dans Zabbix, doit fournir ses identifiants Zabbix lors de la connexion et doit également prouver son identité par d'autres moyens, généralement à l'aide d'un code généré par une application d'authentification sur le téléphone de l'utilisateur.

Plusieurs méthodes de MFA sont disponibles, permettant aux utilisateurs de choisir l'option qui correspond le mieux à leurs exigences de sécurité et à leurs préférences. Ces méthodes sont le mot de passe à usage unique basé sur le temps (TOTP) et Duo Universal Prompt.

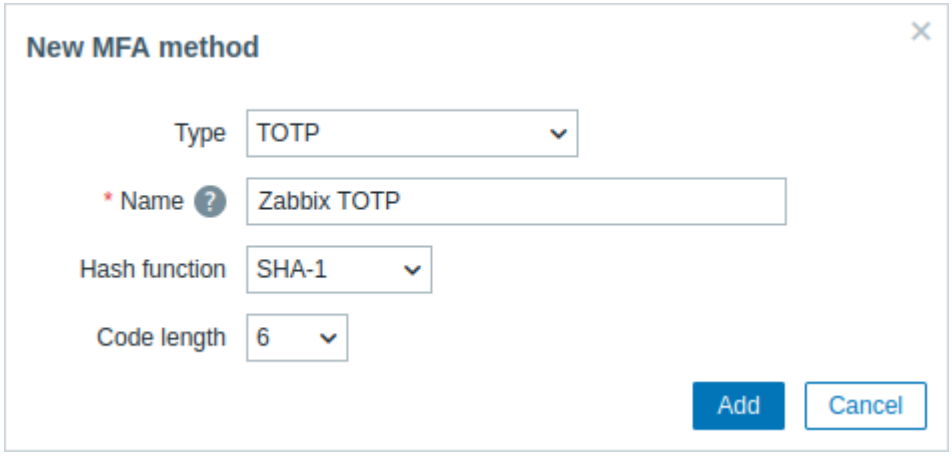
Configuration



Paramètres de configuration :

Paramètre	Description
Activer l'authentification multifacteur	Cochez la case pour activer l'authentification multifacteur.
Méthodes	Cliquez sur *Ajouter* pour configurer une méthode MFA (voir la configuration de la méthode ci-dessous).

Configuration de la méthode



Paramètres de configuration de la méthode :

Paramètre	Description
Type	Sélectionnez le type de méthode MFA : **TOTP** - utilisez une application d'authentification pour générer des mots de passe à usage unique basés sur le temps ; **Duo Universal Prompt** - utilisez le service d'authentification Duo pour fournir une authentification multifacteur.
Nom	Saisissez un nom affiché comme nom du compte à tous les utilisateurs MFA dans les applications d'authentification (par exemple, « Zabbix »).
Fonction de hachage	Sélectionnez la fonction de hachage (SHA-1, SHA-256 ou SHA-512) pour générer les codes TOTP. Ce paramètre est disponible si le type de méthode MFA est défini sur « TOTP ». Notez que le choix de SHA-256 ou SHA-512 peut limiter considérablement la compatibilité, car de nombreuses applications ne prennent actuellement pas en charge ces fonctions.
Longueur du code	Sélectionnez la longueur du code de vérification (6 ou 8). Ce paramètre est disponible si le type de méthode MFA est défini sur « TOTP ».
Nom d'hôte de l'API	Saisissez le nom d'hôte de l'API fourni par le service d'authentification Duo. Ce paramètre est disponible si le type de méthode MFA est défini sur « Duo Universal Prompt ».
ID client	Saisissez l'ID client fourni par le service d'authentification Duo. Ce paramètre est disponible si le type de méthode MFA est défini sur « Duo Universal Prompt ».
Secret client	Saisissez le secret client fourni par le service d'authentification Duo. Ce paramètre est disponible si le type de méthode MFA est défini sur « Duo Universal Prompt ».

Exemples de configuration

Cette section fournit des exemples de configuration de la MFA à l'aide de Time-Based One-Time Password (TOTP) et de Duo Universal Prompt.

TOTP

Pour TOTP, les utilisateurs doivent vérifier leur identité à l'aide d'une application d'authentification (par exemple, l'application Google Authenticator).

1. Accédez aux paramètres MFA dans Zabbix sous *Users* > *Authentication* et activez l'authentification multifacteur.

2. Ajoutez une nouvelle méthode MFA avec la configuration suivante :

- Type: TOTP
- Name: Zabbix TOTP
- Hash function: SHA-1
- Code length: 6

3. Cliquez sur le bouton *Add*, puis sur le bouton *Update*.

4. Allez dans *Users* > *User groups* et créez un nouveau groupe d'utilisateurs avec la configuration suivante :

- Group name: TOTP group
- Users: Admin
- Multi-factor authentication: Default (ou "Zabbix TOTP" s'il n'est pas défini comme valeur par défaut)

5. Déconnectez-vous de Zabbix, puis reconnectez-vous avec vos identifiants. Après une connexion réussie, il vous sera demandé d'enregistrer l'authentification multifacteur, avec l'affichage d'un code QR et d'une clé secrète.



6. Scannez le code QR ou saisissez la clé secrète dans l'application Google Authenticator. L'application générera un code de vérification que vous devrez saisir pour terminer le processus de connexion.

7. Lors des connexions suivantes, récupérez le code de vérification dans l'application Google Authenticator et saisissez-le pendant la connexion.

Duo Universal Prompt

Pour Duo Universal Prompt, les utilisateurs doivent vérifier leur identité à l'aide de l'application d'authentification Duo Mobile.

Attention:
La méthode MFA Duo Universal Prompt nécessite l'installation de l'extension php-curl, l'accès à Zabbix via HTTPS et l'autorisation des connexions sortantes vers les serveurs Duo. De plus, si vous avez activé Content Security Policy (CSP) sur le serveur web, veillez à ajouter "duo.com" à la directive CSP dans le fichier de configuration de votre hôte virtuel.

1. Inscrivez-vous pour obtenir un compte administrateur Duo gratuit sur Duo Signup.

2. Ouvrez le panneau d'administration Duo, allez dans *Applications* > *Protect an Application*, recherchez l'application *Web SDK*, puis cliquez sur *Protect*.

3. Notez les identifiants (Client ID, Client secret, API hostname) nécessaires pour configurer la méthode MFA dans Zabbix.

4. Accédez aux paramètres MFA dans Zabbix sous *Users* > *Authentication* et activez l'authentification multifacteur.

5. Ajoutez une nouvelle méthode MFA avec la configuration suivante :

- Type: Duo Universal Prompt
- Nom: Zabbix Duo
- API hostname: (utilisez l'API hostname de Duo)
- Client ID: (utilisez le Client ID de Duo)
- Client secret: (utilisez le Client secret de Duo)

6. Cliquez sur le bouton *Add*, puis sur le bouton *Update*.

7. Allez dans *Users* > *User groups* et créez un nouveau groupe d'utilisateurs avec la configuration suivante :

- Nom du groupe: Duo group
- Utilisateurs: Admin
- Authentification multifacteur: Default (ou "Zabbix Duo" si elle n'est pas définie comme valeur par défaut)

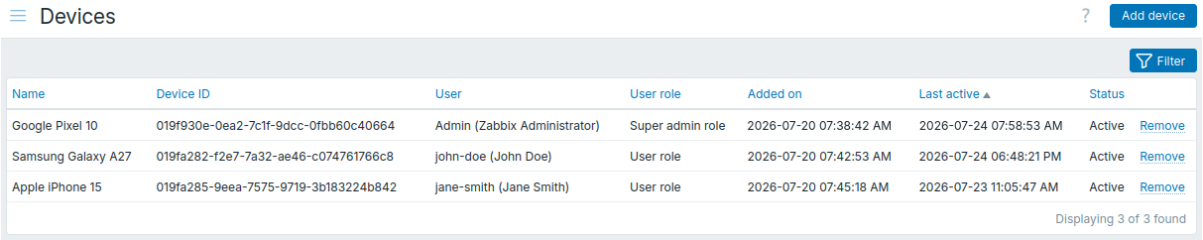
8. Déconnectez-vous de Zabbix, puis reconnectez-vous avec vos identifiants. Après une connexion réussie, il vous sera demandé de vous inscrire à la MFA, puis vous serez redirigé vers Duo. Terminez la configuration Duo et vérifiez votre utilisateur avec l'application Duo de votre téléphone pour vous connecter.

9. Pour les connexions suivantes, utilisez la méthode MFA appropriée fournie par l'application Duo (par exemple, récupérer un code de vérification, répondre aux notifications push ou utiliser des clés matérielles), puis saisissez les informations requises lors de la connexion.

Appareils

Cette section permet aux super administrateurs de gérer les appareils mobiles à utiliser avec l'application mobile Zabbix.

Cette section s'affiche si `EnableMobileDevices` est activé.



Pour ajouter un appareil :

1. Cliquez sur *Ajouter un appareil*.
2. Sélectionnez un utilisateur, puis cliquez sur *Ajouter*. L'utilisateur doit disposer de l'accès aux appareils.
3. Un code QR s'affiche. L'utilisateur doit scanner ce code dans son application mobile Zabbix.

Note:
Lorsque vous ajoutez un appareil pour un utilisateur, Zabbix ajoute également le type de média notification push aux médias de l'utilisateur. Lorsque vous supprimez l'appareil, ce média utilisateur est également supprimé. Si plusieurs types de médias de notification push sont configurés, Zabbix ajoute celui qui est activé et qui apparaît en premier dans l'ordre alphabétique.

Les appareils ajoutés apparaissent dans une liste que vous pouvez filtrer.

Vous pouvez également supprimer des appareils. Lorsque vous supprimez un appareil, ses identifiants d'accès sont révoqués et supprimés de Zabbix. Il ne peut alors plus accéder à Zabbix.

La liste affiche les informations suivantes :

Colonne	Description
Nom	Nom de l'appareil.
ID de l'appareil	ID de l'appareil.
Utilisateur	Utilisateur pour lequel l'appareil a été ajouté, ou *Utilisateur inaccessible* si cet utilisateur a été supprimé.
Rôle utilisateur	Rôle utilisateur.
Ajouté le	Date et heure auxquelles l'appareil a été ajouté.
Dernière activité	Date et heure auxquelles l'appareil a été actif pour la dernière fois.
État	État de l'appareil : **Nouveau** - l'appareil a été ajouté sans scanner le code QR et devra être ajouté à nouveau. **Nouveau (jeton expiré)** - l'appareil a été ajouté sans scanner le code QR et le jeton du code QR a expiré ; l'appareil devra être ajouté à nouveau. **Actif** - l'appareil a été ajouté, le code QR a été scanné et il peut utiliser l'application mobile Zabbix. **Non attribué** - l'appareil a été ajouté, mais son utilisateur a été supprimé et il ne peut pas utiliser l'application mobile Zabbix.

Les utilisateurs disposant d'autorisations suffisantes peuvent également gérer leurs propres appareils dans *Paramètres utilisateur* > *Appareils*.

Administration

Vue d'ensemble

Le menu Administration est destiné aux fonctions administratives de Zabbix. Ce menu est disponible uniquement pour les utilisateurs du type SuperAdmin.

Source de données

Vue d'ensemble

La section *Administration* > *Source de données* vous permet de spécifier la source des données de télémétrie, c'est-à-dire la base de données qui contient les données de télémétrie (traces, métriques et journaux). Les données de télémétrie sont utilisées pour la surveillance des performances des applications (APM), qui vous aide à voir comment les requêtes circulent dans vos applications et services, à mesurer leurs performances et à trouver la cause des opérations lentes ou ayant échoué.

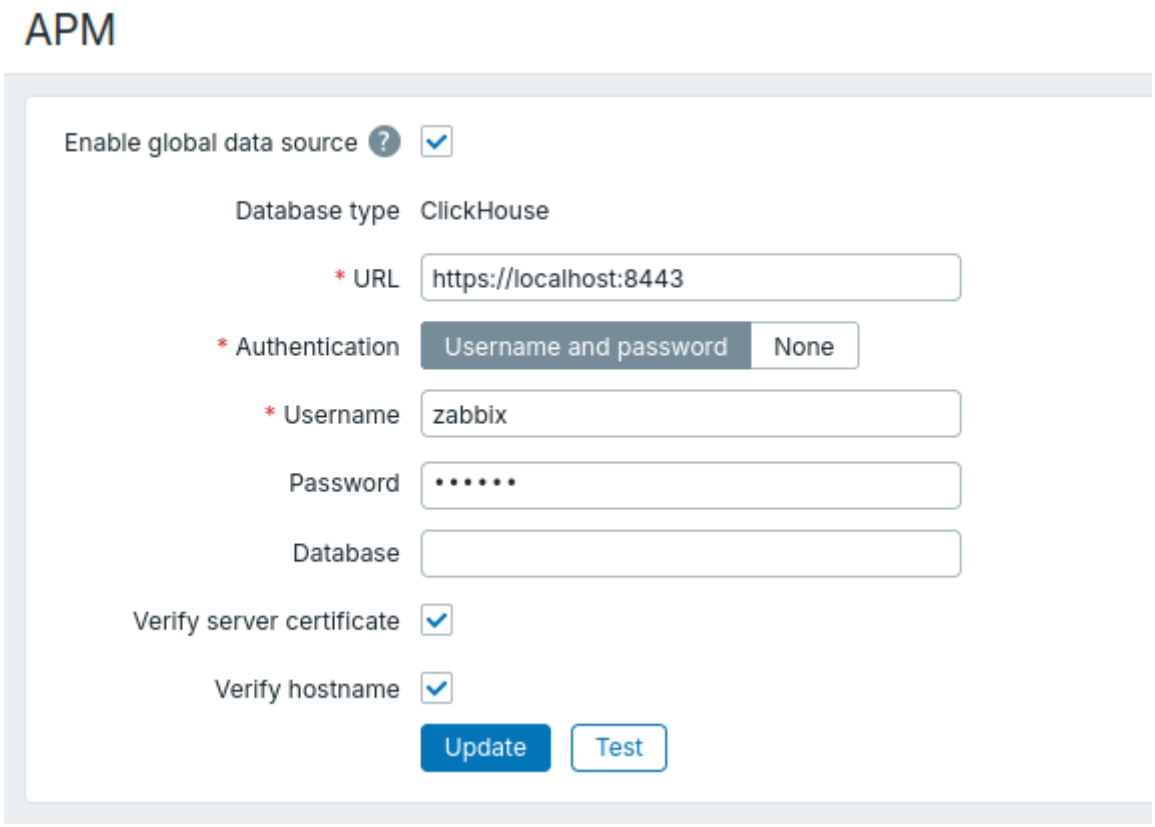
La source des données de télémétrie doit prendre en charge OpenTelemetry.

La source des données de télémétrie spécifiée dans cette section est appelée source de données globale, car elle est partagée par le serveur Zabbix, le proxy et l'interface. Cependant, vous pouvez la remplacer pour chacun de ces composants en spécifiant une autre source de données de télémétrie dans la configuration du serveur, du proxy ou de l'interface.

L'élément telemetry query interroge cette source de données, sauf si une autre source est spécifiée dans la configuration du composant.

APM

Pour spécifier la source de données globale, cochez la case *Activer la source de données globale* et renseignez les paramètres décrits ci-dessous. Avant d'enregistrer, vous pouvez cliquer sur *Tester* pour vérifier la connexion à cette source de données (l'extension php-curl est requise).



Paramètre	Description
Type de base de données	Le type de base de données est défini en dur sur *ClickHouse* et ne peut pas être modifié. Voir également : Prérequis et Configuration de ClickHouse.
URL	Saisissez l'URL HTTP ou HTTPS utilisée pour se connecter à la base de données.
Authentification	Sélectionnez la méthode d'authentification utilisée pour se connecter à la base de données : *Nom d'utilisateur et mot de passe* ou *Aucune*.
Nom d'utilisateur	Indiquez le nom d'utilisateur utilisé pour se connecter à la base de données. Ce paramètre est disponible si *Authentification* est défini sur *Nom d'utilisateur et mot de passe*.
Mot de passe	Indiquez le mot de passe utilisé pour se connecter à la base de données, ou laissez ce champ vide si l'utilisateur de la base de données n'a pas de mot de passe. Le mot de passe est réinitialisé lorsque vous modifiez l'URL. Ce paramètre est disponible si *Authentification* est défini sur *Nom d'utilisateur et mot de passe*.
Base de données	Saisissez le nom de la base de données qui contient les données de télémétrie. Si ce champ est laissé vide, Zabbix utilise la base de données par défaut configurée dans ClickHouse pour l'utilisateur.
Vérifier le certificat du serveur	Cochez la case pour vérifier le certificat du serveur. Si cette case est cochée, les options TLS correspondantes doivent être définies dans la configuration du serveur, du proxy ou de l'interface. Ce paramètre est disponible si l'*URL* contient une URL HTTPS.
Vérifier le nom d'hôte	Cochez la case pour vérifier que le nom d'hôte indiqué dans l'*URL* correspond au nom d'hôte du certificat du serveur. Si cette case est cochée, les options TLS correspondantes doivent être définies dans la configuration du serveur, du proxy ou de l'interface. Ce paramètre est disponible si l'*URL* contient une URL HTTPS.

Général

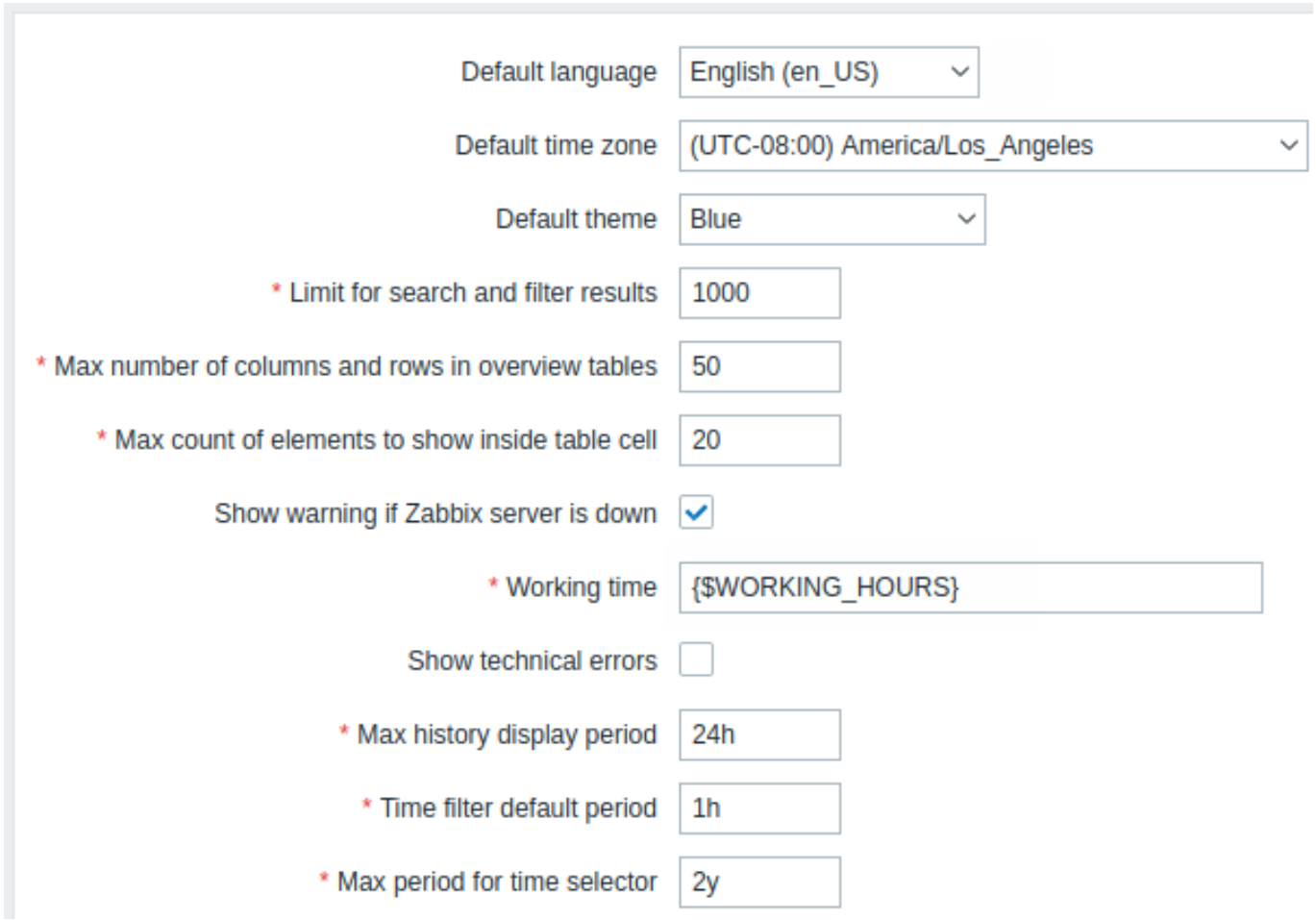
Vue d'ensemble

La section *Administration* > *General* contient un certain nombre de sous-sections permettant de définir les valeurs par défaut liées à l'interface et de personnaliser Zabbix.

La liste des sous-sections disponibles apparaît lorsque vous cliquez sur *General* dans la section de menu *Administration*. Il est également possible de passer d'une sous-section à une autre à l'aide de la liste déroulante du titre dans le coin supérieur gauche.

GUI

Cette section fournit la personnalisation de plusieurs paramètres par défaut liés à l'interface.



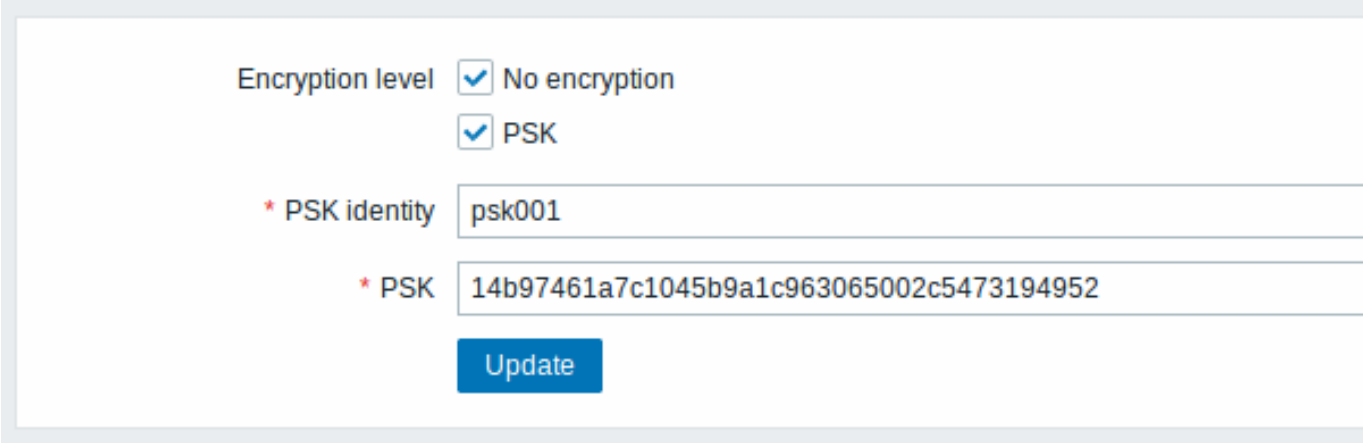
Paramètres de configuration :

Parameter	Description
Default language	Langue par défaut pour les utilisateurs qui n'ont pas spécifié de langue dans leur profil, ainsi que pour les utilisateurs invités. Pour plus d'informations, voir Installation de langues supplémentaires pour l'interface.
Default time zone	Fuseau horaire par défaut pour les utilisateurs qui n'ont pas spécifié de fuseau horaire dans leur profil, ainsi que pour les utilisateurs invités.
Default theme	Thème par défaut pour les utilisateurs qui n'ont pas spécifié de thème dans leur profil, ainsi que pour les utilisateurs invités.
Limit for search and filter results	Quantité maximale d'éléments (lignes) qui seront affichés dans une liste de l'interface web, par exemple dans *Collecte de données* > *Hôtes*. *Note* : si cette valeur est définie, par exemple, sur '50', seuls les 50 premiers éléments seront affichés dans toutes les listes de l'interface concernées. Si une liste contient plus de cinquante éléments, cela sera indiqué par le signe '+' dans *"Affichage de 1 à 50 sur **50+** trouvés"*. De plus, si un filtrage est utilisé et qu'il y a toujours plus de 50 correspondances, seuls les 50 premiers seront affichés. Notez qu'une augmentation de la valeur de ce paramètre peut entraîner une baisse des performances et une augmentation de la consommation mémoire côté serveur web.
Max number of columns and rows in overview tables	Nombre maximal de colonnes et de lignes à afficher dans le widget de tableau de bord *Aperçu des déclencheurs*. La même limite s'applique aux colonnes et aux lignes. Si le nombre de lignes et/ou de colonnes dépasse ce qui est affiché, le système affichera un avertissement en bas du tableau : "Tous les résultats ne sont pas affichés. Veuillez fournir des critères de recherche plus précis."
Max count of elements to show inside table cell	Pour les entrées affichées dans une seule cellule de tableau, le nombre affiché ne dépassera pas la valeur configurée ici.

Parameter	Description
Show warning if Zabbix server is down	Ce paramètre permet d'afficher un message d'avertissement dans une fenêtre du navigateur si le serveur Zabbix est inaccessible (peut-être arrêté). Le message reste visible même si l'utilisateur fait défiler la page vers le bas. Lorsqu'il est survolé, le message est temporairement masqué afin de révéler le contenu situé en dessous.
Working time	Ce paramètre global définit les heures de travail. Dans les graphiques, le temps de travail est affiché sur fond blanc et le temps non travaillé sur fond gris. Voir la page Spécification de la période temporelle pour la description du format de l'heure. Macros utilisateur prises en charge.
Show technical errors	Si cette option est cochée, tous les utilisateurs enregistrés pourront voir les erreurs techniques (PHP/SQL). Si elle n'est pas cochée, ces informations ne sont disponibles que pour les Zabbix Super Admins et les utilisateurs appartenant à des groupes d'utilisateurs pour lesquels le mode débogage est activé.
Max history display period	Période maximale pendant laquelle afficher les données historiques dans *Surveillance* > *Dernières données*, les informations de web scenario de l'hôte dans *Surveillance* > *Hôtes*, ainsi que dans le widget de tableau de bord *Top items*. Plage autorisée : 24 heures (par défaut) - 1 semaine. Les suffixes de temps, par exemple 1w (une semaine), 36h (36 heures), sont pris en charge.
Time filter default period	Période temporelle utilisée par défaut dans les graphiques et les tableaux de bord. Plage autorisée : 1 minute - 10 ans (par défaut : 1 heure). Les suffixes de temps, par exemple 10m (dix minutes), 5w (cinq semaines), sont pris en charge. Note : lorsqu'un utilisateur modifie la période temporelle lors de l'affichage d'un graphique, cette période est enregistrée comme préférence utilisateur, remplaçant la valeur globale par défaut ou une sélection utilisateur précédente.
Max period for time selector	Période temporelle maximale disponible pour les graphiques et les tableaux de bord. Les utilisateurs ne pourront pas visualiser les données plus anciennes. Plage autorisée : 1 an - 10 ans (par défaut : 2 ans). Les suffixes de temps, par exemple 1y (un an), 365w (365 semaines), sont pris en charge.

Autoregistration

Dans cette section, vous pouvez configurer le niveau de chiffrement pour l'autoregistration active de l'agent.



Les paramètres marqués d'un astérisque sont obligatoires.

Paramètres de configuration :

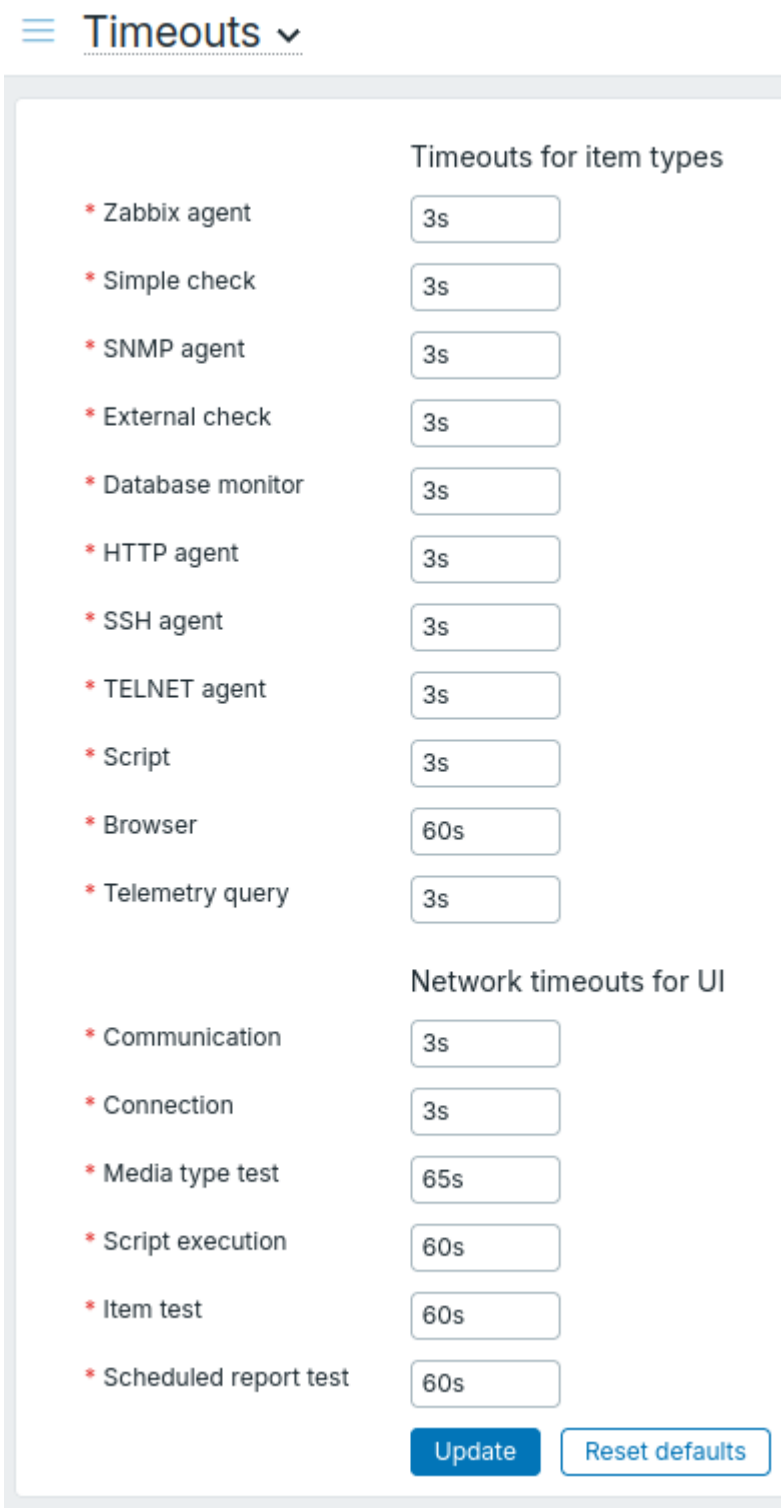
Parameter	Description
Encryption level	Sélectionnez une ou les deux options de niveau de chiffrement : **No encryption** - les connexions non chiffrées sont autorisées **PSK** - les connexions chiffrées TLS avec une clé prépartagée sont autorisées
PSK identity	Saisissez la chaîne d'identité de la clé prépartagée. Ce champ n'est disponible que si 'PSK' est sélectionné comme *Encryption level*. Ne placez pas d'informations sensibles dans l'identité PSK, car elle est transmise non chiffrée sur le réseau afin d'indiquer au destinataire quelle PSK utiliser.

Parameter	Description
PSK	Saisissez la clé prépartagée (un nombre pair de caractères hexadécimaux). Longueur maximale : 512 chiffres hexadécimaux (PSK de 256 octets) si Zabbix utilise la bibliothèque GnuTLS ou OpenSSL, 64 chiffres hexadécimaux (PSK de 32 octets) si Zabbix utilise la bibliothèque mbed TLS (PolarSSL). Exemple : 1f87b595725ac58dd977beef14b97461a7c1045b9a1c963065002c5473194952 Ce champ n'est disponible que si 'PSK' est sélectionné comme *Encryption level*.

Voir aussi : Autoregistration sécurisée

Délais d'attente

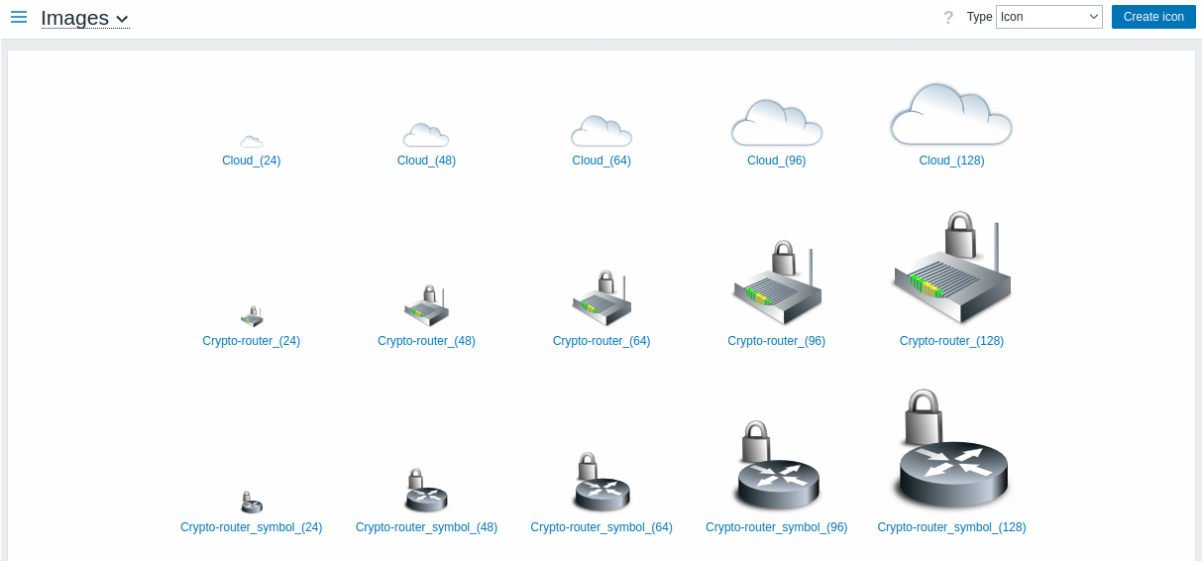
Dans cette section, il est possible de définir les délais d'attente globaux pour les types d'éléments et les délais d'attente réseau. Tous les champs de ce formulaire sont obligatoires.



Parameter		Description
Délais d'attente pour les types d'éléments		Combien de secondes Zabbix doit attendre avant d'interrompre la vérification (selon son type). Pour plus de détails et les limitations, voir délai d'attente d'un élément. Plage autorisée: 1 - 600 s (par défaut: 3 s; par défaut pour les éléments browser: 60 s). Les suffixes de temps, par exemple 30s, 1m, ainsi que les macros utilisateur sont pris en charge. Notez que les délais d'attente au niveau du proxy, s'ils sont configurés, remplacent ces paramètres globaux. De plus, si des délais d'attente sont configurés pour des éléments spécifiques, ils remplacent à la fois les paramètres globaux et ceux du proxy.
Délais d'attente réseau pour l'interface		
	Communication	Combien de secondes attendre avant de fermer un socket inactif (si une connexion au serveur Zabbix a déjà été établie, mais que l'interface ne peut pas terminer l'opération de lecture/envoi des données pendant ce délai, la connexion sera interrompue). Plage autorisée: 1 - 300 s (par défaut: 3 s).
	Connexion	Combien de secondes attendre avant d'arrêter une tentative de connexion au serveur Zabbix. Plage autorisée: 1 - 30 s (par défaut: 3 s).
	Test du type de média	Combien de secondes attendre une réponse lors du test d'un type de média. Plage autorisée: 1 - 300 s (par défaut: 65 s).
	Exécution de script	Combien de secondes attendre une réponse lors de l'exécution d'un script. Plage autorisée: 1 - 300 s (par défaut: 60 s). Ce délai d'attente s'applique à l'ensemble de la chaîne d'exécution du script, qui peut avoir une longueur variable. Par exemple, si le script est exécuté sur l'agent, il s'agit d'un aller-retour via le serveur (éventuellement aussi le proxy) vers l'agent et retour. Pour les scripts exécutés sur l'agent, le délai d'attente du serveur ou du proxy s'applique. Pour les scripts exécutés uniquement sur un agent actif, il est probable que le délai d'attente par défaut du serveur/proxy doive être augmenté. Le délai d'attente du serveur/proxy doit être supérieur à la fréquence d'actualisation de la vérification active, sinon il peut être dépassé avant même que l'agent actif ne reçoive le script. Voir aussi: Délai d'attente du script.
	Test d'élément	Combien de secondes attendre les données renvoyées lors du test d'un élément. Plage autorisée: 1 - 600 s (par défaut: 60 s).
	Test de rapport planifié	Combien de secondes attendre les données renvoyées lors du test d'un rapport planifié. Plage autorisée: 1 - 300 s (par défaut: 60 s).

Images

La section Images affiche toutes les images disponibles dans Zabbix. Les images sont stockées dans la base de données.

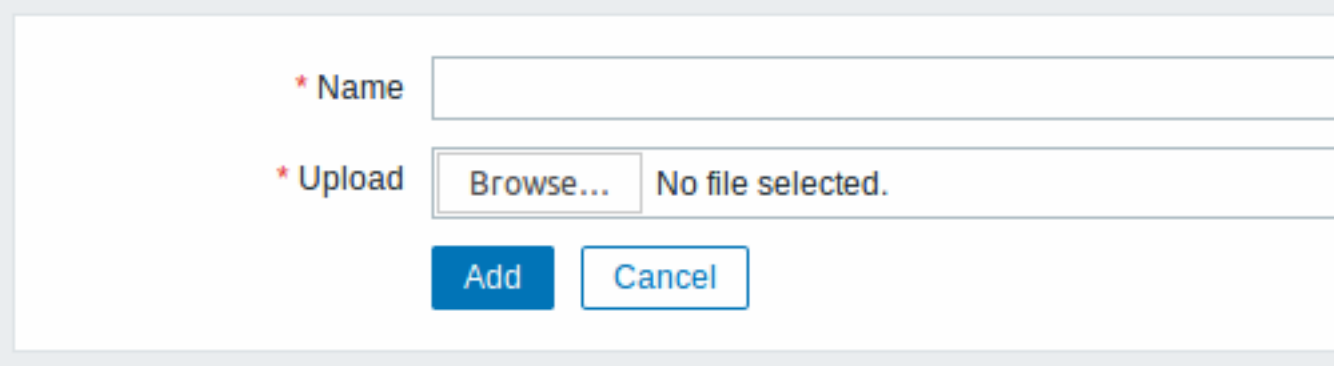


La liste déroulante *Type* permet de choisir entre les icônes et les images d'arrière-plan :

- Les icônes sont utilisées pour afficher les éléments des cartes réseau.
- Les arrière-plans sont utilisés comme images d'arrière-plan des cartes réseau.

Ajout d'une image

Vous pouvez ajouter votre propre image en cliquant sur le bouton *Créer une icône* ou *Créer un arrière-plan* dans le coin supérieur droit.



Attributs de l'image :

Paramètre	Description
Nom	Nom unique d'une image.
Téléchargement	Sélectionnez le fichier (PNG, JPEG, GIF ou WebP) sur un système local pour le télécharger vers Zabbix. *Notez* qu'il peut être possible de télécharger d'autres formats qui seront convertis au format PNG lors du téléchargement. La bibliothèque GD est utilisée pour le traitement des images ; les formats pris en charge dépendent donc de la version de la bibliothèque utilisée (Zabbix nécessite la version 2.0.28 ou ultérieure).

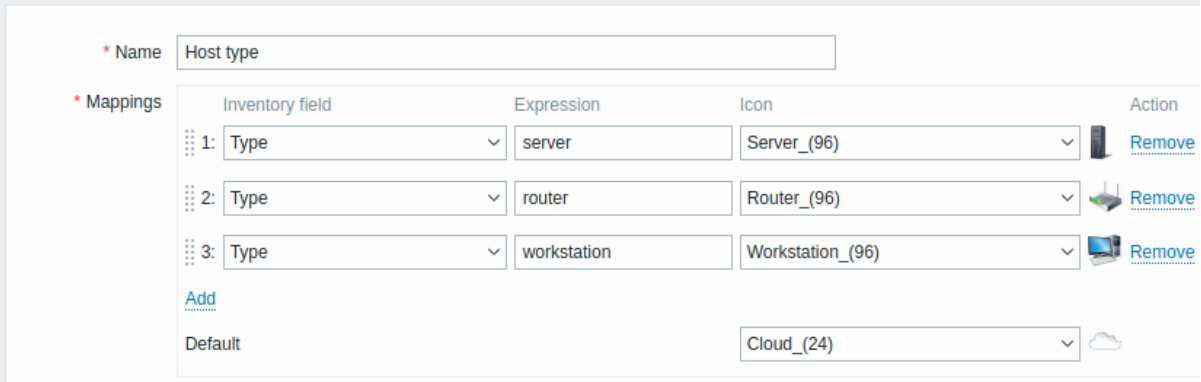
Note:
La taille maximale du fichier à télécharger est limitée par la valeur de `ZBX_MAX_IMAGE_SIZE`, qui est de 1024x1024 octets, soit 1 Mo.
Le téléchargement d'une image peut échouer si sa taille est proche de 1 Mo et que le paramètre de configuration MySQL `max_allowed_packet` est défini sur la valeur par défaut de 1 Mo. Dans ce cas, augmentez la valeur du paramètre `max_allowed_packet`.

Mappage d'icônes

Cette section permet de créer le mappage de certains hôtes avec certaines icônes. Les informations du champ d'inventaire de l'hôte sont utilisées pour créer le mappage.

Les mappages peuvent ensuite être utilisés dans la configuration de la carte réseau pour attribuer automatiquement les icônes appropriées aux hôtes correspondants.

Pour créer un nouveau mappage d'icônes, cliquez sur *Créer un mappage d'icônes* dans le coin supérieur droit.



Paramètres de configuration :

Parameter	Description
Name	Nom unique du mappage d'icônes.
Mappings	Liste des mappages. L'ordre des mappages détermine celui qui aura la priorité. Vous pouvez déplacer les mappages vers le haut ou vers le bas de la liste par glisser-déposer.
Inventory field	Champ d'inventaire de l'hôte qui sera examiné pour rechercher une correspondance.
Expression	Expression régulière décrivant la correspondance.
Icon	Icône à utiliser si une correspondance pour l'expression est trouvée.
Default	Icône par défaut à utiliser.

Expressions régulières

Cette section permet de créer des expressions régulières personnalisées qui peuvent être utilisées à plusieurs endroits dans l'interface. Consultez la section Expressions régulières pour plus de détails.

Options d'affichage des déclencheurs

Cette section permet de personnaliser la manière dont l'état du déclencheur est affiché dans l'interface et les noms et couleurs des niveaux de gravité des déclencheurs.

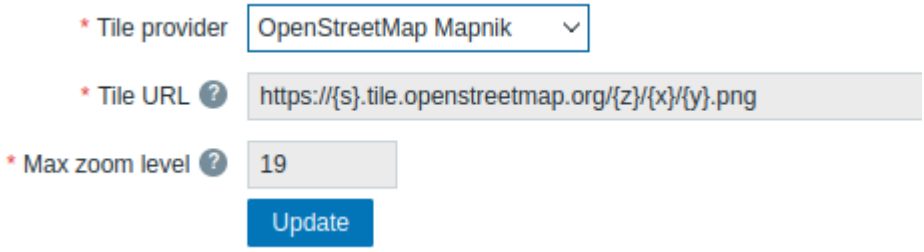


Parameter	Description
Use custom event status colors	Cocher ce paramètre active la personnalisation des couleurs pour les problèmes reconnus/non reconnus.
Unacknowledged PROBLEM events, *Acknowledged PROBLEM events*, *Unacknowledged RESOLVED events*, *Acknowledged RESOLVED events*	Saisissez un nouveau code couleur ou cliquez sur la couleur pour en sélectionner une nouvelle dans la palette fournie. Si la case *blinking* est cochée, les déclencheurs clignoteront pendant un certain temps lors du changement d'état afin d'être plus visibles.
Display OK triggers for	Période d'affichage des déclencheurs OK. Plage autorisée: 0 - 24 heures. Les suffixes de temps, par exemple 5m, 2h, 1d, sont pris en charge.
On status change triggers blink for	Durée du clignotement du déclencheur. Plage autorisée: 0 - 24 heures. Les suffixes de temps, par exemple 5m, 2h, 1d, sont pris en charge.

Parameter	Description
Not classified, *Information*, *Warning*, *Average*, *High*, *Disaster*	Noms et/ou couleurs personnalisés des niveaux de gravité à afficher à la place des valeurs par défaut du système. Saisissez un nouveau code couleur ou cliquez sur la couleur pour en sélectionner une nouvelle dans la palette fournie. Notez que les noms de niveaux de gravité personnalisés saisis ici seront utilisés dans toutes les langues. Si vous devez les traduire dans d'autres langues pour certains utilisateurs, consultez la page Personnalisation des niveaux de gravité des déclencheurs.

Cartes géographiques

Cette section permet de sélectionner le fournisseur de service de tuiles de carte géographique et de configurer les paramètres du fournisseur de service pour le widget de tableau de bord Geomap interface. Pour fournir une visualisation à l'aide des cartes géographiques, Zabbix utilise la bibliothèque JavaScript open source de cartes interactives Leaflet. Veuillez noter que Zabbix n'a aucun contrôle sur la qualité des images fournies par des fournisseurs de tuiles tiers, y compris les fournisseurs de tuiles prédéfinis.



Parameter	Description
Tile provider	Sélectionnez l'un des fournisseurs de service de tuiles disponibles ou sélectionnez *Other* pour ajouter un autre fournisseur de tuiles ou des tuiles auto-hébergées (voir Utilisation d'un fournisseur de service de tuiles personnalisé).
Tile URL	Le modèle d'URL (jusqu'à 2048 caractères) pour charger et afficher la couche de tuiles sur les cartes géographiques. Ce champ est modifiable uniquement si *Tile provider* est défini sur *Other*. Les espaces réservés suivants sont pris en charge : *{s}* représente l'un des sous-domaines disponibles ; *{z}* représente le paramètre de niveau de zoom dans l'URL ; *{x}* et *{y}* représentent les coordonnées de la tuile ; *{r}* peut être utilisé pour ajouter "@2x" à l'URL afin de charger des tuiles retina. Exemple : `https://{s}.example.com/{z}/{x}/{y}{r}.png`
Attribution text	Texte d'attribution du fournisseur de tuiles à afficher dans une petite zone de texte sur la carte. Ce champ est visible uniquement si *Tile provider* est défini sur *Other*.
Max zoom level	Niveau de zoom maximal de la carte. Ce champ est modifiable uniquement si *Tile provider* est défini sur *Other*.

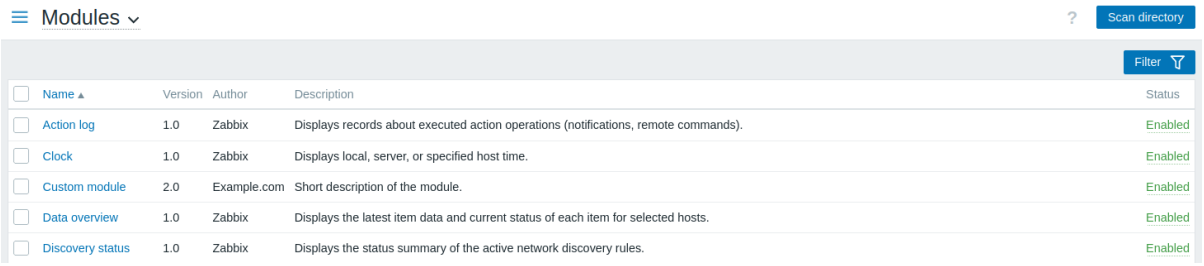
Utilisation d'un fournisseur de service de tuiles personnalisé

Le widget Geomap est capable de charger des images de tuiles raster à partir d'un service de fournisseur de tuiles personnalisé auto-hébergé ou tiers. Pour utiliser un service de fournisseur de tuiles tiers personnalisé ou un dossier ou serveur de tuiles auto-hébergé, sélectionnez *Other* dans le champ *Tile provider* et spécifiez l'URL personnalisée dans le champ *Tile URL* en utilisant les espaces réservés appropriés.

Modules

Cette section vous permet d'administrer les modules de l'interface personnalisés ainsi que ceux intégrés.

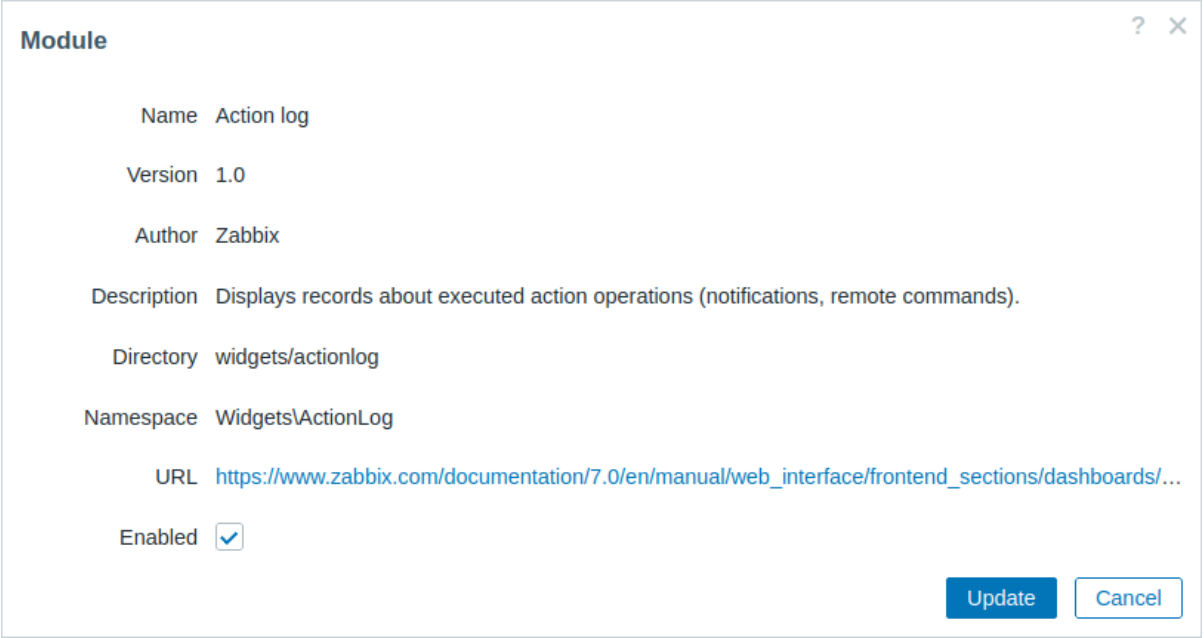
> **Note:**
> Si l'indicateur de fonctionnalité `$ZBX_FEATURE_FLAGS['modules_config_enabled']` est défini sur `false` dans le fichier de configuration de l'interface, l'accès à cette section est désactivé.



Cliquez sur *Analyser le répertoire* pour enregistrer ou désenregistrer les modules personnalisés. Les modules enregistrés apparaîtront dans la liste ; les modules désenregistrés seront supprimés de la liste.

Cliquez sur l'état du module dans la liste pour activer ou désactiver un module. Vous pouvez également activer ou désactiver plusieurs modules à la fois en les sélectionnant dans la liste, puis en cliquant sur les boutons *Activer/Désactiver* situés sous la liste.

Cliquez sur le nom du module dans la liste pour afficher ses détails dans une fenêtre contextuelle.

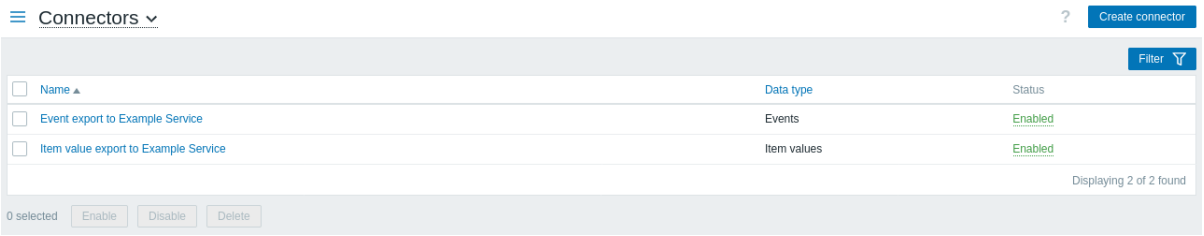


L'état du module peut également être mis à jour dans la fenêtre contextuelle des détails du module ; pour ce faire, cochez ou décochez la case *Activé*, puis cliquez sur *Mettre à jour*.

Vous pouvez filtrer les modules par nom ou par état (activé/désactivé).

Connecteurs

Cette section vous permet de configurer des connecteurs pour le streaming des données Zabbix vers des systèmes externes via HTTP.



Cliquez sur *Créer un connecteur* pour configurer un nouveau connecteur.

Vous pouvez filtrer les connecteurs par nom ou par état (activé/désactivé). Cliquez sur l'état du connecteur dans la liste pour activer/désactiver un connecteur. Vous pouvez également activer/désactiver plusieurs connecteurs à la fois en les sélectionnant dans la liste, puis en cliquant sur les boutons *Activer/Désactiver* situés sous la liste.

Autres

Cette section permet de configurer divers autres paramètres de l'interface.



Paramètre	Description
URL de l'interface	URL (jusqu'à 2048 caractères) de l'interface web Zabbix. Ce paramètre est utilisé par le service web Zabbix pour communiquer avec l'interface et doit être spécifié pour activer les rapports planifiés.
Groupe pour les hôtes découverts	Les hôtes découverts par la découverte réseau et l'autoregistration de l'agent seront automatiquement placés dans le groupe d'hôtes sélectionné ici.
Mode d'inventaire de l'hôte par défaut	Mode par défaut pour l'inventaire de l'hôte. Il sera appliqué chaque fois qu'un nouvel hôte ou prototype d'hôte est créé par le serveur ou l'interface, sauf s'il est remplacé lors de la découverte/autoregistration de l'hôte par l'opération *Définir le mode d'inventaire de l'hôte*.

Paramètre	Description
Groupe d'utilisateurs pour le message de base de données indisponible	Groupe d'utilisateurs pour l'envoi du message d'alarme indiquant que la base de données est indisponible. Le serveur Zabbix dépend de la disponibilité de la base de données backend. Il ne peut pas fonctionner sans base de données. Si la base de données est indisponible, les utilisateurs sélectionnés peuvent être avertis par Zabbix. Les notifications seront envoyées au groupe d'utilisateurs défini ici en utilisant les entrées de média utilisateur activées. Les notifications sont transmises à l'aide des types de média suivants (lorsqu'ils sont activés) : e-mail ; SMS ; scripts d'alerte personnalisés. Même si une entrée de média webhook est configurée et activée, elle ne déclenchera pas de notifications. Le serveur Zabbix ne s'arrêtera pas ; il attendra que la base de données soit de nouveau disponible pour reprendre le traitement. La notification se compose du contenu suivant : `[MySQL|PostgreSQL] database <DB Name> [on <DB Host>:<DB Port>] is not available: <error message depending on the type of DBMS (database)>` <DB Host> n'est pas ajouté au message s'il est défini comme une valeur vide et <DB Port> n'est pas ajouté s'il s'agit de la valeur par défaut (”0”). Le gestionnaire d'alertes (un processus spécial du serveur Zabbix) tente d'établir une nouvelle connexion à la base de données toutes les 10 secondes. Si la base de données est toujours indisponible, le gestionnaire d'alertes répète l'envoi des alertes, mais pas plus souvent que toutes les 15 minutes.
Journaliser les traps SNMP sans correspondance	Journaliser un trap SNMP si aucune interface SNMP correspondante n'a été trouvée.

Autorisation

Paramètre	Description
Tentatives de connexion	Nombre de tentatives de connexion infructueuses avant que la possibilité de se connecter ne soit bloquée.
Intervalle de blocage de connexion	Période pendant laquelle la connexion sera interdite lorsque la limite de *Tentatives de connexion* est dépassée. Plage autorisée : 30 - 3600 secondes. Les suffixes de temps, par exemple 90s, 5m, 1h, sont pris en charge.

Stockage des secrets

Parameter	Description
Vault provider	Sélectionnez le logiciel de gestion des secrets pour stocker les valeurs des macros utilisateur - *HashiCorp Vault* (par défaut) ou *CyberArk Vault*.
Resolve secret vault macros by	Résoudre les macros du coffre-fort de secrets par : **serveur Zabbix** - les secrets sont récupérés par le serveur Zabbix et transmis aux proxies si nécessaire (par défaut) ; **serveur Zabbix et proxy** - les secrets sont récupérés à la fois par le serveur Zabbix et les proxies, ce qui leur permet de résoudre les macros de manière indépendante.

Voir aussi : Stockage des secrets.

Sécurité

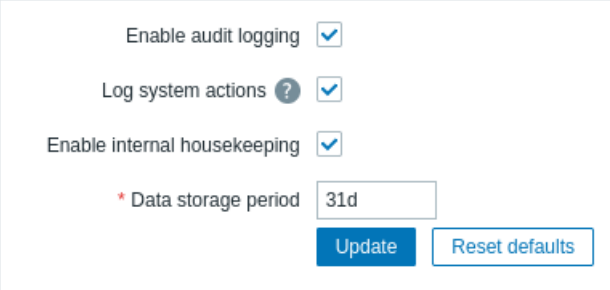
Paramètre	Description
Valider les schémas d'URI	Décochez cette case pour désactiver la validation des schémas d'URI (activée par défaut). Si elle est cochée, vous pouvez spécifier une liste de schémas d'URI autorisés, séparés par des virgules (par défaut : http,https,ftp,file,mailto,tel,ssh). Cela s'applique à tous les champs du frontend où des URI sont utilisés (par exemple, les URL des éléments de carte).

Paramètre	Description
Utiliser l'en-tête HTTP X-Frame-Options	Décochez cette case pour désactiver l'en-tête HTTP X-Frame-options (non recommandé). Si elle est cochée, vous pouvez spécifier la valeur de l'en-tête HTTP X-Frame-options. Valeurs prises en charge : **SAMEORIGIN** (par défaut) ou **'self'** (doit être entre apostrophes) - la page ne peut être affichée dans un cadre que sur la même origine que la page elle-même ; **DENY** ou **'none'** (doit être entre apostrophes) - la page ne peut pas être affichée dans un cadre, quel que soit le site qui tente de le faire ; **une chaîne de noms d'hôte séparés par des espaces** ; l'ajout de **'self'** (doit être entre apostrophes) à la liste permet à la page d'être affichée dans un cadre sur la même origine que la page elle-même. Notez que l'utilisation de **'self'** ou **'none'** sans apostrophes aura pour conséquence qu'ils seront considérés comme des noms d'hôte.
Utiliser le sandboxing des iframe	Décochez cette case pour désactiver le placement du contenu de l'URL récupérée dans un sandbox (non recommandé). Si elle est cochée, vous pouvez spécifier les exceptions de sandboxing de l'iframe ; les restrictions non spécifiées seront toujours appliquées. Si ce champ est vide, toutes les restrictions de l'attribut sandbox s'appliquent. Pour plus d'informations, consultez la description de l'attribut `sandbox`.

Journal d'audit

Aperçu

Cette section permet de configurer les paramètres du journal d'audit.



Les paramètres suivants sont disponibles :

Paramètre	Description
Activer la journalisation d'audit	Activer (par défaut) ou désactiver la journalisation d'audit.
Journaliser les actions système	Activer (par défaut) ou désactiver la journalisation d'audit des activités de découverte de bas niveau, de découverte réseau et d'auto-enregistrement effectuées par le serveur (utilisateur système).
Activer le nettoyage interne	Activer (par défaut) ou désactiver le nettoyage interne des enregistrements du journal d'audit.
Période de conservation des données	Nombre de jours pendant lesquels les enregistrements du journal d'audit doivent être conservés avant d'être supprimés par le processus de nettoyage. Obligatoire si le nettoyage est activé. Par défaut : 31 jours.

Les paramètres de compression du journal d'audit sont disponibles dans la section *Administration* > *Housekeeping*, dans le bloc *Compression de l'historique, des tendances et du journal d'audit*, qui devient visible lorsque TimescaleDB est utilisé.

Maintenance

Vue d'ensemble

La section *Administration* > *Housekeeping* vous permet de configurer le housekeeping interne de Zabbix, qui supprime automatiquement de la base de données les données obsolètes ou supprimées par l'utilisateur. Cela empêche la base de données de croître indéfiniment, ce qui consommerait inutilement de l'espace disque et dégraderait les performances de la base de données.

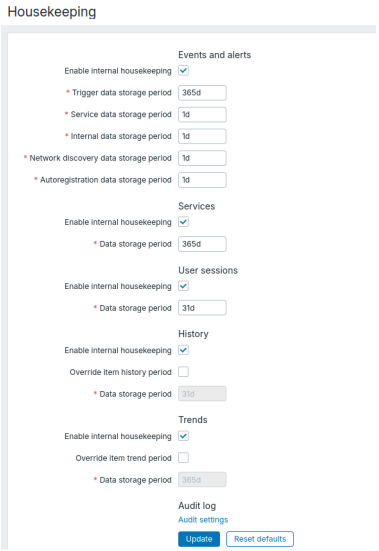
Le housekeeping peut être activé et configuré pour les types de données suivants :

- Événements et alertes provenant des déclencheurs, des services, des données internes, de la découverte réseau et de l'enregistrement automatique
- Services
- Sessions utilisateur
- Historique et tendances pour les éléments

Note:
Le housekeeping du journal d'audit est configuré séparément.

Configuration

Les paramètres suivants sont disponibles :



Parameter	Description
Enable internal housekeeping	Activer ou désactiver le housekeeping interne (activé par défaut). Si cette option est activée, la procédure de housekeeping supprime automatiquement de la base de données les données qui dépassent la *Data storage period*.
Data storage period	Spécifiez la durée de conservation des données avant leur suppression par le housekeeper. Obligatoire si le housekeeping interne est activé. Plage : 1 jour (1 heure pour l'historique) - 25 ans ; ou "0". Les suffixes de temps (par exemple, 1d, 1w) sont pris en charge. Pour *Events and alerts*, la période de conservation des données est définie séparément pour les déclencheurs, les services, les données internes, la découverte réseau et l'autenregistrement. Le housekeeper supprime uniquement les événements qui ne sont pas associés à des problèmes. Par exemple, un événement de problème/rétablissement plus ancien que la *Data storage period* ne sera pas supprimé s'il est associé à un problème non résolu. Lorsque le housekeeper supprime des entités obsolètes, il supprime d'abord les problèmes, puis les événements. Pour *History* et *Trends*, les périodes de conservation des données déterminent également combien de temps les données restent visibles dans *Monitoring* > *Latest data*, même si le housekeeping interne est désactivé. Notez que lorsqu'un élément ou un déclencheur est supprimé, ses problèmes et les événements associés sont également supprimés.
Override item history period	Si cette option est activée, la période de conservation de l'historique/des tendances spécifiée dans la configuration de l'élément est remplacée par le paramètre *Data storage period* (sauf pour les éléments avec l'option *Do not store* activée). Cette option peut être utilisée même lorsque le housekeeping interne est désactivé et qu'un housekeeper externe est utilisé.
Override item trend period	

Si vous utilisez **ClickHouse**, la section *ClickHouse data storage period* devient disponible. Elle affiche la période de conservation des données pour chaque type de valeur d'élément. Pour modifier cette période, consultez Configuration de la période de conservation des données ClickHouse. Notez que le housekeeper ne supprime pas les données de ClickHouse.

Si vous utilisez **Elasticsearch**, la section *Elasticsearch data storage period* devient disponible. Elle affiche les types de valeurs d'élément enregistrés dans Elasticsearch, avec la période de conservation des données affichée comme *Unknown*. Pour contrôler la durée de conservation des données, consultez Index Lifecycle Management. Notez que le housekeeper ne supprime pas les données d'Elasticsearch.

Si vous utilisez **TimescaleDB**, la section *History, trends and audit log compression* devient disponible.

Note:
Pour TimescaleDB, activez *Override item history period*, *Override item trend period* et *Enable internal housekeeping* pour l'historique et les tendances afin de profiter pleinement du partitionnement automatique. Si ces options sont désactivées, les données conservées dans les tables d'historique et de tendances seront tout de même partitionnées, mais le housekeeper ne supprimera pas les partitions obsolètes, et des avertissements de configuration apparaîtront. Lorsque la suppression des partitions obsolètes est activée, le serveur Zabbix et l'interface ne suivront plus les éléments supprimés, et l'historique de ces éléments sera effacé lorsqu'une partition obsolète sera supprimée.

Le housekeeper est exécuté automatiquement toutes les heures par défaut. Vous pouvez modifier la fréquence d'exécution en définissant le paramètre de configuration du serveur Zabbix `HousekeepingFrequency`.

Le bouton *Reset defaults* vous permet d'annuler toutes les modifications apportées.

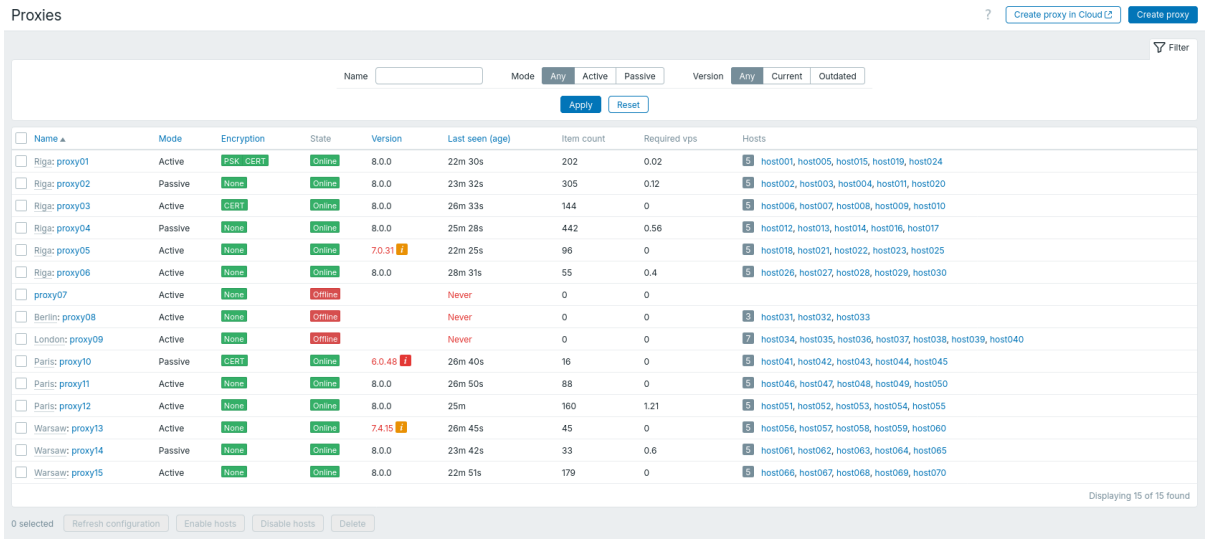
Proxies

Vue d'ensemble

Dans la section *Administration* > *Proxies*, les proxy pour la supervision distribuée peuvent être configurés dans l'interface Zabbix.

Proxies

Une liste des proxies existants avec leurs détails est affichée.



Données affichées :

Colonne	Description
Nom	Nom du proxy. Cliquer sur le nom du proxy ouvre le formulaire de configuration du proxy. Si le proxy appartient à un groupe de proxies, le nom du groupe est affiché avant le nom du proxy sous forme de lien gris. Cliquer sur le nom du groupe ouvre le formulaire de configuration du groupe de proxies.
Mode	Mode du proxy - *Actif* ou *Passif*.
Chiffrement	État du chiffrement des connexions depuis le proxy : **Aucun** - aucun chiffrement ; **PSK** - utilisation d'une clé prépartagée ; **Cert** - utilisation d'un certificat.

Colonne	Description
État	État du proxy : **Inconnu** - le proxy a été créé alors que le serveur Zabbix était arrêté, ou le serveur n'a pas encore mis à jour l'état ; **En ligne** - le proxy a communiqué avec le serveur Zabbix (le proxy passif a répondu à une requête du serveur ; le proxy actif a envoyé une requête) pendant la période de basculement ; **Hors ligne** - le proxy n'a pas communiqué avec le serveur Zabbix pendant la période de basculement.
Version	Version du proxy (numéro de version à trois chiffres). Si le proxy est obsolète ou non pris en charge, le numéro de version est mis en évidence (en rouge) et une icône d'état informatif (jaune ou rouge) est affichée. Survolez l'icône pour afficher les détails.
Dernière détection (ancienneté)	Moment où le proxy a été détecté pour la dernière fois par le serveur.
Nombre d'éléments	Nombre d'éléments activés sur les hôtes activés affectés au proxy.
VPS requis	Performances requises du proxy (nombre de valeurs devant être collectées par seconde).
Hôtes	Nombre d'hôtes activés affectés au proxy et liste des hôtes surveillés par le proxy. Cliquer sur le nom de l'hôte ouvre le formulaire de configuration de l'hôte.

Pour configurer un nouveau proxy, cliquez sur le bouton *Créer un proxy* dans le coin supérieur droit. Pour déployer un nouveau proxy dans Zabbix Cloud, cliquez sur le bouton *Créer un proxy dans le Cloud* situé à côté. Zabbix Cloud s'ouvre dans un nouvel onglet du navigateur, où vous pouvez créer un nœud proxy. Une fois le nœud proxy créé, configurez le proxy à l'aide du bouton *Créer un proxy*. Pour plus d'informations, consultez Configurer un proxy Zabbix Cloud.

Options de modification en masse

Les boutons sous la liste offrent plusieurs options de modification en masse :

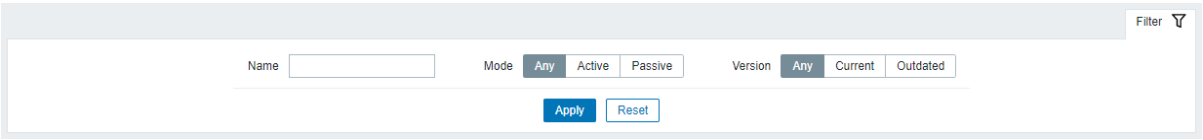
- *Actualiser la configuration* - actualiser la configuration des proxies ;
- *Activer les hôtes* - changer l'état des hôtes surveillés par le proxy en *Surveillé* ;
- *Désactiver les hôtes* - changer l'état des hôtes surveillés par le proxy en *Non surveillé* ;
- *Supprimer* - supprimer les proxies.

Pour utiliser ces options, cochez les cases avant les proxies concernés, puis cliquez sur le bouton requis.

Utilisation du filtre

Vous pouvez utiliser le filtre pour afficher uniquement les proxies qui vous intéressent. Pour de meilleures performances de recherche, les données sont recherchées avec les macros non résolues.

Le lien *Filtre* est disponible au-dessus de la liste des proxies. Si vous cliquez dessus, un filtre devient disponible et vous permet de filtrer les proxies par nom, mode et version. Notez que l'option de filtre *Obsolète* affiche à la fois les proxies obsolètes (partiellement pris en charge) et les proxies non pris en charge.



Groupes de proxy

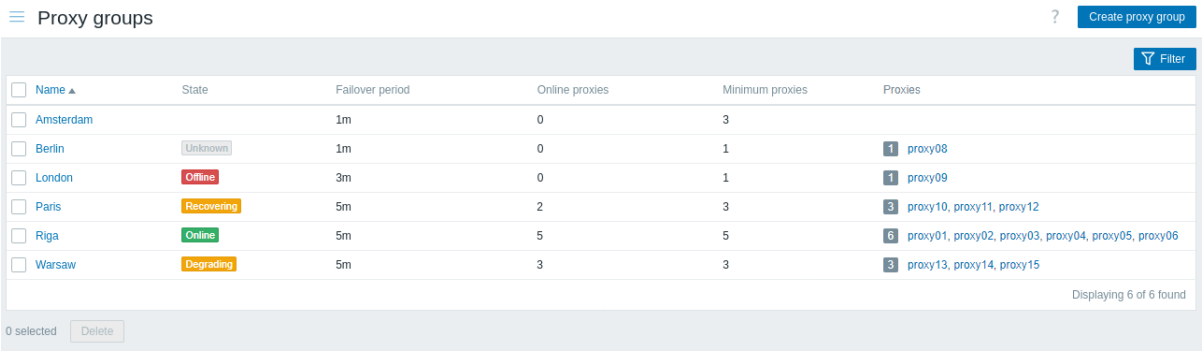
Vue d'ensemble

Dans *Administration* > *Groupes de proxy*, il est possible de configurer des groupes de proxy.

Les groupes de proxy sont utilisés dans l'équilibrage de charge des proxy avec une répartition automatisée des hôtes entre les proxy et une haute disponibilité entre les proxy.

Groupes de proxy

Une liste des groupes de proxy existants avec leurs détails s'affiche.



Données affichées :

Colonne	Description
Nom	Nom du groupe de proxy. Cliquer sur le nom du groupe de proxy ouvre le formulaire de configuration du groupe de proxy.
État	État du groupe de proxy (affiché si le groupe contient au moins un proxy) : **Inconnu** - le groupe a été créé alors que le serveur Zabbix était arrêté, ou le serveur n'a pas encore mis à jour l'état ; **En ligne** - le nombre minimal de proxy a communiqué avec le serveur Zabbix pendant la période de basculement ; **Dégradation** - le groupe va bientôt être mis hors ligne, car le nombre de proxy en ligne est inférieur au seuil *Nombre minimal de proxy* ; **Hors ligne** - moins que le nombre minimal de proxy a communiqué avec le serveur Zabbix pendant la période de basculement ; **Récupération** - le groupe va bientôt repasser en ligne, car le nombre de proxy en ligne a atteint le seuil *Nombre minimal de proxy*. L'état du groupe de proxy peut être surveillé avec les éléments zabbix[proxy group,<name>,state] et zabbix[proxy group,discovery].
Période de basculement	Période pendant laquelle un proxy du groupe de proxy doit communiquer avec le serveur Zabbix pour être considéré comme étant en ligne.
Proxy en ligne	Nombre de proxy en ligne (affiché en rouge s'il est inférieur au minimum du groupe).
Nombre minimal de proxy	Nombre minimal de proxy en ligne requis pour maintenir le groupe de proxy en ligne.
Proxy	Nombre de proxy dans le groupe et liste des proxy du groupe, avec des liens vers le formulaire de configuration du proxy. Le nombre maximal de proxy affichés est limité par la valeur *Nombre maximal d'éléments à afficher dans une cellule de tableau*.

Pour configurer un nouveau groupe de proxy, cliquez sur le bouton *Créer des groupes de proxy* dans le coin supérieur droit.

Options de modification en masse

Les boutons sous la liste offrent plusieurs options de modification en masse :

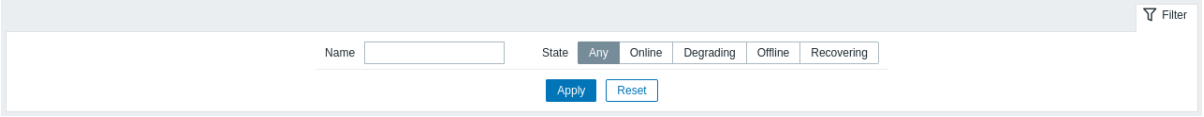
- *Supprimer* - supprimer les groupes de proxy.

Pour utiliser ces options, cochez les cases devant les groupes de proxy concernés, puis cliquez sur le bouton requis.

Utilisation du filtre

Vous pouvez utiliser le filtre pour afficher uniquement les groupes de proxy qui vous intéressent. Pour de meilleures performances de recherche, les données sont recherchées avec les macros non résolues.

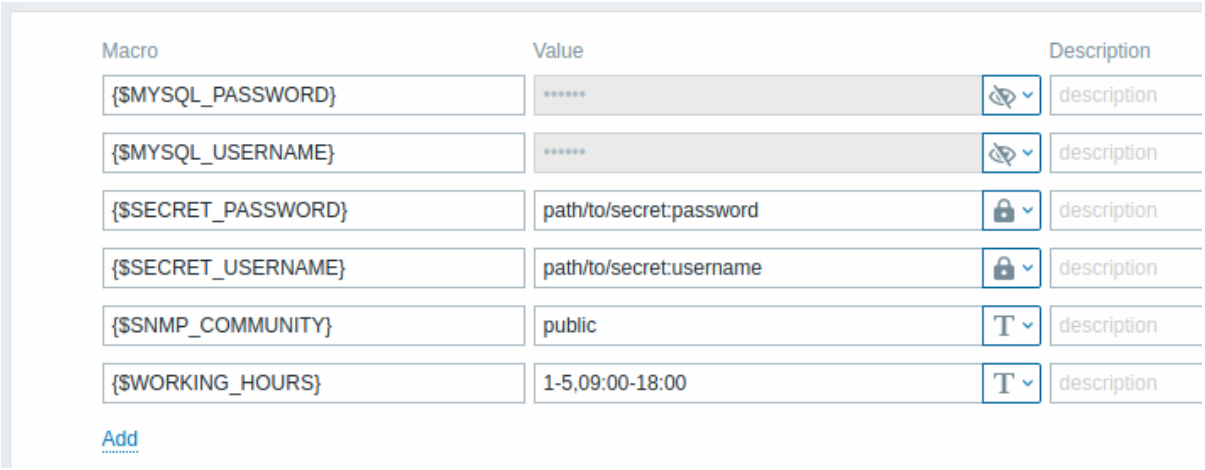
Le lien *Filtre* est disponible au-dessus de la liste des groupes de proxy. Si vous cliquez dessus, un filtre devient disponible, où vous pouvez filtrer les groupes de proxy par nom et par statut.



Macros

Vue d'ensemble

Cette section vous permet de définir des macros utilisateur globales sous forme de paires nom-valeur. Notez que les valeurs des macros peuvent être conservées sous forme de texte brut, de texte secret ou de secret Vault. L'ajout d'une description est également pris en charge.



File d'attente

Vue d'ensemble

Dans la section *Administration* > *File d'attente*, les éléments en attente de mise à jour sont affichés.

Idéalement, lorsque vous ouvrez cette section, tout devrait être « vert », ce qui signifie qu'aucun élément ne se trouve dans la file d'attente. Si tous les éléments sont mis à jour sans délai, aucun élément n'est en attente. Cependant, en raison de performances insuffisantes du serveur, certains éléments peuvent être retardés et les informations correspondantes sont affichées dans cette section. Pour plus de détails, consultez la section File d'attente.

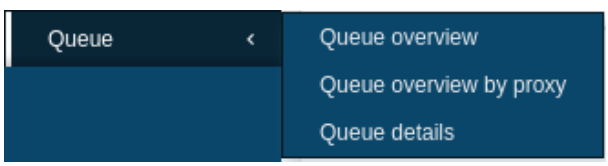
Note:
Le proxy Zabbix n'a pas connaissance des périodes de maintenance ; consultez Calcul des files d'attente pendant la maintenance pour plus de détails.

Note:
La file d'attente est disponible uniquement si le serveur Zabbix est en cours d'exécution. Les éléments ne sont pas comptabilisés dans la file d'attente si l'interface de l'élément devient indisponible en raison de problèmes de connexion ou si l'agent ne fonctionne pas correctement.

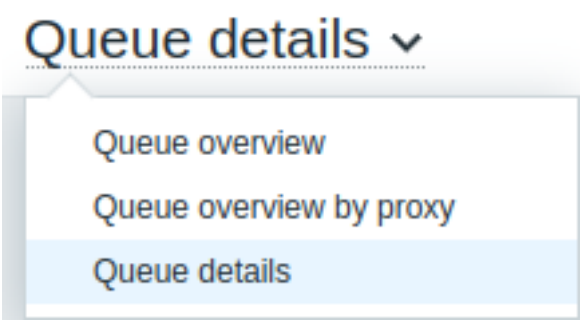
La section *Administration* > *File d'attente* contient les pages suivantes :

- Vue d'ensemble de la file d'attente - affiche la file d'attente par type d'élément ;
- Vue d'ensemble de la file d'attente par proxy - affiche la file d'attente par proxy ;
- Détails de la file d'attente - affiche la liste des éléments retardés.

La liste des pages disponibles s'affiche lorsque vous cliquez sur *File d'attente* dans la section de menu *Administration*. Il est également possible de passer d'une page à l'autre à l'aide de la liste déroulante du titre située dans le coin supérieur gauche.



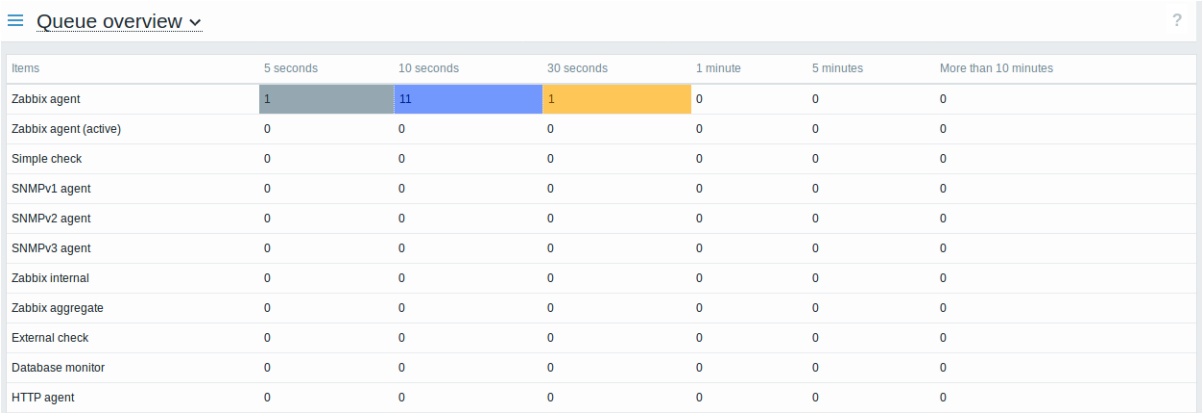
Menu de troisième niveau.



Liste déroulante du titre.

Aperçu par type d'élément

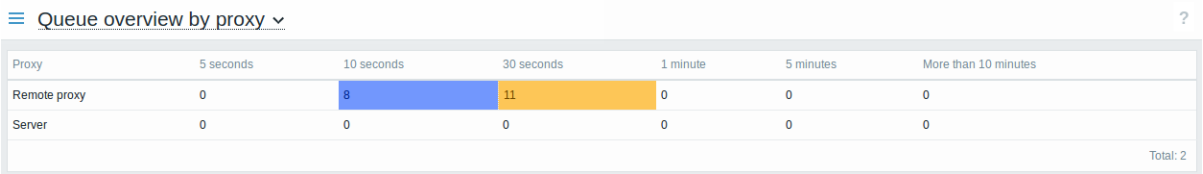
Dans cet écran, il est facile de localiser si le problème est lié à un ou plusieurs types d'éléments.



Chaque ligne contient un type d'élément. Chaque colonne indique le nombre d'éléments en attente - en attente de 5-10 secondes/10-30 secondes/30-60 secondes/1-5 minutes/5-10 minutes ou plus de 10 minutes respectivement.

Aperçu par proxy

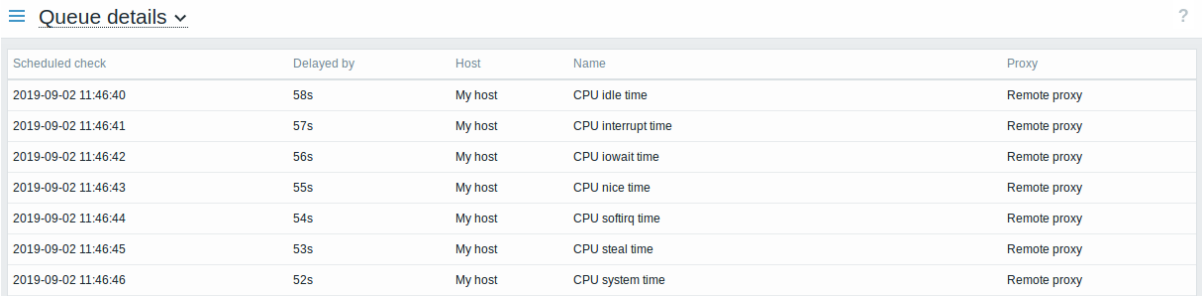
Dans cet écran, il est facile de localiser si le problème est lié à l'un des proxies ou au serveur.



Chaque ligne contient un proxy, avec le serveur en dernier dans la liste. Chaque colonne indique le nombre d'éléments en attente - en attente de 5-10 secondes/10-30 secondes/30-60 secondes/1-5 minutes/5-10 minutes ou plus de 10 minutes respectivement.

Liste des éléments en attente

Dans cet écran, chaque élément en attente est répertorié.



Données affichées :

Colonne	Description
Vérification planifié	L'heure à laquelle la vérification due est affichée.
Retardé de	La durée du retard est affichée.
Hôte	L'hôte de l'élément est affiché.
Nom	Le nom de l'élément en attente est affiché.
Proxy	Le nom du proxy est affiché, si l'hôte est surveillé par proxy.

Messages d'erreur possibles

Il peut arriver qu'aucune donnée ne s'affiche et que le message d'erreur suivant apparaisse :



Dans ce cas, le message d'erreur est le suivant :

```
Cannot display item queue. Permission denied
```

Cela se produit lorsque les paramètres de configuration PHP du fichier de configuration de l'interface - `$ZBX_SERVER` ou `$ZBX_SERVER` et `$ZBX_SERVER_PORT` - pointent vers un serveur Zabbix existant qui utilise une autre base de données.

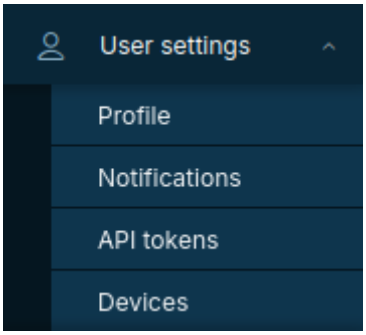
Paramètres utilisateur

Vue d'ensemble

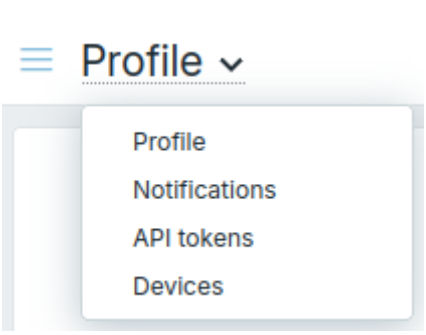
Selon votre rôle utilisateur, la section *Paramètres utilisateur* peut contenir les pages suivantes :

- *Profil* ou *Profil utilisateur* - pour personnaliser certaines fonctionnalités de l'interface Zabbix ;
- *Notifications* - pour personnaliser les notifications de l'utilisateur actuel ;
- *Jetons API* - pour gérer les jetons API attribués à l'utilisateur actuel ;
- *Appareils* - pour gérer les appareils mobiles appartenant à l'utilisateur actuel.

La liste des pages disponibles s'affiche lorsque vous cliquez sur l'icône utilisateur située vers le bas du menu Zabbix (non disponible pour l'utilisateur *guest*). Il est également possible de passer d'une page à l'autre à l'aide de la liste déroulante du titre située dans le coin supérieur gauche.



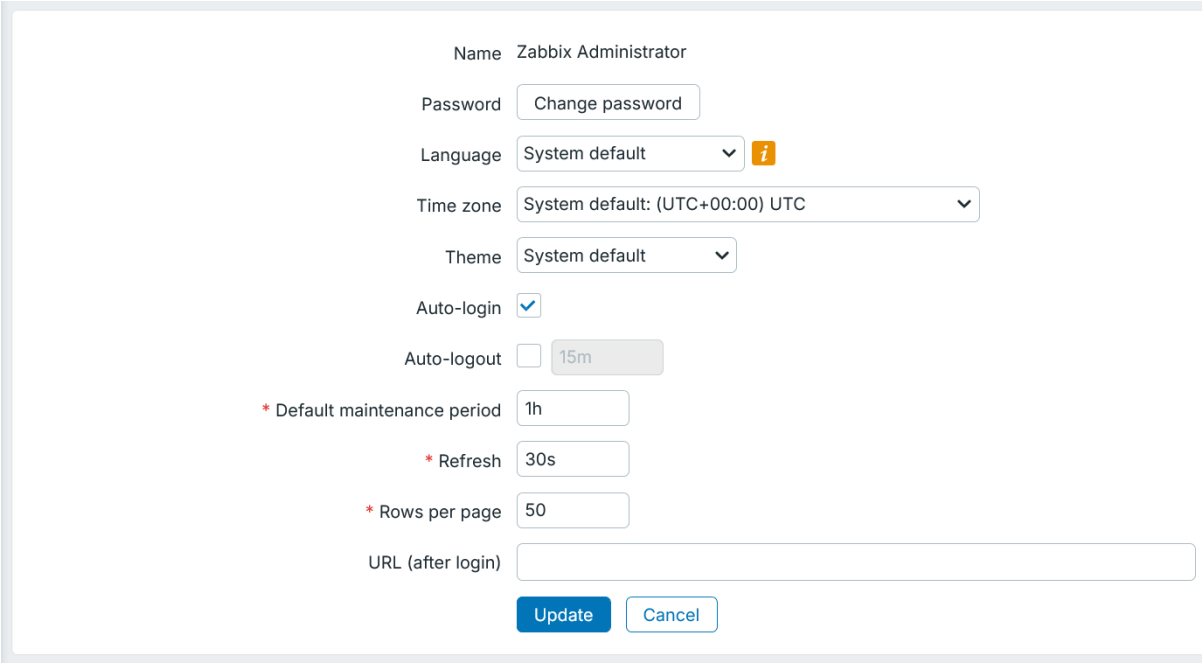
Menu de troisième niveau.



Liste déroulante du titre.

Profil utilisateur

La section *Profil utilisateur* propose des options permettant de définir une langue d'interface personnalisée, un thème de couleurs, le nombre de lignes affichées dans les listes, etc. Les modifications effectuées ici s'appliqueront uniquement à l'utilisateur actuel.



Paramètre	Description
Nom	Le nom et le prénom de l'utilisateur sont affichés. En l'absence de nom et de prénom, le nom d'utilisateur est affiché.

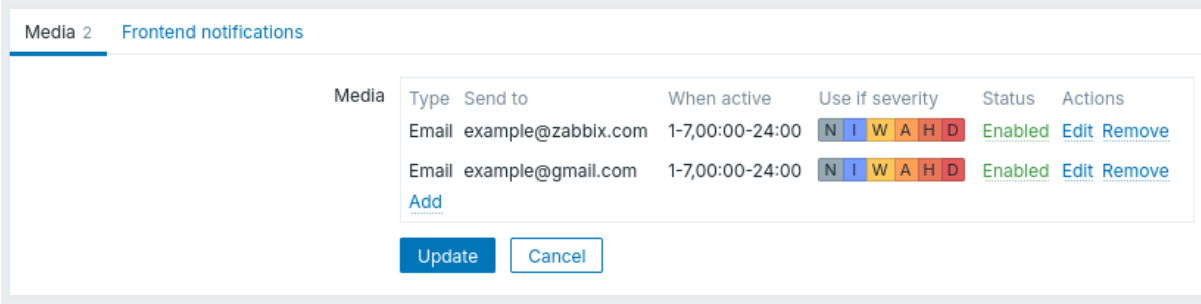
Paramètre	Description
Mot de passe	Cliquez sur le bouton *Modifier le mot de passe* pour ouvrir trois champs : *Ancien mot de passe*, *Nouveau mot de passe*, *Nouveau mot de passe (une nouvelle fois)*. Après une modification réussie du mot de passe, l'utilisateur sera déconnecté de toutes les sessions actives. Notez que le mot de passe peut uniquement être modifié pour les utilisateurs utilisant l'authentification interne de Zabbix.
Langue	Sélectionnez la langue de l'interface de votre choix ou sélectionnez **Valeur par défaut du système** pour utiliser les paramètres système par défaut. La sélection de *Anglais (en_US)* active également le format américain de l'heure et de la date dans l'interface. Pour plus d'informations, consultez Installation de langues supplémentaires pour l'interface.
Fuseau horaire	Sélectionnez le fuseau horaire pour remplacer le fuseau horaire global au niveau de l'utilisateur, ou sélectionnez **Valeur par défaut du système** pour utiliser les paramètres globaux du fuseau horaire.
Thème	Sélectionnez un thème de couleurs spécifique à votre profil : **Valeur par défaut du système** - utiliser les paramètres système par défaut ; **Bleu** - thème bleu standard ; **Bleu (classique)** - thème bleu standard utilisant les anciennes polices ; **Sombre** - thème sombre alternatif ; **Bleu sombre** - thème bleu sombre ; **Sombre (classique)** - thème sombre alternatif utilisant les anciennes polices ; **Clair à contraste élevé** - thème clair à contraste élevé ; **Sombre à contraste élevé** - thème sombre à contraste élevé.
Connexion automatique	Cochez cette case pour que Zabbix se souvienne de vous et vous connecte automatiquement pendant 30 jours. Lors de la connexion avec *Se souvenir pendant 30 jours* activé : - Le délai de déconnexion automatique est réinitialisé (la session persiste jusqu'à la fin de la période de 30 jours). - La connexion automatique est activée lors des visites suivantes pendant 30 jours. Lors de la connexion sans *Se souvenir pendant 30 jours* : - Le paramètre de connexion automatique est effacé. - La déconnexion automatique suit normalement le délai configuré. Des cookies du navigateur sont utilisés à cette fin.
Déconnexion automatique	Lorsque cette case est cochée, vous serez automatiquement déconnecté après le nombre de secondes défini (minimum 90 secondes, maximum 1 jour). Les suffixes de temps sont pris en charge, par exemple : 90s, 5m, 2h, 1d. Notez que cette option ne fonctionnera pas dans les cas suivants : * Lorsque les pages du menu *Surveillance* actualisent les informations en arrière-plan. Si des pages qui actualisent les données à un intervalle précis (tableaux de bord, graphiques, dernières données, etc.) restent ouvertes, la durée de vie de la session est prolongée, ce qui désactive la fonctionnalité de déconnexion automatique. * Si vous vous connectez avec l'option *Se souvenir de moi pendant 30 jours* cochée ; dans ce cas, le délai de déconnexion automatique est remplacé et votre session reste active pendant toute la période. * Lorsque l'authentification est effectuée via un fournisseur d'identité SSO (IdP), la déconnexion automatique met uniquement fin à la session Zabbix. La session SSO avec l'IdP reste active ; l'utilisateur peut donc être en mesure de se reconnecter sans saisir de mot de passe. *Déconnexion automatique* peut également accepter « 0 », ce qui signifie que la fonctionnalité de déconnexion automatique est désactivée après la mise à jour des paramètres du profil.
Période de maintenance par défaut	Définissez la période de maintenance par défaut pour une maintenance ponctuelle.
Actualisation	Définissez la fréquence d'actualisation des informations sur les pages du menu *Surveillance* (minimum 0 seconde, maximum 1 heure). Les suffixes de temps sont pris en charge, par exemple : 30s, 90s, 1m, 1h.
Lignes par page	Définissez le nombre de lignes affichées par page dans les listes. Un nombre inférieur de lignes (et donc moins d'enregistrements à afficher) permet un chargement plus rapide.

Paramètre	Description
URL (après la connexion)	Définissez une URL spécifique à afficher après la connexion. Au lieu des *Tableaux de bord* par défaut, il peut s'agir, par exemple, de l'URL de *Surveillance* > *Déclencheurs*. Ce champ est désactivé si l'URL après la connexion est limitée à une page spécifique pour votre rôle utilisateur par un utilisateur *Super administrateur*. Si le champ est désactivé ou si l'utilisateur ne spécifie aucune valeur, la valeur par défaut est affichée sous le champ, par exemple `zabbix.php?action=dashboard.view`

Notifications

La section *Notifications* fournit des options permettant de personnaliser les notifications de l'utilisateur actuel.

L'onglet *Media* vous permet de spécifier les détails des médias pour l'utilisateur, tels que les types de médias et les adresses à utiliser, ainsi que le moment où les utiliser pour envoyer des notifications.



Si le type de média a été désactivé :

- une icône d'information jaune s'affiche après le nom ;
- « Disabled » s'affiche dans la colonne *Status*.

Note:
Les autorisations d'un utilisateur pour modifier ses propres détails de médias peuvent être accordées/révoquées en fonction de son rôle utilisateur (voir l'option *Create and edit own media*). Les autorisations d'un utilisateur super administrateur pour modifier les détails de médias d'autres utilisateurs peuvent également être accordées/révoquées en fonction de son rôle utilisateur (voir l'option *Create and edit user media*)

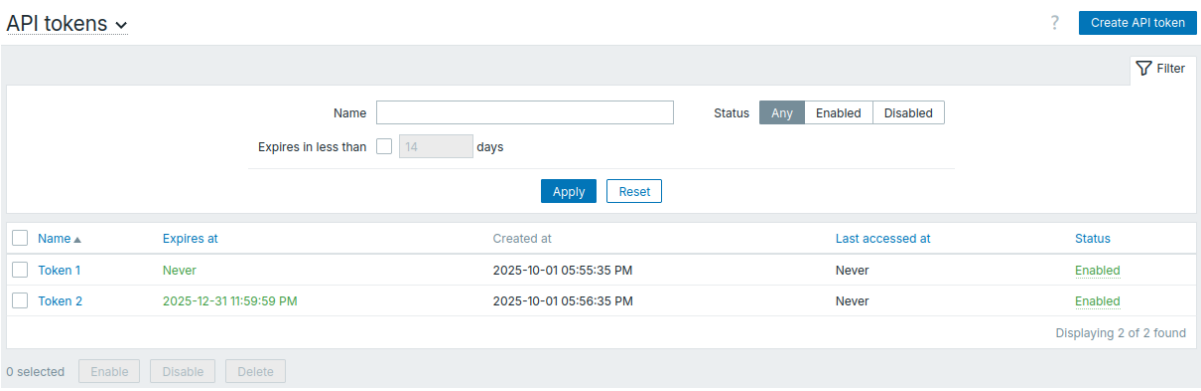
Notez que pour les utilisateurs provisionnés :

- les médias d'un utilisateur provisionné ne peuvent pas être supprimés ;
- les médias d'un utilisateur provisionné peuvent être désactivés/activés ;
- les champs de médias d'un utilisateur provisionné tels que *When active*, *Use if severity* et *Enabled* peuvent être modifiés manuellement ;
- des médias utilisateur supplémentaires peuvent être ajoutés manuellement pour les utilisateurs provisionnés (par exemple, des adresses e-mail supplémentaires) ;
- les médias utilisateur ajoutés manuellement peuvent être supprimés.

L'onglet *Frontend notifications* vous permet de définir les notifications globales.

Jetons API

La section *Jetons API* vous permet de consulter les jetons attribués à l'utilisateur, de modifier les détails d'un jeton et de créer de nouveaux jetons. Cette section n'est disponible pour un utilisateur que si l'action *Gérer les jetons API* est autorisée dans les paramètres du rôle utilisateur.



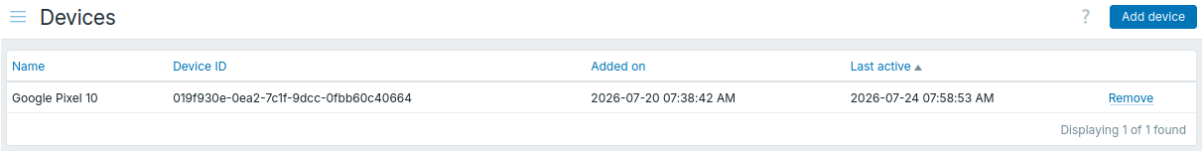
Vous pouvez filtrer les jetons API par nom, date d'expiration ou état (*Activé/Désactivé*). Cliquez sur l'état du jeton dans la liste pour activer/désactiver rapidement un jeton. Vous pouvez également activer/désactiver plusieurs jetons à la fois en les sélectionnant dans la liste, puis en cliquant sur les boutons *Activer/Désactiver* sous la liste.

Attention:
Les utilisateurs ne peuvent pas consulter la valeur du *jeton d'authentification* des jetons qui leur sont attribués dans Zabbix. La valeur du *jeton d'authentification* n'est affichée qu'une seule fois, immédiatement après la création d'un jeton. Si le jeton a été perdu, il doit être régénéré.

Appareils

La section *Appareils* vous permet d'ajouter ou de supprimer vos appareils mobiles à utiliser avec l'application mobile Zabbix.

Cette section s'affiche si `EnableMobileDevices` est activé.



Pour ajouter un appareil :

1. Cliquez sur *Ajouter un appareil*.
2. Scannez le code QR dans votre application mobile Zabbix.

Note:
Lorsque vous ajoutez un appareil, Zabbix ajoute également le type de média notification push à vos médias utilisateur. Lorsque vous supprimez l'appareil, ce média utilisateur est également supprimé. Si plusieurs types de médias de notification push sont configurés, Zabbix ajoute celui qui est activé et qui apparaît en premier dans l'ordre alphabétique.

Les appareils ajoutés apparaissent dans une liste. Celle-ci affiche leur nom, leur ID, ainsi que les dates et heures auxquelles ils ont été ajoutés et ont été actifs pour la dernière fois.

Vous pouvez également supprimer des appareils. Lorsque vous supprimez un appareil, ses identifiants d'accès sont révoqués et supprimés de Zabbix. Il ne peut alors plus accéder à Zabbix.

Les super administrateurs peuvent également gérer les appareils des autres utilisateurs dans *Utilisateurs* > *Appareils*.

Notifications globales

Vue d'ensemble

Les notifications globales permettent d'afficher les problèmes en temps réel directement sur votre écran actuel dans l'interface web de Zabbix.

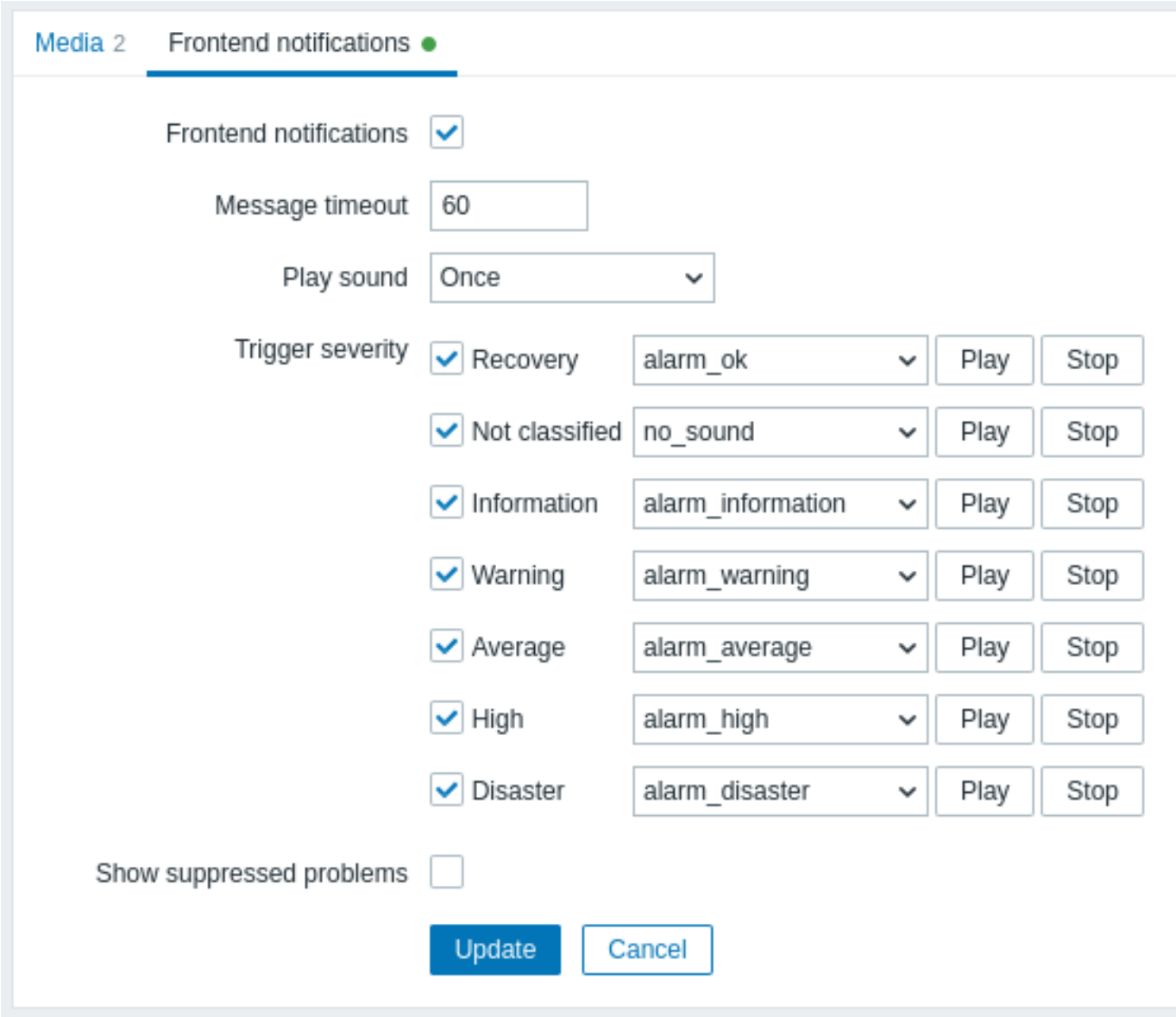
Sans notifications globales, lorsque vous travaillez en dehors des sections *Problèmes* ou *Tableau de bord*, vous ne recevriez aucune information sur les problèmes en cours. Les notifications globales garantissent que ces informations sont affichées, quel que soit l'endroit où vous vous trouvez dans l'interface web de Zabbix.

Les notifications globales comprennent à la fois l'affichage d'un message et la lecture d'un son.

Attention:
La lecture automatique des sons peut être désactivée (par défaut) dans les versions récentes des navigateurs. Dans ce cas, vous devez activer ce paramètre manuellement.

Configuration

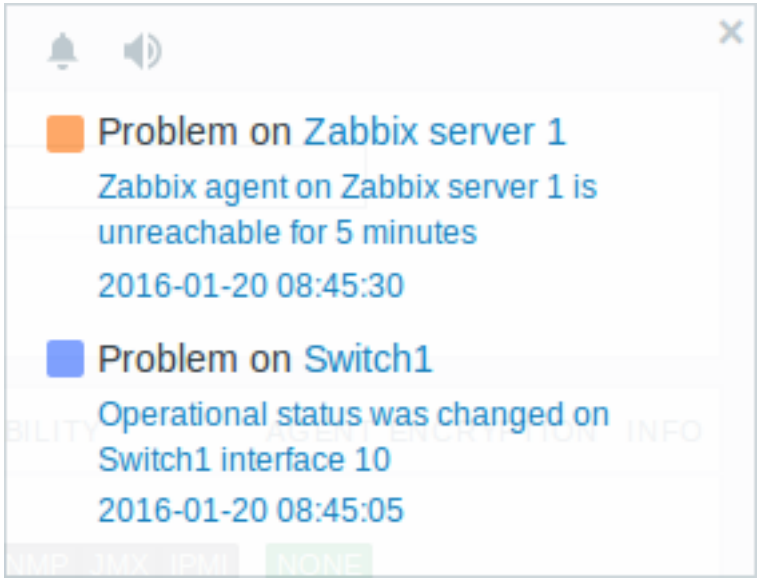
Les notifications globales peuvent être activées par utilisateur dans l'onglet **Frontend notifications** de la section Notifications.



Paramètre	Description
Frontend notifications	Cochez la case pour activer les notifications globales.
Message timeout	Définissez la durée pendant laquelle le message sera affiché. Par défaut, les messages restent à l'écran pendant 60 secondes. Les suffixes de temps sont pris en charge, par exemple : 30s, 5m, 2h, 1d.
Play sound	Définissez la durée pendant laquelle le son sera joué. **Once** - le son est joué une seule fois, en entier ; **10 seconds** - le son est répété pendant 10 secondes ; **Message timeout** - le son est répété tant que le message est visible.
Trigger severity	Définissez les sévérités de déclencheur pour lesquelles les notifications globales et les sons seront activés. Vous pouvez également sélectionner des sons adaptés aux différentes sévérités. Si aucune sévérité n'est cochée, aucun message ne sera affiché. De plus, les messages de rétablissement ne seront affichés que pour les sévérités cochées. Par exemple, si *Recovery* et *Disaster* sont cochés, les notifications globales seront affichées pour les problèmes et les rétablissements des déclencheurs de sévérité *Disaster*.
Show suppressed problems	Cochez la case pour afficher les notifications des problèmes qui seraient autrement supprimés (non affichés) en raison de la maintenance de l'hôte.

Messages globaux affichés

À mesure que les messages arrivent, ils s'affichent dans une section flottante sur le côté droit. Vous pouvez repositionner librement cette section en faisant glisser son en-tête.



Pour cette section, plusieurs commandes sont disponibles :

- le bouton **Snooze** coupe temporairement le son d'alarme actuellement actif ;
- le bouton **Mute/Unmute** permet de basculer entre la lecture et l'absence totale de lecture des sons d'alarme.

Son dans les navigateurs

Aperçu

Le son est utilisé dans les notifications globales.

Pour que les sons soient lus dans le frontend Zabbix, les *Notifications du frontend* doivent être activées dans l'onglet *Notifications du frontend* du profil utilisateur, avec toutes les sévérités de déclencheur cochées. De plus, les sons doivent être activés dans la fenêtre contextuelle des notifications globales.

Si, pour une raison quelconque, l'audio ne peut pas être lu sur l'appareil, le bouton dans la fenêtre contextuelle des notifications globales restera en permanence à l'état « muet », accompagné du message « Cannot support notification audio for this device » lors du survol du bouton .

Les sons, y compris les extraits audio par défaut, sont pris en charge uniquement au format MP3.

Les sons du frontend Zabbix ont été testés avec succès dans les versions récentes des navigateurs Firefox et Opera sous Linux, ainsi que dans les navigateurs Chrome, Firefox, Microsoft Edge et Opera sous Windows.

Attention:
La lecture automatique des sons peut être désactivée (par défaut) dans les versions récentes des navigateurs. Dans ce cas, vous devez activer ce paramètre manuellement.

Recherche globale

Il est possible d'effectuer une recherche dans l'interface Zabbix pour les hôtes, les groupes d'hôtes, les modèles et les groupes de modèles.

Le champ de saisie de recherche se trouve sous le logo Zabbix dans le menu. La recherche peut être lancée en appuyant sur *Entrée* ou en cliquant sur l'icône de recherche .



S'il existe un hôte dont le nom contient la chaîne saisie dans n'importe quelle partie du nom, une liste déroulante apparaîtra, affichant tous ces hôtes (avec la partie correspondante surlignée en orange). La liste déroulante affichera également un hôte si le nom visible de cet hôte correspond au nom technique saisi comme chaîne de recherche ; l'hôte correspondant sera affiché, mais sans surlignage.

Attributs recherchables

Les hôtes peuvent être recherchés selon les propriétés suivantes :

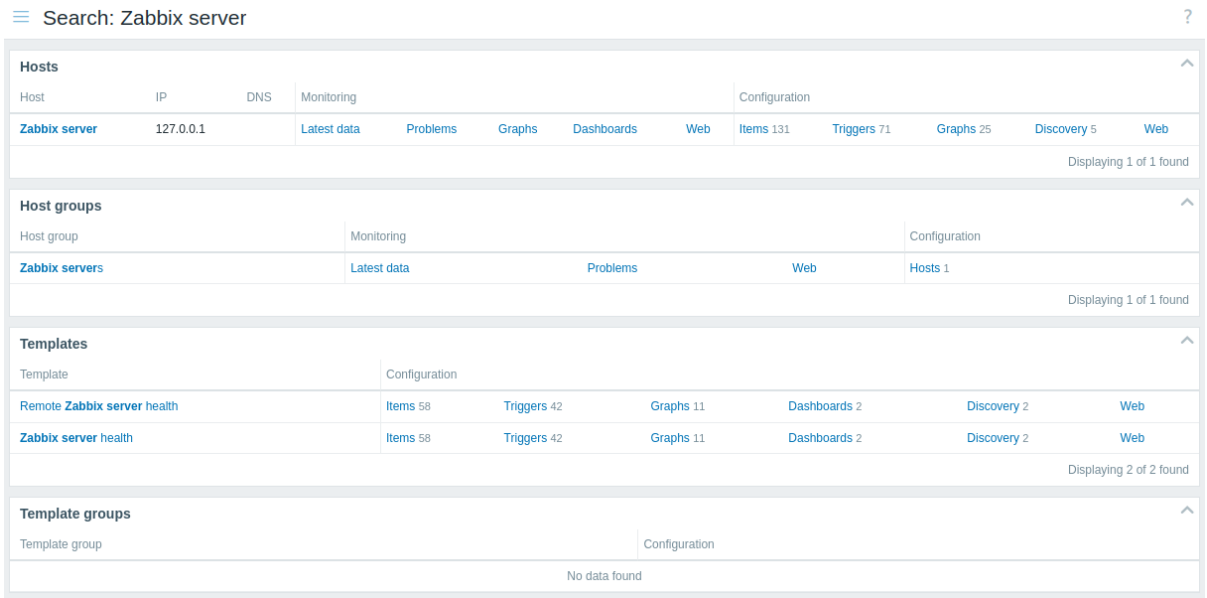
- Nom de l'hôte
- Nom visible
- Adresse IP
- Nom DNS

Les modèles peuvent être recherchés par nom ou par nom visible. Si vous effectuez une recherche par un nom différent du nom visible (d'un modèle/hôte), dans les résultats de recherche, il est affiché sous le nom visible, entre parenthèses.

Les groupes d'hôtes et de modèles peuvent être recherchés par nom. La spécification d'un groupe parent sélectionne implicitement tous les groupes imbriqués.

Résultats de recherche

Les résultats de recherche se composent de quatre blocs distincts pour les hôtes, les groupes d'hôtes, les modèles et les groupes de modèles.



Il est possible de réduire/développer chaque bloc individuellement. Le nombre d'entrées est affiché en bas de chaque bloc, par exemple, *Affichage de 13 sur 13 trouvés*. S'il n'y a aucune entrée, le nombre d'entrées n'est pas affiché. Le nombre total d'entrées affichées dans un bloc est limité à 100.

Chaque entrée fournit des liens vers les données de supervision et de configuration. Consultez la liste complète des liens.

Pour toutes les données de configuration (telles que les éléments, les déclencheurs, les graphiques), le nombre d'entités trouvées est affiché par un nombre à côté du nom de l'entité, en gris. **Notez** que s'il n'y a aucune entité, aucun nombre n'est affiché.

Les hôtes activés sont affichés en bleu, les hôtes désactivés en rouge.

Liens disponibles

Pour chaque entrée, les liens suivants sont disponibles :

- Hôtes
  - Surveillance
    - Dernières données
    - Problèmes
    - Graphiques
    - Tableaux de bord d'hôte
    - Scénarios web
  - Configuration
    - Éléments
    - Déclencheurs
    - Graphiques
    - Règles de découverte
    - Scénarios web
- Groupes d'hôtes
  - Surveillance
    - Dernières données
    - Problèmes
    - Scénarios web
  - Configuration
    - Hôtes
- Modèles
  - Configuration
    - Éléments
    - Déclencheurs
    - Graphiques
    - Tableaux de bord de modèle
    - Règles de découverte
    - Scénarios web
- Groupes de modèles
  - Configuration
    - Modèles

Mode de maintenance de l'interface

Aperçu

Il est possible de désactiver temporairement l'interface web de Zabbix afin de restreindre l'accès. Cela est utile pour protéger la base de données Zabbix contre les modifications initiées par les utilisateurs, et ainsi préserver son intégrité.

Lorsque l'interface web de Zabbix est en mode maintenance, vous pouvez arrêter la base de données en toute sécurité et effectuer des tâches de maintenance.

Les utilisateurs provenant d'adresses IP définies pourront interagir normalement avec l'interface web pendant le mode maintenance.

Configuration

Pour activer le mode maintenance, ouvrez le fichier `maintenance.inc.php` (situé dans `/conf` du répertoire de documents HTML de Zabbix sur le serveur web) et décommentez les lignes suivantes :

```
// Mode maintenance.
define('ZBX_DENY_GUI_ACCESS', 1);

// Tableau d'adresses IP autorisées à se connecter à l'interface (facultatif).
$ZBX_GUI_ACCESS_IP_RANGE = array('127.0.0.1');

// Message affiché sur l'écran d'avertissement (facultatif).
```

```
$ZBX_GUI_ACCESS_MESSAGE = 'Nous mettons à niveau la base de données MySQL jusqu'à 15:00. Restez à l'écoute
```

Note:
Dans la plupart des cas, le fichier `maintenance.inc.php` se trouve dans `/conf` du répertoire des documents HTML de Zabbix sur le serveur web. Cependant, certains systèmes d'exploitation et serveurs web peuvent utiliser un emplacement différent.
Par exemple, l'emplacement pour :

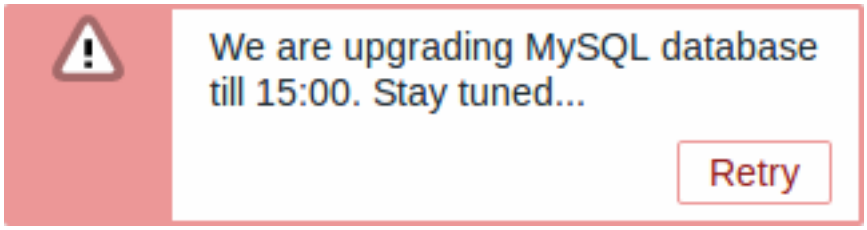
- SUSE et RedHat est `/etc/zabbix/web/maintenance.inc.php`.
- Les systèmes basés sur Debian est `/usr/share/zabbix/conf/`.

Voir également Copie des fichiers PHP.

Parameter	Details
ZBX_DENY_GUI_ACCESS	Si ce paramètre est défini avec une valeur quelconque, le mode maintenance sera activé. Pour désactiver le mode maintenance, commentez-le ou supprimez-le.
ZBX_GUI_ACCESS_IP_RANGE	Tableau d'adresses IP autorisées à se connecter à l'interface (facultatif). Par exemple : `array('192.168.1.1', '192.168.1.2')`
ZBX_GUI_ACCESS_MESSAGE	Message destiné à informer les utilisateurs de la maintenance (facultatif). S'il n'est pas défini, le message par défaut « *Zabbix est en maintenance* » sera utilisé.

Affichage

Les utilisateurs verront l'écran suivant lorsqu'ils tenteront d'accéder au frontend de Zabbix pendant le mode de maintenance. L'écran est actualisé toutes les 30 secondes afin de revenir à un état normal sans intervention de l'utilisateur lorsque la maintenance est terminée.



Les adresses IP définies dans *ZBX_GUI_ACCESS_IP_RANGE* pourront accéder au frontend comme d'habitude.

Paramètres de la page

Vue d'ensemble

La plupart des pages de l'interface web de Zabbix prennent en charge divers paramètres HTTP GET qui contrôlent ce qui sera affiché. Ils peuvent être transmis en spécifiant des paires paramètre=valeur après l'URL, séparées de l'URL par un point d'interrogation (?) et les unes des autres par des esperluettes (&).

Surveillance > Problèmes

Les paramètres suivants sont pris en charge :

Parameter	Description	Example
show	Option de filtre *Afficher*. Valeurs possibles : 1 - problèmes récents; 2 - tous; 3 - en état de problème.	show=1
name	Option de filtre *Problème* : chaîne libre.	name=Zabbix agent

Parameter	Description	Example
severities	Option de filtre *Sévérité* : tableau des sévérités sélectionnées au format `severities[*]=*` (remplacez `*` par le niveau de sévérité). Valeurs possibles : 0 - non classé; 1 - information; 2 - avertissement; 3 - moyen; 4 - élevé; 5 - catastrophe.	severities[3]=3
inventory	Option de filtre *Inventaire de l'hôte* : tableau des champs d'inventaire `[field]`, `[value]`	inventory[0][field]=type&invento
evaltype	Option de filtre *Tags* : méthode d'évaluation des tags [(/manual/web_interface/frontend_sections/monitoring/problems#using-filter)]. Valeurs possibles : 0 - et/ou; 2 - ou.	evaltype=0
tags	Option de filtre *Tags* : tableau des tags définis `[tag]`, `[operator]`, `[value]` Valeurs possibles de operator : 0 - contient; 1 - égal à; 2 - ne contient pas; 3 - n'est pas égal à; 4 - existe; 5 - n'existe pas.	tags[0][tag]=target&tags[0][oper
show_tags	Option de filtre *Afficher les tags*. Valeurs possibles : 0 - aucun; 1 - un; 2 - deux; 3 - trois.	show_tags=3
tag_name_format	Option de filtre *Nom du tag*. Valeurs possibles : 0 - nom complet; 1 - nom abrégé; 2 - aucun.	tag_name_format=1
tag_priority	Option de filtre *Priorité d'affichage des tags* : chaîne séparée par des virgules de priorités d'affichage des tags	tag_priority=customer, target
show_suppressed	Option de filtre *Afficher les problèmes supprimés*. Valeurs possibles : 1 - afficher; 2 - ne pas afficher.	show_suppressed=1
acknowledgement_status	Option de filtre *État de prise en compte*. Valeurs possibles : 0 - tous; 1 - non pris en compte; 2 - pris en compte.	acknowledgement_status=0
acknowledged_by_me	Option de filtre *Par moi* ; pris en charge uniquement avec `acknowledgement_status=2`. Valeurs possibles : 0 - désactivé; 1 - activé.	acknowledged_by_me=1

Parameter	Description	Example
compact_view	Option de filtre *Vue compacte*. Valeurs possibles : 0 - désactivé; 1 - activé.	compact_view=1
highlight_row	Option de filtre *Mettre en surbrillance toute la ligne*. Valeurs possibles : 0 - désactivé; 1 - activé.	highlight_row=1
filter_name	Option des propriétés du filtre *Nom* : chaîne libre	filter_name=Databases
filter_show_counter	Option des propriétés du filtre *Afficher le nombre d'enregistrements*. Valeurs possibles : 0 - désactivé; 1 - activé.	filter_show_counter=1
filter_custom_time	Option des propriétés du filtre *Définir une période personnalisée*. Valeurs possibles : 0 - désactivé; 1 - activé.	filter_custom_time=1
sort	Colonne de tri. Valeurs possibles : clock - trier par la colonne *Heure*; host - trier par la colonne *Hôte*; severity - trier par la colonne *Sévérité*; name - trier par la colonne *Problème*.	sort=clock
sortorder	Ordre de tri des résultats. Valeurs possibles : DESC - décroissant; ASC - croissant.	sortorder=DESC
age_state	Option de filtre *Âge inférieur à* ; pris en charge uniquement avec `show`=3. Valeurs possibles : 0 - désactiver le paramètre `age`; 1 - activer le paramètre `age`.	age_state=1
age	Option de filtre *Âge inférieur à* : entier, nombre de jours ; pris en charge uniquement avec `age_state`=1 et `show`=3.	age=7
groupids	Option de filtre *Groupes d'hôtes* : tableau des ID de groupes d'hôtes.	groupids[]=4
hostids	Option de filtre *Hôtes* : tableau des ID d'hôtes	hostids[]=10084
triggerids	Option de filtre *Déclencheurs* : tableau des ID de déclencheurs	triggerids[]=22382
show_timeline	Option de filtre *Afficher la chronologie* ; non pris en charge avec `compact_view`=1. Valeurs possibles : 0 - ne pas afficher; 1 - afficher.	show_timeline=1
details	Option de filtre *Afficher les détails*. Valeurs possibles : 0 - ne pas afficher; 1 - afficher.	details=1
from	Début de la plage de dates, peut être relatif (par exemple, now-1m) ; pris en charge uniquement avec `filter_custom_time`=1	from=now-2h
to	Fin de la plage de dates, peut être relative (par exemple, now-1m) ; pris en charge uniquement avec `filter_custom_time`=1	to=now

Voir aussi : page Problèmes, options de filtre.

Mode kiosque

Le mode kiosque des pages d'interface prises en charge peut être activé à l'aide de paramètres d'URL. Par exemple, dans les tableaux de bord :

- `/zabbix.php?action=dashboard.view&kiosk=1` - activer le mode kiosque
- `/zabbix.php?action=dashboard.view&kiosk=0` - activer le mode normal

Diaporama

Il est possible d'activer un diaporama dans le tableau de bord :

- `/zabbix.php?action=dashboard.view&slideshow=on` - activer le diaporama

Définitions

Vue d'ensemble

Bien que de nombreux éléments de l'interface web puissent être configurés à l'aide de l'interface web elle-même, certaines personnalisations ne sont actuellement possibles qu'en modifiant un fichier de définitions.

Ce fichier est `defines.inc.php`, situé dans /include du répertoire des documents HTML de Zabbix.

Paramètres

Paramètres de ce fichier susceptibles d'intéresser les utilisateurs :

- ZBX_MIN_PERIOD

Période minimale du graphique, en secondes. Une minute par défaut.

- GRAPH_YAXIS_SIDE_DEFAULT

Emplacement par défaut de l'axe Y dans les graphiques simples et valeur par défaut de la liste déroulante lors de l'ajout d'éléments aux graphiques personnalisés. Valeurs possibles : 0 - gauche, 1 - droite.

Par défaut : 0

- ZBX_SESSION_NAME

Chaîne utilisée comme nom du cookie de session de l'interface Zabbix.

Par défaut : zbx_sessionid

- ZBX_DATA_CACHE_TTL

Délai d'expiration TTL, en secondes, utilisé pour invalider le cache de données de la réponse Vault. Définissez 0 pour désactiver la mise en cache des réponses Vault.

Par défaut : 60

- SUBFILTER_VALUES_PER_GROUP

Nombre de valeurs de sous-filtre par groupe (par exemple, dans le sous-filtre des dernières données).

Par défaut : 1000

- ZBX_MAX_WIDGET_LINES

Nombre maximal de lignes de widget à afficher.

Par défaut : 1000

Langues supplémentaires de l'interface

Vue d'ensemble Pour utiliser une autre langue que l'anglais dans l'interface web Zabbix, sa locale doit être installée sur le serveur web. De plus, l'extension PHP gettext est requise pour que les traductions fonctionnent.

Installation des locales Pour lister toutes les langues installées, exécutez :

```
locale -a
```

Si certaines langues nécessaires ne sont pas répertoriées, ouvrez le fichier `/etc/locale.gen` et décommentez les locales requises. Comme Zabbix utilise l'encodage UTF-8, vous devez sélectionner des locales avec un jeu de caractères UTF-8.

Exécutez maintenant :

```
locale-gen
```

Redémarrez le serveur web. Notez que si PHP-FPM est installé et utilisé par le serveur web, il doit également être redémarré en même temps que le serveur web.

Les locales devraient maintenant être installées. Il peut être nécessaire de recharger la page de l'interface Zabbix dans le navigateur à l'aide de `Ctrl + F5` pour que les nouvelles langues apparaissent.

Installation de Zabbix Si vous installez Zabbix directement depuis le dépôt git Zabbix, les fichiers de traduction doivent être générés manuellement. Pour générer les fichiers de traduction, exécutez :

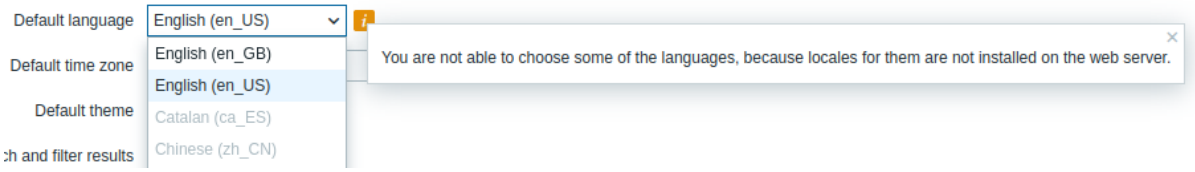
```
make gettext
locale/make_mo.sh
```

Cette étape n'est pas nécessaire lors de l'installation de Zabbix à partir de paquets ou de fichiers source `tar.gz`.

Sélection d'une langue Il existe plusieurs façons de sélectionner une langue dans l'interface web Zabbix :

- Lors de l'installation de l'interface web - dans l'assistant d'installation du frontend installation wizard. La langue sélectionnée sera définie comme langue système par défaut.
- Après l'installation, la langue système par défaut peut être modifiée dans *Administration* > *General* > *GUI*.
- La langue d'un utilisateur particulier peut être modifiée dans le profil utilisateur.

Si un paramètre régional pour une langue n'est pas installé sur la machine, cette langue apparaîtra en grisé dans le sélecteur de langue de Zabbix. Une icône orange s'affiche à côté du sélecteur de langue si au moins un paramètre régional est manquant. En cliquant sur cette icône, le message suivant s'affichera : "Vous ne pouvez pas choisir certaines langues, car les paramètres régionaux correspondants ne sont pas installés sur le serveur web."



Créer votre propre thème

Vue d'ensemble

Par défaut, Zabbix fournit un certain nombre de thèmes prédéfinis pour l'interface. Pour créer un thème personnalisé, suivez la procédure étape par étape fournie ici.

Étape 1: Créez votre thème

Pour créer votre propre thème :

1. Créez un fichier CSS à partir de zéro ou copiez un thème existant (par exemple `blue-theme.css`, `dark-theme.css` ou un autre) depuis le répertoire `assets/styles/` et modifiez-le.
2. Enregistrez votre thème (par exemple, `custom-theme.css`) dans le répertoire `assets/styles/`.

Étape 2 : Activer votre thème dans la liste des thèmes

Pour rendre votre thème personnalisé disponible dans l'interface, vous devez l'ajouter à la liste des thèmes renvoyée par la méthode `APP::getThemes()`. Pour ce faire, remplacez la méthode `ZBase::getThemes()` dans la classe `APP` comme suit :

1. Ouvrez le fichier `include/classes/core/APP.php` et localisez la classe `APP` :

```
class APP extends ZBase {

}
```

2. Dans la classe, avant son accolade fermante finale `}`, ajoutez la méthode suivante :

```
public static function getThemes() {
          return array_merge(parent::getThemes(), [
              'custom-theme' => _('Custom theme')
          ]);
}
```

où :

- `custom-theme` - l'identifiant interne de votre thème, qui doit correspondre au nom de votre fichier CSS sans l'extension `.css` ;
- `Custom theme` - le nom d'affichage que vous verrez dans l'interface.

Pour ajouter plusieurs thèmes, répertoriez-les dans le tableau en séparant chaque entrée par une virgule. La virgule finale après la dernière entrée est facultative.

Par exemple :

```
      public static function getThemes() {
          return array_merge(parent::getThemes(), [
              'custom-theme' => _('Custom theme'),
              'anothertheme' => _('Another theme'),
              'onemoretheme' => _('One more theme')
          ]);
      }
```

Étape 3 : Modifier les couleurs des graphiques

Pour modifier les couleurs des graphiques, un enregistrement correspondant au nouveau thème doit être ajouté à la table de base de données `graph_theme`.

Exemple d'ajout d'un enregistrement dans une base de données MySQL/MariaDB :

```
    mysql -u zabbix -p
    # Enter password:
```

```
mysql> USE zabbix;
mysql> INSERT INTO graph_theme (
    graphthemeid,
    theme,
    backgroundcolor,
    graphcolor,
    gridcolor,
    maingridcolor,
    gridbordercolor,
    textcolor,
    highlightcolor,
    leftpercentilecolor,
    rightpercentilecolor,
    nonworktimecolor,
    colorpalette
)
SELECT
    MAX(graphthemeid) + 1,
    'custom-theme',
    'FFFFFF',
    'FFFFFF',
    'CCD5D9',
    'ACBBC2',
    'ACBBC2',
    '1F2C33',
    'CC745E',
    '5ECCAB',
    'CC745E',
    'EBEBEB',
    '329F7E,C2583D,346D91,B26E44,CC6C91,7A6DC2,C4AA56,7C2731,BF75B8,73A350,B04833,633A60,879CCC,7FAD6C,324
FROM graph_theme;
```

Étape 4 : Utiliser le nouveau thème

Dans l'interface Zabbix, vous pouvez définir ce thème comme thème par défaut ou le sélectionner dans un profil utilisateur.

Mode de débogage

Aperçu

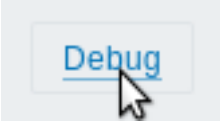
Le mode Debug peut être utilisé pour diagnostiquer les problèmes de performance de l'interface web.

Configuration

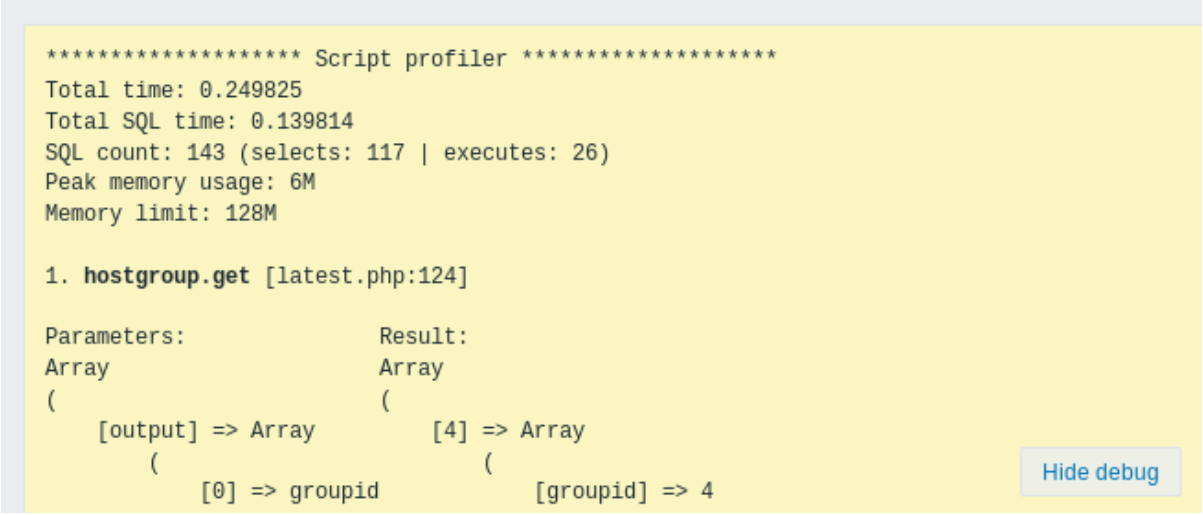
Le mode de débogage peut être activé pour des utilisateurs individuels appartenant à un groupe d'utilisateurs :

- lors de la configuration d'un groupe d'utilisateurs ;
- lors de l'affichage des groupes d'utilisateurs configurés.

Lorsque le *mode de débogage* est activé pour un groupe d'utilisateurs, ses utilisateurs verront un bouton *Debug* dans le coin inférieur droit de la fenêtre du navigateur :



En cliquant sur le bouton *Debug*, une nouvelle fenêtre s'ouvre sous le contenu de la page et contient les statistiques SQL de la page, ainsi qu'une liste des appels API et des instructions SQL individuelles :



En cas de problèmes de performance avec la page, cette fenêtre peut être utilisée pour rechercher la cause racine du problème.

Warning:
L'activation du *mode de débogage* a un impact négatif sur les performances de l'interface.

Cookies utilisés par Zabbix

Aperçu

Cette page fournit une liste des cookies utilisés par Zabbix.

Nom	Description	Valeurs	Expiration/Max-Age	HttpOnly[1]	Secure[2]
ZBX_SESSION_NAME	Données de session du frontend Zabbix, stockées au format JSON encodé en base64		Session (expire à la fin de la session de navigation)	+	+ (uniquement si HTTPS est activé sur un serveur web)
tab	Numéro de l'onglet actif ; ce cookie est utilisé uniquement sur les pages comportant plusieurs onglets (par ex. page de configuration *Host*, *Trigger* ou *Action*) et est créé lorsqu'un utilisateur navigue d'un onglet principal vers un autre onglet (tel que l'onglet *Tags* ou *Dependencies*). 0 est utilisé pour l'onglet principal.	Exemple : 1	Session (expire à la fin de la session de navigation)	-	-
browserwarning_ignore	Indique si un avertissement concernant l'utilisation d'un navigateur obsolète doit être ignoré.	yes	Session (expire à la fin de la session de navigation)	-	-
system-message-ok	Message à afficher dès que la page est rechargée.	Message en texte brut	Session (expire à la fin de la session de navigation) ou dès que la page est rechargée	+	-
system-message-error	Message d'erreur à afficher dès que la page est rechargée.	Message en texte brut	Session (expire à la fin de la session de navigation) ou dès que la page est rechargée	+	-

Note:
Le forçage de l'indicateur 'HttpOnly' sur les cookies Zabbix par une directive du serveur web n'est pas pris en charge.

Notes de bas de page

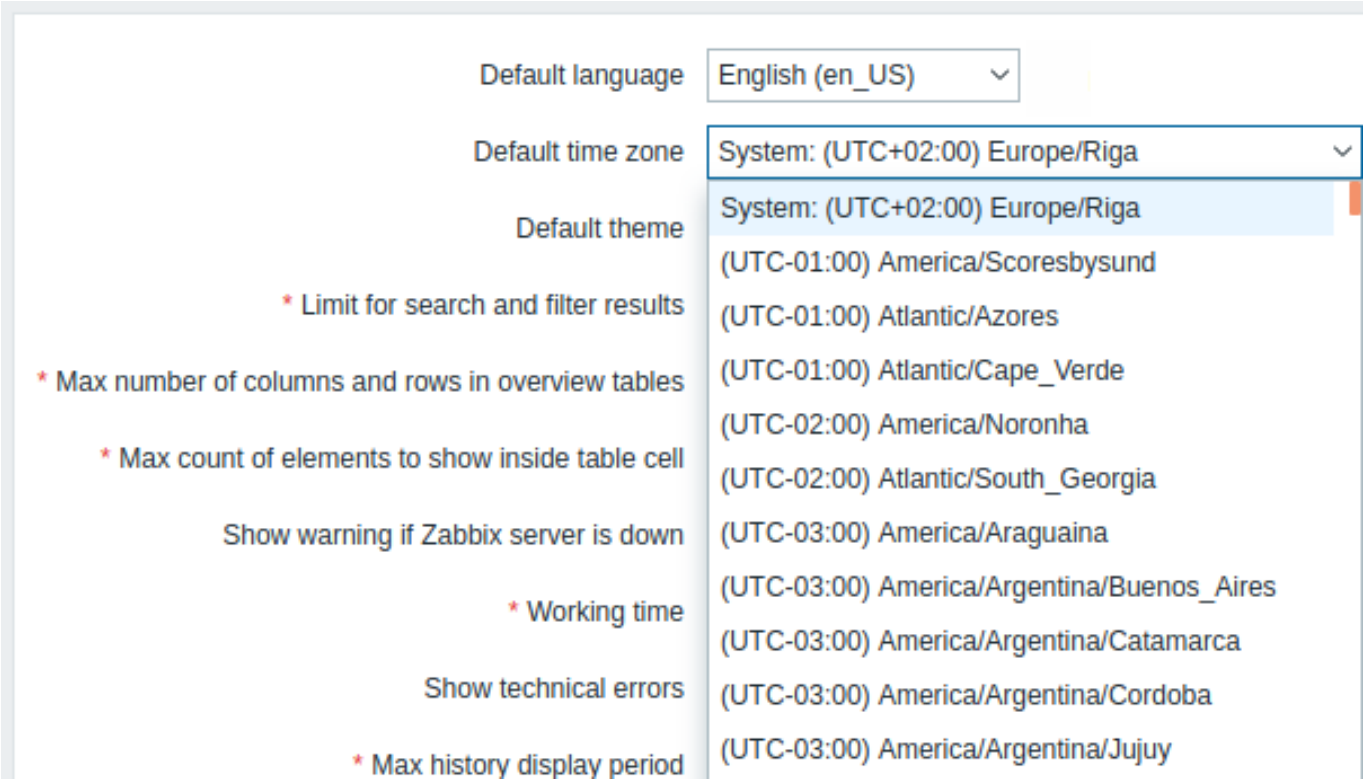
[1] Lorsque `HttpOnly` vaut 'true', le cookie sera accessible uniquement via le protocole HTTP. Cela signifie que le cookie ne sera pas accessible par des langages de script, tels que JavaScript. Ce paramètre peut contribuer efficacement à réduire le vol d'identité via des attaques XSS (bien qu'il ne soit pas pris en charge par tous les navigateurs).

[2] `Secure` indique que le cookie ne doit être transmis que via une connexion HTTPS sécurisée depuis le client. Lorsqu'il est défini sur 'true', le cookie ne sera défini que si une connexion sécurisée existe.

Fuseaux horaires

Vue d'ensemble

Le fuseau horaire de l'interface peut être défini globalement dans l'interface et ajusté pour chaque utilisateur.



Si *Système* est sélectionné, le fuseau horaire du serveur web sera utilisé pour l'interface (y compris la valeur de `date.timezone` dans `php.ini`, si elle est définie), tandis que le serveur Zabbix utilisera le fuseau horaire de la machine sur laquelle il s'exécute.

Note:
Le serveur Zabbix utilisera le fuseau horaire global ou celui de l'utilisateur spécifié uniquement lors du développement des macros dans les notifications (par exemple, `{EVENT.TIME}` peut être développé dans un fuseau horaire différent pour chaque utilisateur) et pour la limite de temps d'envoi des notifications (voir le paramètre « Quand actif » dans la configuration des médias utilisateur).

Le choix du fuseau horaire n'affecte pas le format de date et d'heure de l'interface. Vous pouvez plutôt ajuster la langue de l'interface (lors de l'installation ou dans les paramètres utilisateur) : la sélection de *Anglais (en_US)* active également le format de date et d'heure américain dans l'interface.

Configuration

Le fuseau horaire global :

- Peut être défini manuellement lors de l'installation de l'interface.
- Peut être modifié dans *Administration* > *Général* > *GUI*.

Le fuseau horaire au niveau utilisateur :

- Peut être défini lors de la configuration/mise à jour d'un utilisateur.
- Peut être défini par chaque utilisateur dans son profil utilisateur.

Voir aussi : Alignement des fuseaux horaires lors de l'utilisation des intervalles de planification.

Réinitialisation du mot de passe

Aperçu Cette section décrit les étapes de réinitialisation des mots de passe utilisateur dans Zabbix.

Étapes Adressez-vous à votre administrateur Zabbix si vous avez oublié votre mot de passe Zabbix et ne pouvez pas vous connecter.

Un utilisateur Super administrateur peut modifier les mots de passe de tous les utilisateurs dans le formulaire de configuration des utilisateurs.

Si un Super administrateur a oublié son mot de passe et ne peut pas se connecter, la requête SQL suivante doit être exécutée pour appliquer le mot de passe par défaut à l'utilisateur Super administrateur (remplacez 'Admin' par le nom d'utilisateur Super administrateur approprié) :

```
UPDATE users SET passwd = '$2a$10$ZXIvHAEP2ZM.dLXTm6uPHOMVlARXX7cqjbhM6Fn0cANzkCQBWpMrS' WHERE username =
```

Après l'exécution de cette requête, le mot de passe de l'utilisateur sera défini sur *zabbix*. Veillez à modifier le mot de passe par défaut lors de la première connexion.

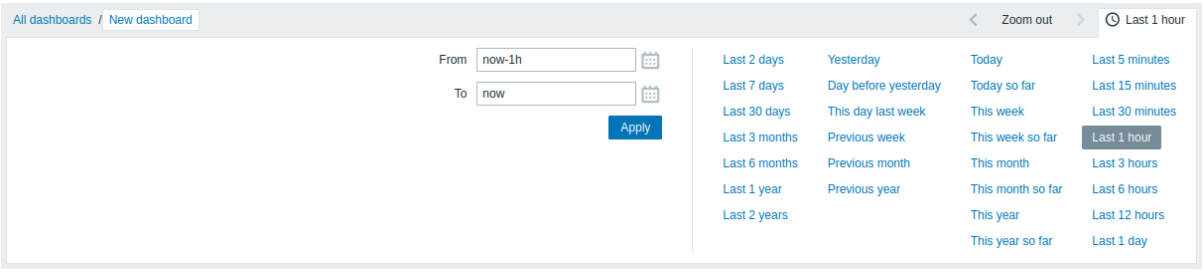
Période et sélecteurs d'hôte

Vue d'ensemble

Cette page décrit la période et les sélecteurs de hôtes disponibles sur les tableaux de bord. Ces sélecteurs permettent d'ajuster dynamiquement les données affichées dans tous les widgets de tableau de bord compatibles, sans modifier les paramètres de chaque widget.

Sélecteur de période

Le sélecteur de période vous permet de sélectionner les périodes fréquemment requises en un seul clic de souris. Vous pouvez le développer ou le réduire en cliquant sur le bouton du sélecteur de période situé dans le coin supérieur droit du tableau de bord.



Les options telles que *Aujourd'hui*, *Cette semaine*, etc. affichent la période entière, y compris les heures/jours à venir. Les options telles que *Aujourd'hui jusqu'à présent*, *Cette semaine jusqu'à présent*, etc. affichent uniquement les heures écoulées.

Une fois une période sélectionnée, vous pouvez la déplacer vers l'avant ou vers l'arrière dans le temps en cliquant sur les boutons fléchés . Le bouton *Réduire* permet de réduire la période de 50 % dans chaque direction.

Note:
Pour les graphiques (à l'exception de ceux utilisant la période *Personnalisée*), il est également possible de sélectionner la période affichée en mettant une zone en surbrillance dans le graphique avec le bouton gauche de la souris. Une fois le bouton gauche de la souris relâché, le graphique effectue un zoom sur la zone mise en surbrillance. Il est également possible de réduire le zoom en double-cliquant dans le graphique.

Les champs *De/À* affichent la période sélectionnée selon une syntaxe temporelle absolue (au format `Y-m-d H:i:s`) ou une syntaxe temporelle relative. Une période relative peut contenir une ou plusieurs opérations mathématiques (- ou +), par exemple `now-1d` ou `now-1d-2h+5m`.

Les abréviations temporelles relatives suivantes sont prises en charge :

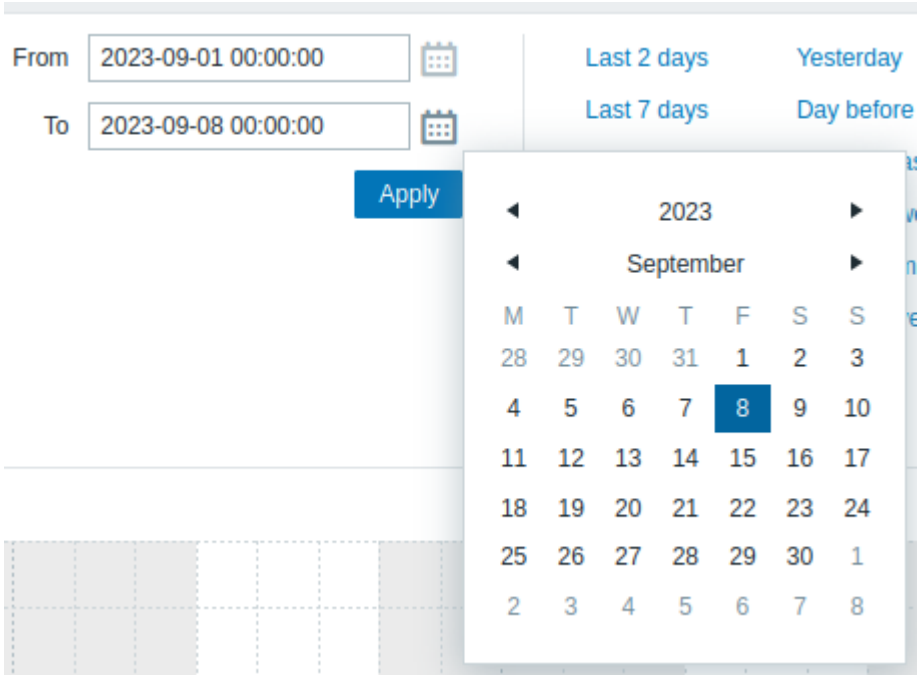
- `now`
- `s` (secondes)
- `m` (minutes)
- `h` (heures)
- `d` (jours)
- `w` (semaines)
- `M` (mois)
- `y` (années)

La précision est prise en charge dans le sélecteur de période (par exemple, `/M` dans `now-1d/M`). Détails de la précision :

Precision	From	To
`m`	Y-m-d H:m:00	Y-m-d H:m:59
`h`	Y-m-d H:00:00	Y-m-d H:59:59
`d`	Y-m-d 00:00:00	Y-m-d 23:59:59

Precision	From	To
w	Lundi de la semaine 00:00:00	Dimanche de la semaine 23:59:59
M	Premier jour du mois 00:00:00	Dernier jour du mois 23:59:59
y	1er janvier de l'année 00:00:00	31 décembre de l'année 23:59:59

Il est également possible de sélectionner une période à l'aide du sélecteur de date. Pour l'ouvrir, cliquez sur l'icône de calendrier située à côté des champs *De/À*.



Note:
Dans le sélecteur de date, vous pouvez naviguer entre l'année, le mois et la date à l'aide des touches Tab, Shift+Tab et des touches fléchées du clavier. Appuyez sur Enter pour confirmer la sélection.

Exemples :

De	À	Période sélectionnée
now/d	now/d	00:00 - 23:59 aujourd'hui
now/d	now/d+1d	00:00 aujourd'hui - 23:59 demain
now/w	now/w	Lundi 00:00:00 - dimanche 23:59:59 cette semaine
now-1y/w	now-1y/w	La semaine de lundi 00:00:00 à dimanche 23:59:59 il y a un an

Attention:
L'utilisation de « now/M+1M » pour le paramètre *À* peut ajouter 31 jours, ce qui peut entraîner un décalage de la date de 1 à 3 jours selon le nombre de jours dans le mois. Par exemple, si cette valeur est utilisée en janvier, le résultat peut être le 02 mars au lieu du 28 février attendu. Pour éviter ce problème, utilisez « now/M-3d+1M/M », qui tient correctement compte de la longueur du mois. De même, si vous configurez *De* pour remonter dans le temps, utilisez « now/M+3d-1M/M ».

Sélecteur d'hôte

Le sélecteur d'hôte vous permet de sélectionner un hôte dont les données seront affichées dans les widgets de tableau de bord compatibles.

Il devient disponible après avoir modifié et enregistré un tableau de bord, si au moins un widget est configuré avec le paramètre *Remplacer l'hôte* défini pour utiliser le tableau de bord comme source de données.



Paramètres de configuration : interface Zabbix

Vue d'ensemble Les paramètres de configuration permettent de personnaliser le comportement de l'interface Zabbix.

Indiquez les valeurs des paramètres dans le fichier de configuration de l'interface (`zabbix.conf.php`).

Cette page est structurée comme suit :

- Vue d'ensemble des paramètres (pour afficher tous les détails d'un paramètre, cliquez sur son nom)
- Détails des paramètres

Paramètre	Description
$DB['TYPE']	Type de base de données.
$DB['SERVER']	Nom d'hôte ou adresse IP de la base de données.
$DB['PORT']	Port de la base de données.
$DB['DATABASE']	Nom de la base de données utilisé par l'interface Zabbix pour se connecter à la base de données configurée.
$DB['USER']	Nom d'utilisateur utilisé par l'interface Zabbix pour se connecter à la base de données.
$DB['PASSWORD']	Mot de passe utilisé par l'interface Zabbix pour se connecter à la base de données.
$DB['SCHEMA']	Nom du schéma de base de données utilisé pour PostgreSQL.
$DB['ENCRYPTION']	Active ou désactive le chiffrement TLS des connexions entre l'interface Zabbix et la base de données.
$DB['KEY_FILE']	Chemin vers le fichier de clé privée du client utilisé pour la connexion TLS à la base de données.
$DB['CERT_FILE']	Chemin vers le fichier de certificat du client utilisé pour la connexion TLS à la base de données.
$DB['CA_FILE']	Chemin vers le fichier de certificat de l'autorité de certification utilisé pour la connexion TLS à la base de données.
$DB['VERIFY_HOST']	Active ou désactive la vérification du nom d'hôte du serveur de base de données dans le certificat TLS.
$DB['CIPHER_LIST']	Liste des chiffrements TLS autorisés pour la connexion à la base de données.
$DB['VAULT']	Indique le gestionnaire de secrets Vault utilisé pour stocker les identifiants de la base de données.
$DB['VAULT_URL']	URL du gestionnaire de secrets Vault utilisé pour récupérer les identifiants de la base de données.
$DB['VAULT_PREFIX']	Préfixe du chemin ou de la requête Vault utilisé pour récupérer les identifiants de la base de données.
$DB['VAULT_DB_PATH']	Chemin vers les identifiants de la base de données dans le gestionnaire de secrets Vault.
$DB['VAULT_TOKEN']	Jeton d'authentification utilisé pour accéder au gestionnaire de secrets Vault.
$DB['VAULT_APP_ROLE_ID']	ID de rôle AppRole utilisé pour s'authentifier auprès de HashiCorp Vault.
$DB['VAULT_APP_SECRET_ID']	ID secret AppRole utilisé pour s'authentifier auprès de HashiCorp Vault.
$DB['VAULT_CERT_FILE']	Chemin vers le fichier de certificat du client utilisé pour s'authentifier auprès du gestionnaire de secrets Vault.
$DB['VAULT_KEY_FILE']	Chemin vers le fichier de clé privée du client utilisé pour s'authentifier auprès du gestionnaire de secrets Vault.
$DB['VAULT_CACHE']	Active ou désactive la mise en cache des identifiants récupérés depuis le gestionnaire de secrets Vault.
$ZBX_SERVER	Indique une valeur pour remplacer le nom d'hôte ou l'adresse IP de Zabbix.
$ZBX_SERVER_PORT	Indique une valeur pour remplacer le port du serveur Zabbix.
$ZBX_SERVER_NAME	Nom du serveur Zabbix.
$IMAGE_FORMAT_DEFAULT	Format d'image par défaut.
$HISTORY_PROVIDERS[]	Configuration des fournisseurs de stockage de l'historique pour Elasticsearch ou ClickHouse.
$SSO['SETTINGS']	Paramètres de configuration pour la connexion SAML.
$SSO['CERT_STORAGE']	Indique où sont stockés les certificats SAML.
$SSO['SP_KEY']	Chemin vers le fichier de clé privée du fournisseur de services (SP) SAML lorsque la clé privée est stockée dans le système de fichiers.
$SSO['SP_CERT']	Chemin vers le fichier de certificat du fournisseur de services (SP) SAML lorsque le certificat est stocké dans le système de fichiers.
$SSO['IDP_CERT']	Chemin vers le fichier de certificat du fournisseur d'identité SAML lorsque le certificat est stocké dans le système de fichiers.
$ZBX_FEATURE_FLAGS['banners_enabled']	Active ou désactive la prise en charge des bannières.
$ZBX_FEATURE_FLAGS['http_auth_enabled']	Active ou désactive l'authentification HTTP des utilisateurs.
$ZBX_FEATURE_FLAGS['module_config_enabled']	Active ou désactive l'accès aux modules.
$ZBX_FEATURE_FLAGS['media_type_legacy']	Désactive la modification de types de média préconfigurés spécifiques.
$ZBX_SERVER_TLS['ACTIVE']	Indique si TLS est activé pour les connexions au serveur Zabbix.

Paramètre	Description
$ZBX_SERVER_TLS['CA_FILE']	Chemin vers le fichier de certificat de l'autorité de certification utilisé pour vérifier le certificat du serveur.
$ZBX_SERVER_TLS['KEY_FILE']	Chemin vers le fichier de clé privée du client correspondant au certificat du client.
$ZBX_SERVER_TLS['CERT_FILE']	Chemin vers le fichier de certificat du client lorsque l'authentification TLS mutuelle est requise.
$ZBX_SERVER_TLS['CERTIFICATE_ISSUER']	Nom distinctif (DN) de l'émetteur à comparer au certificat du serveur.
$ZBX_SERVER_TLS['CERTIFICATE_SUBJECT']	Nom distinctif (DN) du sujet à comparer au certificat du serveur.
$TELEMETRY_PROVIDERS[]	Configuration d'une source de données de télémétrie.

Détails des paramètres

$DB['TYPE']

Type de base de données.

Valeurs : `MYSQL` - MySQL, `POSTGRESQL` - PostgreSQL.

$DB['SERVER']

Nom d'hôte ou adresse IP de la base de données.

Si `$DB['SERVER']` n'est pas défini, le socket Unix par défaut est utilisé. Pour MySQL, définir `$DB['SERVER']` sur `localhost` entraîne également une connexion via le socket Unix par défaut.
 Pour PostgreSQL, `$DB['SERVER']` peut contenir plusieurs entrées `host:port` séparées par des virgules. Si aucun port n'est défini pour un hôte dans la liste séparée par des virgules, la valeur définie dans `$DB['PORT']` est utilisée comme port par défaut. Si un chemin de socket est spécifié dans `$DB['SERVER']` (par exemple, `/var/run/pgbouncer`), ce socket Unix est utilisé.

Exemples :

```
$DB['SERVER'] = 'localhost';
$DB['SERVER'] = 127.0.0.1;
```

$DB['PORT']

Port de la base de données.

Si cette valeur n'est pas définie, la valeur 0 est utilisée.

$DB['DATABASE']

Nom de la base de données utilisé par l'interface Zabbix pour se connecter à la base de données configurée.

Exemple :

```
$DB['DATABASE'] = 'zabbix';
```

$DB['USER']

Nom d'utilisateur utilisé par l'interface Zabbix pour se connecter à la base de données.

$DB['PASSWORD']

Mot de passe utilisé par l'interface Zabbix pour se connecter à la base de données.

$DB['SCHEMA']

Nom du schéma de base de données utilisé pour PostgreSQL.

$DB['ENCRYPTION']

Active ou désactive le chiffrement des connexions entre l'interface Zabbix et la base de données.

Valeurs : `true` - activé, `false` - désactivé.

$DB['KEY_FILE']

Chemin vers le fichier de clé privée du client utilisé pour la connexion TLS à la base de données.

Exemple :

```
// Pour MySQL :
$DB['KEY_FILE'] = '/etc/ssl/mysql/client-key.pem';
```

$DB['CERT_FILE']

Chemin vers le fichier de certificat client utilisé pour la connexion TLS à la base de données.

Exemple :

```
// Pour MySQL :
$DB['CERT_FILE'] = '/etc/ssl/mysql/client-cert.pem';
```

$DB['CA_FILE']

Chemin vers le fichier de certificat CA utilisé pour la connexion TLS à la base de données.

Exemples :

```
// Pour MySQL :
$DB['CA_FILE'] = '/etc/ssl/mysql/ca.pem';
// Pour PostgreSQL :
$DB['CA_FILE'] = '/etc/ssl/pgsql/root.crt';
```

$DB['VERIFY_HOST']

Active ou désactive la vérification du nom d'hôte du serveur de base de données dans le certificat TLS.

Valeurs : `true` - activée, `false` - désactivée.

$DB['CIPHER_LIST']

Liste des chiffrements TLS autorisés pour la connexion à la base de données.

Exemples :

```
// Pour MySQL :
$DB['CIPHER_LIST'] = 'TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256:TLS_AES_1
```

$DB['VAULT']

Spécifiez le gestionnaire de secrets Vault utilisé pour stocker les identifiants de la base de données.

Valeurs : `CyberArk` ou `HashiCorp`.

Exemples :

```
// Pour CyberArk :
$DB['VAULT'] = 'CyberArk';
// Pour HashiCorp :
$DB['VAULT'] = 'HashiCorp';
```

$DB['VAULT_URL']

URL du gestionnaire de secrets Vault utilisé pour récupérer les identifiants de la base de données.

Exemples :

```
// Pour CyberArk :
$DB['VAULT_URL'] = 'https://127.0.0.1:1858';
// Pour HashiCorp :
$DB['VAULT_URL'] = 'https://localhost:8200';
```

$DB['VAULT_PREFIX']

Préfixe du chemin ou de la requête Vault utilisé pour récupérer les identifiants de la base de données.

Exemple :

```
// Pour HashiCorp :
$DB['VAULT_PREFIX'] = '/v1/secret/data/zabbix/';
```

$DB['VAULT_DB_PATH']

Chemin d'accès aux identifiants de la base de données dans le gestionnaire de secrets Vault.

Exemples :

```
// Pour CyberArk :
$DB['VAULT_DB_PATH'] = 'AppID=foo&Query=Safe=bar;Object=buzz';
// Pour HashiCorp :
$DB['VAULT_DB_PATH'] = 'database';
```

$DB['VAULT_TOKEN']

Jeton d'authentification utilisé pour accéder au gestionnaire de secrets Vault.

Exemple :

```
// Pour HashiCorp :
$DB['VAULT_TOKEN'] = 'hvs.CAESIIG_PILmULFYOsEyWHxkZ2mF2a8VPKNLE8eHqd4autYGGh4KHGh2cy5aeTYONFNSaUp3ZnpWbDF1
```

$DB['VAULT_APP_ROLE_ID']

ID de rôle VaultAppSecretID utilisé pour s'authentifier auprès de HashiCorp Vault.

Exemple :

```
$DB['VAULT_APP_ROLE_ID'] = '6f1567f1-538d-7d70-e8e4-5d3e59471b2e';
```

$DB['VAULT_APP_SECRET_ID']

VaultAppSecretID utilisé pour s'authentifier auprès de HashiCorp Vault.

Exemple :

```
$DB['VAULT_APP_SECRET_ID'] = '89dac8b1-f7d6-96a7-2ecf-f94d2de3d6f1';
```

$DB['VAULT_CERT_FILE']

Chemin vers le fichier de certificat client utilisé pour s'authentifier auprès du gestionnaire de secrets Vault.

Exemple :

```
// Pour CyberArk :
$DB['VAULT_CERT_FILE'] = 'conf/certs/cyberark-cert.pem';
```

$DB['VAULT_KEY_FILE']

Chemin vers le fichier de clé privée du client utilisé pour s'authentifier auprès du gestionnaire de secrets Vault.

Exemple :

```
// Pour CyberArk :
$DB['VAULT_KEY_FILE'] = 'conf/certs/cyberark-key.pem';
```

$DB['VAULT_CACHE']

Active ou désactive la mise en cache des identifiants récupérés depuis le gestionnaire de secrets Vault.

Valeurs : `true` - activée, `false` - désactivée.

$ZBX_SERVER

Indiquez une valeur pour remplacer le nom d'hôte ou l'adresse IP de Zabbix.
 Si `$ZBX_SERVER` est spécifié sans `$ZBX_SERVER_PORT`, le port par défaut (10051) est utilisé.

$ZBX_SERVER_PORT

Spécifiez une valeur pour remplacer le port du serveur Zabbix.
 Ce paramètre est utilisé avec `$ZBX_SERVER`.

$ZBX_SERVER_NAME

Nom du serveur Zabbix.

$IMAGE_FORMAT_DEFAULT

Format d'image par défaut.

Par défaut : `IMAGE_FORMAT_PNG`
 Valeurs : - `IMAGE_FORMAT_PNG` - Format PNG - `IMAGE_FORMAT_JPEG` - Format JPEG - `IMAGE_FORMAT_GIF` - Format GIF

$HISTORY_PROVIDERS[]

Configuration des fournisseurs de stockage de l'historique pour Elasticsearch ou ClickHouse.

Paramètres de configuration pris en charge : - `types` - tableau des types de données à stocker dans le stockage externe ; - `provider` - type de fournisseur d'historique : `elasticsearch` ou `clickhouse` ; - `url` - URL du fournisseur d'historique ; - `db` - nom de la base de données (utilisé pour ClickHouse) ; - `username` - utilisateur de la base de données (utilisé pour ClickHouse) ; - `password` - mot de passe de la base de données (utilisé pour ClickHouse).

Exemples :

```
// Pour ClickHouse :
$HISTORY_PROVIDERS[] = [
    'types' => ['uint', 'dbl', 'str', 'log', 'text', 'json'],
    'provider' => 'clickhouse',
    'url' => 'http://localhost:8123',
    'db' => 'zabbix',
    'username' => 'zabbix',
    'password' => 'zabbix'
];
// Pour Elasticsearch :
$HISTORY_PROVIDERS[] = [
    'types' => ['uint', 'dbl', 'str', 'log', 'text', 'json'],
    'provider' => 'elasticsearch',
    'url' => 'http://localhost:9200'
];
```

$SSO['SETTINGS']

Paramètres de configuration pour la connexion SAML. Le tableau `$SSO['SETTINGS']` doit respecter la structure attendue par la bibliothèque *SAML PHP Toolkit* (fournie avec Zabbix).

Seules les options suivantes peuvent être spécifiées dans `$SSO['SETTINGS']` :

- `strict`
- `baseurl`
- `compress`
- `contactPerson`
- `organization`
- `sp` (uniquement les options spécifiées dans cette liste)
  - `attributeConsumingService`
  - `x509certNew`
- `idp` (uniquement les options spécifiées dans cette liste)
  - `singleLogoutService` (une seule option)
    * `responseUrl`
  - `certFingerprint`
  - `certFingerprintAlgorithm`
  - `x509certMulti`
- `security` (uniquement les options spécifiées dans cette liste)
  - `signMetadata`
  - `wantNameId`
  - `requestedAuthnContext`
  - `requestedAuthnContextComparison`
  - `wantXMLValidation`
  - `relaxDestinationValidation`
  - `destinationStrictlyMatches`
  - `rejectUnsolicitedResponsesWithInResponseTo`
  - `signatureAlgorithm`
  - `digestAlgorithm`
  - `lowercaseUrlencoding`

Toutes les autres options sont récupérées depuis la base de données et ne peuvent pas être remplacées. L'option `debug` sera ignorée.

Si l'interface Zabbix se trouve derrière un proxy ou un répartiteur de charge, l'option personnalisée `use_proxy_headers` peut être utilisée :

- `false` (par défaut) - ignorer les en-têtes HTTP `X-Forwarded-*` ;
- `true` - utiliser les en-têtes HTTP `X-Forwarded-*` pour construire l'URL de base.

Pour obtenir une description complète des options de configuration disponibles, consultez la documentation officielle de la bibliothèque.

Exemple :

```
$SSO['SETTINGS'] = [
    'security' => [
        'signatureAlgorithm' => 'http://www.w3.org/2001/04/xmldsig-more#rsa-sha384',
```

```
        'digestAlgorithm' => 'http://www.w3.org/2001/04/xmldsig-more#sha384',
        // ...
    ],
    // ...
];
```

$SSO['CERT_STORAGE']

Indiquez où les certificats SAML sont stockés.

Par défaut : `file`
 Valeurs : - `database` - la clé privée et les certificats sont stockés dans la base de données ; - `file` - la clé privée et les certificats sont stockés dans le système de fichiers.

Exemple :

```
$SSO['CERT_STORAGE'] = 'database';
```

$SSO['SP_KEY']

Chemin vers le fichier de clé privée du fournisseur de services SAML (SP) lorsque la clé privée est stockée dans le système de fichiers.

Exemple :

```
$SSO['SP_KEY'] = 'conf/certs/sp.key';
```

$SSO['SP_CERT']

Chemin vers le fichier de certificat du fournisseur de services SAML (SP) lorsque le certificat est stocké dans le système de fichiers.

Exemple :

```
$SSO['SP_CERT'] = 'conf/certs/sp.crt';
```

$SSO['IDP_CERT']

Chemin vers le fichier de certificat du fournisseur d'identité SAML lorsque le certificat est stocké dans le système de fichiers.

Exemple :

```
$SSO['IDP_CERT'] = 'conf/certs/idp.crt';
```

$ZBX_FEATURE_FLAGS['banners_enabled']

Active ou désactive la prise en charge des bannières.

Par défaut : `true`
 Valeurs : `true` - activée, `false` - désactivée.

$ZBX_FEATURE_FLAGS['http_auth_enabled']

Active ou désactive l'authentification HTTP des utilisateurs.

Par défaut : `true`
 Valeurs : `true` - activée, `false` - désactivée.

$ZBX_FEATURE_FLAGS['modules_config_enabled']

Active ou désactive l'accès aux modules.

Par défaut : `true`
 Valeurs : `true` - activé, `false` - désactivé.

$ZBX_FEATURE_FLAGS['media_type_denylist']

Désactivez la modification de certains types de média préconfigurés.

Valeurs : `email`, `script`, `sms`, `webhook`, `push`.

Exemple :

```
$ZBX_FEATURE_FLAGS['media_type_denylist'] = ['webhook','push'];
```

$ZBX_SERVER_TLS['ACTIVE']

Indique si une connexion TLS entre l'interface Zabbix et le serveur Zabbix est activée.

Valeur par défaut : `false`
 Valeurs : `true` - activée, `false` - désactivée.

$ZBX_SERVER_TLS['CA_FILE']

Chemin vers le fichier de certificat de l'autorité de certification (CA) utilisé pour vérifier le certificat du serveur.

Exemple :

```
$ZBX_SERVER_TLS['CA_FILE'] = '/etc/zabbix/ssl/ca.crt';
```

$ZBX_SERVER_TLS['KEY_FILE']

Chemin vers le fichier de clé privée du client correspondant au certificat client utilisé pour l'authentification TLS mutuelle.

Exemple :

```
$ZBX_SERVER_TLS['KEY_FILE'] = '/etc/zabbix/ssl/frontend.key';
```

$ZBX_SERVER_TLS['CERT_FILE']

Chemin vers le fichier de certificat client lorsque l'authentification TLS mutuelle est requise.

Exemple :

```
$ZBX_SERVER_TLS['CERT_FILE'] = '/etc/zabbix/ssl/frontend.crt';
```

$ZBX_SERVER_TLS['CERTIFICATE_ISSUER']

Nom distinctif (DN) de l'émetteur à comparer au certificat du serveur.

Exemple :

```
$ZBX_SERVER_TLS['CERTIFICATE_ISSUER']  = 'CN=MyZabbixCA';
```

$ZBX_SERVER_TLS['CERTIFICATE_SUBJECT']

Nom distinctif (DN) du sujet à comparer au certificat du serveur.

Exemple :

```
$ZBX_SERVER_TLS['CERTIFICATE_SUBJECT'] = 'CN=zabbix-server.example.com';
```

$TELEMETRY_PROVIDERS[]

Configuration d'une source de données de télémétrie.

Paramètres de configuration pris en charge pour ClickHouse : - `url` - définit l'URL HTTP ou HTTPS utilisée pour se connecter à la source de données de télémétrie. - `db` - définit le nom de la base de données. - `username` - définit le nom d'utilisateur utilisé pour se connecter à la base de données. - `password` - définit le mot de passe utilisé pour se connecter à la base de données. - `ssl_cert_file`, `ssl_key_file`, `ssl_key_password`, `ssl_verify_peer`, `ssl_verify_host`, `ssl_ca_location`, `ssl_ca_file` - définissent ces options pour configurer une connexion sécurisée (TLS/SSL) à la base de données. - `vault_path` - définit le chemin vers un secret Vault contenant les valeurs `username` et `password`. Ne pas utiliser conjointement avec les options `username` et `password`.

Exemple :

```
$TELEMETRY_PROVIDERS[] = [
    'provider' => 'clickhouse',
    'url' => 'http://localhost:8123',
    'db' => 'zabbix',
    'username' => 'zabbix',
    'password' => 'zabbix'
];
```

API

Vue d'ensemble L'API Zabbix vous permet de récupérer et de modifier par programmation la configuration de Zabbix et fournit un accès aux données historiques. Elle est largement utilisée pour :

- Créer de nouvelles applications pour fonctionner avec Zabbix.
- Intégrer Zabbix dans un logiciel tiers.
- Automatiser les tâches de routine.

L'API Zabbix est une API basée sur HTTP, et elle est fournie en tant que partie de l'interface. Elle utilise le protocole JSON-RPC 2.0, ce qui signifie deux choses :

- L'API se compose d'un ensemble de méthodes distinctes.

- Les requêtes et les réponses entre les clients et l'API sont encodées au format JSON.

Pour plus d'informations sur le protocole et JSON, consultez la spécification JSON-RPC 2.0 et la page d'accueil du format JSON.

Pour plus d'informations sur l'intégration des fonctionnalités Zabbix dans vos applications Python, consultez la bibliothèque Python pour Zabbix.

> **Note:**
> L'accès des utilisateurs dans Zabbix, y compris la configuration et les données historiques, dépend du type d'utilisateur, du rôle utilisateur attribué et des groupes d'utilisateurs.

Structure L'API se compose d'un certain nombre de méthodes, regroupées nominalement en API distinctes. Chacune des méthodes effectue une tâche spécifique. Par exemple, la méthode `host.create` appartient à l'API *host* et est utilisée pour créer de nouveaux hôtes. Historiquement, les API sont parfois appelées "classes".

La plupart des API contiennent au moins quatre méthodes: `get`, `create`, `update` et `delete` pour récupérer, créer, mettre à jour et supprimer des données respectivement, mais certaines API peuvent fournir un ensemble de méthodes totalement différent.

Exécution de requêtes Une fois l'interface configurée, vous pouvez utiliser des requêtes HTTP distantes pour appeler l'API. Pour ce faire, vous devez envoyer des requêtes HTTP POST au fichier `api_jsonrpc.php` situé dans le répertoire de l'interface.
 Par exemple, si votre interface Zabbix est installée sous `https://example.com/zabbix`, une requête HTTP pour appeler la méthode `apiinfo.version` peut ressembler à ceci :

```
curl --request POST \
  --url 'https://example.com/zabbix/api_jsonrpc.php' \
  --header 'Content-Type: application/json-rpc' \
  --data '{"jsonrpc":"2.0","method":"apiinfo.version","params":{},"id":1}'
```

La requête doit avoir l'en-tête `Content-Type` défini sur l'une de ces valeurs : `application/json-rpc`, `application/json` ou `application/jsonrequest`.

L'objet de requête doit contenir les propriétés suivantes :

- `jsonrpc` - la version du protocole JSON-RPC utilisée par l'API (l'API Zabbix implémente la version 2.0 de JSON-RPC) ;
- `method` - la méthode de l'API appelée ;
- `params` - les paramètres qui seront transmis à la méthode de l'API ;
- `id` - un identifiant arbitraire de la requête (s'il est omis, l'API traite la requête comme une notification).

Si la requête est correcte, la réponse renvoyée par l'API doit ressembler à ceci :

```
{
    "jsonrpc": "2.0",
    "result": "8.0.0",
    "id": 1
}
```

L'objet de réponse, quant à lui, contient les propriétés suivantes :

- `jsonrpc` - la version du protocole JSON-RPC ;
- `result` - les données renvoyées par la méthode ;
- `id` - un identifiant de la requête correspondante.

Authentification Pour accéder à des données dans Zabbix, vous devez soit :

- Utiliser un jeton API existant créé dans l'interface Zabbix, section *Users* > *API tokens*, ou créé à l'aide de l'API Token.
- Utiliser un jeton d'authentification obtenu avec la méthode `user.login`.

Par exemple, si vous souhaitez obtenir un nouveau jeton d'authentification en vous connectant en tant qu'utilisateur *Admin* standard, une requête JSON ressemblerait à ceci :

```
curl --request POST \
  --url 'https://example.com/zabbix/api_jsonrpc.php' \
  --header 'Content-Type: application/json-rpc' \
  --data '{"jsonrpc":"2.0","method":"user.login","params":{"username":"Admin","password":"zabbix"},"id":1}
```

Si vous avez fourni correctement les identifiants, la réponse renvoyée par l'API doit contenir le jeton d'authentification de l'utilisateur :

```
{
    "jsonrpc": "2.0",
    "result": "0424bd59b807674191e7d77572075f33",
    "id": 1
}
```

Méthodes d'autorisation Par l'en-tête « Authorization »

Toutes les requêtes API nécessitent une authentification ou un jeton API. Vous pouvez fournir les identifiants en utilisant l'en-tête Authorization dans la requête :

```
curl --request POST \
  --url 'https://example.com/zabbix/api_jsonrpc.php' \
  --header 'Authorization: Bearer 0424bd59b807674191e7d77572075f33'
```

Attention:
Si vous rencontrez des problèmes d'authentification, consultez Authorization header forwarding.

L'API Zabbix accepte les en-têtes sans tenir compte de la casse (par exemple, `authorization`, `Authorization` et `AUTHORIZATION` sont traités de la même manière).

L'en-tête Authorization est pris en charge dans les requêtes cross-origin (CORS).

Par le cookie Zabbix

Un cookie *"zbx_session"* est utilisé pour autoriser une requête API depuis l'interface utilisateur Zabbix effectuée à l'aide de JavaScript (à partir d'un module ou d'un widget personnalisé).

Exemple de flux de travail La section suivante vous guide à travers plusieurs exemples d'utilisation plus en détail.

Récupération des hôtes Vous disposez maintenant d'un jeton d'authentification utilisateur valide (représenté comme une variable dans les exemples suivants) qui peut être utilisé pour accéder aux données dans Zabbix. Par exemple, vous pouvez utiliser la méthode `host.get` pour récupérer les ID, les noms d'hôte et les interfaces de tous les hôtes configurés :

Requête :

```
curl --request POST \
  --url 'https://example.com/zabbix/api_jsonrpc.php' \
  --header 'Authorization: Bearer ${AUTHORIZATION_TOKEN}' \
  --header 'Content-Type: application/json-rpc' \
  --data @data.json
```

Note:
`data.json` est un fichier qui contient une requête JSON. Au lieu d'un fichier, vous pouvez passer la requête dans l'argument `--data`.

`data.json`

```
{
    "jsonrpc": "2.0",
    "method": "host.get",
    "params": {
        "output": [
            "hostid",
            "host"
        ],
        "selectInterfaces": [
            "interfaceid",
            "ip"
        ]
    },
    "id": 2
}
```

L'objet de réponse contiendra les données demandées sur les hôtes :

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "hostid": "10084",
            "host": "Zabbix server",
            "interfaces": [
                {
                    "interfaceid": "1",
                    "ip": "127.0.0.1"
                }
            ]
        }
    ],
    "id": 2
}
```


Note:
Pour des raisons de performance, il est toujours recommandé de lister les propriétés de l'objet que vous souhaitez récupérer. Ainsi, vous éviterez de tout récupérer.

Création d'un nouvel élément Maintenant, créez un nouvel élément sur l'hôte "Zabbix server" en utilisant les données que vous avez obtenues à partir de la requête `host.get` précédente. Cela peut être fait à l'aide de la méthode `item.create` :

```
curl --request POST \
  --url 'https://example.com/zabbix/api_jsonrpc.php' \
  --header 'Authorization: Bearer ${AUTHORIZATION_TOKEN}' \
  --header 'Content-Type: application/json-rpc' \
  --data '{"jsonrpc":"2.0","method":"item.create","params":{"name":"Free disk space on /home/joe/","key_":
```

Une réponse réussie contiendra l'ID du nouvel élément créé, qui pourra être utilisé pour référencer l'élément dans les requêtes suivantes :


```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "24759"
        ]
    },
    "id": 3
}
```


Note:
La méthode `item.create`, ainsi que les autres *méthodes de création*, peuvent également accepter des tableaux d'objets et créer plusieurs éléments avec un seul appel API.

Création de plusieurs déclencheurs Ainsi, si les *méthodes de création* acceptent des tableaux, vous pouvez ajouter plusieurs déclencheurs, par exemple celui-ci :

```
curl --request POST \
  --url 'https://example.com/zabbix/api_jsonrpc.php' \
  --header 'Authorization: Bearer ${AUTHORIZATION_TOKEN}' \
  --header 'Content-Type: application/json-rpc' \
  --data '{"jsonrpc":"2.0","method":"trigger.create","params":[{"description":"Processor load is too high
```

La réponse réussie contiendra les identifiants des déclencheurs nouvellement créés :


```
{
    "jsonrpc": "2.0",
    "result": {
        "triggerids": [
            "17369",
```

```
            "17370"
        ]
    },
    "id": 4
}
```

Mise à jour d'un élément Activez un élément en définissant son statut sur 0 :

```
curl --request POST \
  --url 'https://example.com/zabbix/api_jsonrpc.php' \
  --header 'Authorization: Bearer ${AUTHORIZATION_TOKEN}' \
  --header 'Content-Type: application/json-rpc' \
  --data '{"jsonrpc":"2.0","method":"item.update","params":{"itemid":"10092","status":0},"id":5}'
```

La réponse réussie contiendra l'ID de l'élément mis à jour :

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "10092"
        ]
    },
    "id": 5
}
```

> **Note:**
> La méthode `item.update`, ainsi que les autres *méthodes de mise à jour*, peut également accepter des tableaux d'objets et mettre à jour plusieurs éléments avec un seul appel API.

Mise à jour de plusieurs déclencheurs Activez plusieurs déclencheurs en définissant leur statut sur 0 :

```
curl --request POST \
  --url 'https://example.com/zabbix/api_jsonrpc.php' \
  --header 'Authorization: Bearer ${AUTHORIZATION_TOKEN}' \
  --header 'Content-Type: application/json-rpc' \
  --data '{"jsonrpc":"2.0","method":"trigger.update","params":[{"triggerid":"13938","status":0},{"triggeri
```

La réponse réussie contiendra les identifiants des déclencheurs mis à jour :

```
{
    "jsonrpc": "2.0",
    "result": {
        "triggerids": [
            "13938",
            "13939"
        ]
    },
    "id": 6
}
```

Gestion des erreurs Jusqu'à présent, tout ce que vous avez essayé a fonctionné correctement. Mais que se passerait-il si vous essayiez d'effectuer un appel incorrect à l'API ? Essayez de créer un autre hôte en appelant `host.create` tout en omettant le paramètre obligatoire `groups` :

```
curl --request POST \
  --url 'https://example.com/zabbix/api_jsonrpc.php' \
  --header 'Authorization: Bearer ${AUTHORIZATION_TOKEN}' \
  --header 'Content-Type: application/json-rpc' \
  --data '{"jsonrpc":"2.0","method":"host.create","params":{"host":"Linux server","interfaces":[{"type":1,
```

La réponse contiendra alors un message d'erreur :

```
{
    "jsonrpc": "2.0",
```

```
    "error": {
        "code": -32602,
        "message": "Invalid params.",
        "data": "No groups for host \"Linux server\"."
    },
    "id": 7
}
```

Si une erreur s'est produite, au lieu de la propriété `result`, l'objet de réponse contiendra la propriété `error` avec les données suivantes :

- `code` - un code d'erreur ;
- `message` - un bref résumé de l'erreur ;
- `data` - un message d'erreur plus détaillé.

Des erreurs peuvent survenir dans divers cas, par exemple lors de l'utilisation de valeurs d'entrée incorrectes, d'un délai d'expiration de session ou d'une tentative d'accès à des objets inexistants. Votre application doit être capable de gérer correctement ce type d'erreurs.

Versions de l'API Pour simplifier le versionnement de l'API, depuis Zabbix 2.0.4, la version de l'API correspond à la version de Zabbix elle-même. Vous pouvez utiliser la méthode `apiinfo.version` pour connaître la version de l'API avec laquelle vous travaillez. Cela peut être utile pour adapter votre application afin d'utiliser des fonctionnalités spécifiques à une version.

Zabbix garantit la compatibilité ascendante des fonctionnalités au sein d'une version majeure. Lors de l'introduction de changements incompatibles entre versions majeures, Zabbix laisse généralement les anciennes fonctionnalités comme obsolètes dans la version suivante, puis ne les supprime que dans la version d'après. Il peut arriver que Zabbix supprime des fonctionnalités entre des versions majeures sans fournir de compatibilité ascendante. Il est important de ne jamais dépendre de fonctionnalités obsolètes et de migrer vers des alternatives plus récentes dès que possible.

> **Note:**
> Vous pouvez suivre toutes les modifications apportées à l'API dans le journal des modifications de l'API.

Lecture complémentaire À présent, vous disposez de suffisamment de connaissances pour commencer à travailler avec l'API Zabbix, cependant, ne vous arrêtez pas là. Pour aller plus loin, nous vous conseillons de consulter la liste des API disponibles.

Référence des méthodes

Cette section fournit une vue d'ensemble des fonctions proposées par la Zabbix API et vous aidera à vous repérer parmi les classes et méthodes disponibles.

Surveillance L'API Zabbix vous permet d'accéder à l'historique et à d'autres données collectées pendant la surveillance.

Tableaux de bord

Gérez les tableaux de bord et créez des rapports planifiés à partir de ceux-ci.

API Dashboard | API des tableaux de bord de modèle | API Report

Cluster haute disponibilité

Récupérez une liste des nœuds du serveur et leur statut.

API du cluster haute disponibilité

Historique

Récupérez les valeurs historiques collectées par les processus de surveillance Zabbix pour les présenter ou les traiter ultérieurement.

API History

Tendances

Récupérez les valeurs de tendance calculées par le serveur Zabbix pour leur présentation ou leur traitement ultérieur.

API des tendances

Événements

Récupérez les événements générés par les déclencheurs, la découverte réseau et d'autres systèmes Zabbix pour une gestion plus flexible des situations ou l'intégration d'outils tiers.

API des événements

Problèmes

Récupérez les problèmes selon les paramètres donnés.

API Problem

Cartes

Configurez des cartes pour créer des représentations dynamiques détaillées de votre infrastructure informatique.

API des cartes

Tâches

Interagissez avec le gestionnaire de tâches du serveur Zabbix, en créant des tâches et en récupérant la réponse.

API des tâches

APM L'API Zabbix vous permet d'accéder aux données de télémétrie collectées pour la surveillance des performances des applications (APM).

Traces

Récupérer les traces et les spans qui les composent.

API des traces APM | API des spans APM

Métriques

Récupérer les points de données des métriques.

API des métriques APM

Journaux

Récupérer les enregistrements de journaux.

API des journaux APM

Services L'API Zabbix vous permet d'accéder aux données collectées pendant la surveillance des services.

Accord de niveau de service

Définissez des objectifs de niveau de service (SLO), récupérez des informations détaillées sur les indicateurs de niveau de service (SLI) concernant les performances du service.

SLA API

Services

Gérez les services pour la surveillance des niveaux de service et récupérez des informations détaillées sur les SLA de n'importe quel service.

API des services

Collecte des données L'API Zabbix vous permet de gérer la configuration de votre système de supervision.

Hôtes et groupes d'hôtes

Gérez les groupes d'hôtes, les hôtes et tout ce qui s'y rapporte, notamment les interfaces des hôtes, les macros d'hôte et les périodes de maintenance.

API des hôtes | API des groupes d'hôtes | API des macros utilisateur | API des mappages de valeurs | API de la maintenance

Éléments

Définissez les éléments à surveiller.

API des éléments

Déclencheurs

Configurez les déclencheurs pour vous informer des problèmes dans votre système. Gérez les dépendances entre déclencheurs.

API des déclencheurs

Graphiques

Modifiez les graphiques ou les éléments de graphique séparés pour une meilleure présentation des données collectées.

API Graph | API Graph item

Modèles et groupes de modèles

Gérez les modèles et liez-les à des hôtes ou à d'autres modèles.

API des modèles | API des groupes de modèles | API des correspondances de valeurs

Découverte de bas niveau

Configurez des règles de découverte de bas niveau ainsi que des prototypes d'élément, de déclencheur et de graphique pour surveiller des entités dynamiques.

API des règles LLD | API des prototypes d'élément | API des prototypes de déclencheur | API des prototypes de graphique | API des prototypes d'hôte

Complex event processing

Create complex event processing rules.

Complex event processing API

Corrélation d'événements

Créez des règles de corrélation d'événements personnalisées.

API de corrélation

Découverte du réseau

Gérez les règles de découverte au niveau du réseau pour trouver et surveiller automatiquement de nouveaux hôtes. Bénéficiez d'un accès complet aux informations sur les services et les hôtes découverts.

API des règles de découverte | API des vérifications de découverte | API des hôtes découverts | API des services découverts

Exportation et importation

Exportez et importez les données de configuration Zabbix pour les sauvegardes de configuration, la migration ou les mises à jour de configuration à grande échelle.

API de configuration

Surveillance web

Configurez des scénarios web pour surveiller vos applications et services web.

API des scénarios web

Alertes L'API Zabbix vous permet de gérer les actions et alertes de votre système de surveillance.

Actions et alertes

Définissez des actions et des opérations pour notifier les utilisateurs de certains événements ou exécuter automatiquement des commandes à distance. Accédez aux informations sur les alertes générées et leurs destinataires.

API Action | API Alert

Types de médias

Configurez les types de médias et les différentes manières dont les utilisateurs recevront des alertes.

API de type de média

Scripts

Configurez et exécutez des scripts pour vous aider dans vos tâches quotidiennes.

API des scripts

Utilisateurs L'API Zabbix vous permet de gérer les utilisateurs de votre système de surveillance.

Utilisateurs et groupes d'utilisateurs

Ajoutez des utilisateurs qui auront accès à Zabbix, assignez-les à des groupes d'utilisateurs et accordez des permissions. Créez des rôles pour une gestion granulaire des droits des utilisateurs.

API utilisateur | API des groupes d'utilisateurs | API du répertoire d'utilisateurs | API des rôles utilisateur

Jetons d'API

Gérez les jetons d'autorisation.

API des jetons

Authentification

Modifier les options de configuration de l'authentification.

API d'authentification

Appareils

Gérez les appareils mobiles à utiliser avec l'application mobile Zabbix.

API des appareils

Administration Avec l'API Zabbix, vous pouvez modifier les paramètres d'administration de votre système de supervision.

Général

Modifier certaines options de configuration globales.

API d'auto-enregistrement | API de mappage d'icônes | API d'image | API des paramètres | API d'expression régulière | API de module | API de connecteur

Journal d'audit

Suivez les modifications de configuration effectuées par chaque utilisateur.

API du journal d'audit

Gestion interne

Configurez la gestion interne.

API de gestion interne

Proxys et groupes de proxys

Gérez les proxys utilisés dans votre configuration de supervision distribuée.

API Proxy | API Groupe de proxys

Macros

Gérer les macros.

API de macro utilisateur

Informations Zabbix Récupérer la version de l'API Zabbix ou l'heure de la dernière mise à jour depuis le serveur Zabbix.

API info API | Server info API

Action

Cette classe est conçue pour fonctionner avec les actions.

Références d'objet :

- Action
- Opération d'action
  - Message d'opération d'action
  - Condition d'opération d'action
- Opération de récupération d'action
- Opération de mise à jour d'action
- Filtre d'action

- Condition de filtre d'action

Méthodes disponibles :

- action.create - créer de nouvelles actions
- action.delete - supprimer des actions
- action.get - récupérer des actions
- action.update - mettre à jour des actions

Objet Action

Les objets suivants sont directement liés à l'API `action`.

Action

L'objet action possède les propriétés suivantes.

Property	Type	Description
actionid	ID	ID de l'action. Comportement de la propriété : - *en lecture seule* - *obligatoire* pour les opérations de mise à jour
esc_period	string	Durée par défaut d'une étape d'opération. Doit être d'au moins 60 secondes. Accepte un nombre de secondes (par exemple, 60, 300), une valeur avec un suffixe temporel (par exemple, 30s, 1m, 2h) ou une macro utilisateur. Comportement de la propriété : - *prise en charge* si `eventsource` est défini sur « événement créé par un déclencheur », « événement interne » ou « événement créé lors de la mise à jour de l'état d'un service »
eventsource	integer	Type d'événements que l'action traitera. Consultez la propriété `source` de l'événement pour obtenir la liste des types d'événements pris en charge. Comportement de la propriété : - *constante* - *obligatoire* pour les opérations de création
name	string	Nom de l'action. Comportement de la propriété : - *obligatoire* pour les opérations de création
status	integer	Indique si l'action est activée ou désactivée. Valeurs possibles : 0 - *(par défaut)* activée ; 1 - désactivée.
pause_symptoms	integer	Indique s'il faut suspendre l'escalade si l'événement est un événement symptôme. Valeurs possibles : 0 - Ne pas suspendre l'escalade pour les problèmes symptômes ; 1 - *(par défaut)* Suspendre l'escalade pour les problèmes symptômes. Comportement de la propriété : - *prise en charge* si `eventsource` est défini sur « événement créé par un déclencheur »

Property	Type	Description
pause_suppressed	integer	Indique s'il faut suspendre l'escalade pendant les périodes de maintenance. Valeurs possibles : 0 - Ne pas suspendre l'escalade ; 1 - *(par défaut)* Suspendre l'escalade. Comportement de la propriété : - *prise en charge* si `eventsource` est défini sur « événement créé par un déclencheur »
notify_if_canceled	integer	Indique s'il faut envoyer une notification lorsque l'escalade est annulée. Valeurs possibles : 0 - Ne pas envoyer de notification lorsque l'escalade est annulée ; 1 - *(par défaut)* Envoyer une notification lorsque l'escalade est annulée. Comportement de la propriété : - *prise en charge* si `eventsource` est défini sur « événement créé par un déclencheur »

Opération d'action

L'objet d'opération d'action définit une opération qui sera effectuée lorsqu'une action est exécutée. Il possède les propriétés suivantes.

Property	Type	Description
operationtype	integer	Type d'opération. Valeurs possibles : 0 - envoyer un message ; 1 - script global ; 2 - ajouter un hôte ; 3 - supprimer un hôte ; 4 - ajouter à un groupe d'hôtes ; 5 - supprimer d'un groupe d'hôtes ; 6 - lier un modèle ; 7 - délier un modèle ; 8 - activer un hôte ; 9 - désactiver un hôte ; 10 - définir le mode d'inventaire de l'hôte ; 13 - ajouter des étiquettes à l'hôte ; 14 - supprimer des étiquettes de l'hôte. Valeurs possibles si `eventsource` de l'objet Action est défini sur « événement créé par un déclencheur » ou « événement créé lors de la mise à jour de l'état d'un service » : 0 - « envoyer un message » ; 1 - « script global ». Valeurs possibles si `eventsource` de l'objet Action est défini sur « événement interne » : 0 - « envoyer un message ». Comportement de la propriété : - *required*

Property	Type	Description
esc_period	string	Durée d'une étape d'escalade en secondes. Doit être supérieure à 60 secondes. Accepte un nombre de secondes (par exemple, 60, 300), une valeur avec un suffixe temporel (par exemple, 30s, 1m, 2h) ou une macro utilisateur. Si la valeur est définie sur 0 ou 0s, la période d'escalade par défaut de l'action sera utilisée. Valeur par défaut : 0s. Comportement de la propriété : - *supported* si `eventsource` de l'objet Action est défini sur « événement créé par un déclencheur », « événement interne » ou « événement créé lors de la mise à jour de l'état d'un service »
esc_step_from	integer	Étape à partir de laquelle commencer l'escalade. Valeur par défaut : 1. Comportement de la propriété : - *supported* si `eventsource` de l'objet Action est défini sur « événement créé par un déclencheur », « événement interne » ou « événement créé lors de la mise à jour de l'état d'un service »
esc_step_to	integer	Étape à laquelle terminer l'escalade. Valeur par défaut : 1. Comportement de la propriété : - *supported* si `eventsource` de l'objet Action est défini sur « événement créé par un déclencheur », « événement interne » ou « événement créé lors de la mise à jour de l'état d'un service »
evaltype	integer	Méthode d'évaluation des conditions de l'opération. Valeurs possibles : 0 - *(par défaut)* And/Or ; 1 - And ; 2 - Or.
opcommand	object	Script global à exécuter. Le script global doit avoir la propriété `scriptid` définie. Comportement de la propriété : - *required* si `operationtype` est défini sur « script global »
opcommand_grp	array	Groupes d'hôtes sur lesquels exécuter les scripts globaux. Les groupes d'hôtes doivent avoir la propriété `groupid` définie. Comportement de la propriété : - *required* si `operationtype` est défini sur « script global » et que `opcommand_hst` n'est pas défini
opcommand_hst	array	Hôte sur lequel exécuter les scripts globaux. Les hôtes doivent avoir la propriété `hostid` définie. Comportement de la propriété : - *required* si `operationtype` est défini sur « script global » et que `opcommand_grp` n'est pas défini
opconditions	array	Conditions de l'opération utilisées pour les actions de déclencheur. L'objet de condition d'opération est décrit en détail ci-dessous.

Property	Type	Description
opgroup	array	Groupes d'hôtes auxquels ajouter les hôtes. Les groupes d'hôtes doivent avoir la propriété `groupid` définie. Comportement de la propriété : - *required* si `operationtype` est défini sur « ajouter à un groupe d'hôtes » ou « supprimer d'un groupe d'hôtes »
opmessage	object	Objet contenant les données du message envoyé par l'opération. L'objet de message d'opération est décrit en détail ci-dessous. Comportement de la propriété : - *required* si `operationtype` est défini sur « envoyer un message »
opmessage_grp	array	Groupes d'utilisateurs auxquels envoyer des messages. Les groupes d'utilisateurs doivent avoir la propriété `usrgrpid` définie. Comportement de la propriété : - *required* si `operationtype` est défini sur « envoyer un message » et que `opmessage_usr` n'est pas défini
opmessage_usr	array	Utilisateurs auxquels envoyer des messages. Les utilisateurs doivent avoir la propriété `userid` définie. Comportement de la propriété : - *required* si `operationtype` est défini sur « envoyer un message » et que `opmessage_grp` n'est pas défini
optemplate	array	Modèles à lier aux hôtes. Les modèles doivent avoir la propriété `templateid` définie. Comportement de la propriété : - *required* si `operationtype` est défini sur « lier un modèle » ou « délier un modèle »
opinventory	object	Mode d'inventaire à définir pour l'hôte. L'inventaire doit avoir la propriété `inventory_mode` définie. Comportement de la propriété : - *required* si `operationtype` est défini sur « définir le mode d'inventaire de l'hôte »
optag	array	Étiquettes de l'hôte à ajouter ou à supprimer. Les étiquettes doivent avoir la propriété `tag` définie. La propriété `value` est facultative. Comportement de la propriété : - *supported* si `operationtype` est défini sur « ajouter des étiquettes à l'hôte » ou « supprimer des étiquettes de l'hôte ».

Message d'opération d'action

L'objet de message d'opération contient des données sur le message qui sera envoyé par l'opération. Il possède les propriétés suivantes.

Propriété	Type	Description
default_msg	integer	Indique s'il faut utiliser le texte et le sujet par défaut du message d'action. Valeurs possibles : 0 - utiliser les données de l'opération ; 1 - *(par défaut)* utiliser les données du type de média.
mediatypeid	ID	ID du type de média qui sera utilisé pour envoyer le message. Comportement de la propriété : - *pris en charge* si `operationtype` de l'objet d'opération d'action, de l'objet d'opération de récupération d'action ou de l'objet d'opération de mise à jour d'action est défini sur "send message", ou si `operationtype` de l'objet d'opération de mise à jour d'action est défini sur "notify all involved"
message	string	Texte du message d'opération. Comportement de la propriété : - *pris en charge* si `default_msg` est défini sur "use the data from the operation"
subject	string	Sujet du message d'opération. Comportement de la propriété : - *pris en charge* si `default_msg` est défini sur "use the data from the operation"

Condition d'opération d'action

L'objet de condition d'opération d'action définit une condition qui doit être remplie pour exécuter l'opération actuelle. Il possède les propriétés suivantes.

Propriété	Type	Description
conditiontype	integer	Type de condition. Valeurs possibles : 14 - événement acquitté. Comportement de la propriété : - *obligatoire*
value	string	Valeur à comparer. Comportement de la propriété : - *obligatoire*
operator	integer	Opérateur de condition. Valeurs possibles : 0 - *(par défaut)* =

Les opérateurs et valeurs suivants sont pris en charge pour chaque type de condition d'opération.

Condition	Nom de la condition	Opérateurs pris en charge	Valeur attendue
14	Événement acquitté	=	Indique si l'événement est acquitté. Valeurs possibles : 0 - non acquitté ; 1 - acquitté.

Opération de récupération d'action

L'objet d'opération de récupération d'action définit une opération qui sera exécutée lorsqu'un problème est résolu. Les opérations de récupération sont possibles **uniquement** pour les actions de déclencheur, internes et de service. Il possède les propriétés suivantes.

Propriété	Type	Description
operationtype	integer	Type d'opération. Valeurs possibles si `eventsource` de l'objet Action est défini sur "événement créé par un déclencheur" ou "événement créé lors d'une mise à jour de l'état du service" : 0 - envoyer un message ; 1 - script global ; 11 - notifier toutes les parties impliquées. Valeurs possibles si `eventsource` de l'objet Action est défini sur "événement interne" : 0 - envoyer un message ; 11 - notifier toutes les parties impliquées. Comportement de la propriété : - *obligatoire*
opcommand	object	Script global à exécuter. Le script global doit avoir la propriété `scriptid` définie. Comportement de la propriété : - *obligatoire* si `operationtype` est défini sur "script global"
opcommand_grp	array	Groupes d'hôtes sur lesquels exécuter les scripts globaux. Les groupes d'hôtes doivent avoir la propriété `groupid` définie. Comportement de la propriété : - *obligatoire* si `eventsource` de l'objet Action est défini sur "événement créé par un déclencheur", que `operationtype` est défini sur "script global" et que `opcommand_hst` n'est pas défini
opcommand_hst	array	Hôte sur lequel exécuter les scripts globaux. Les hôtes doivent avoir la propriété `hostid` définie. Comportement de la propriété : - *obligatoire* si `eventsource` de l'objet Action est défini sur "événement créé par un déclencheur", que `operationtype` est défini sur "script global" et que `opcommand_grp` n'est pas défini
opmessage	object	Objet contenant les données du message envoyé par l'opération de récupération. L'objet de message d'opération est décrit en détail ci-dessus. Comportement de la propriété : - *obligatoire* si `operationtype` est défini sur "envoyer un message"
opmessage_grp	array	Groupes d'utilisateurs auxquels envoyer des messages. Les groupes d'utilisateurs doivent avoir la propriété `usrgrpid` définie. Comportement de la propriété : - *obligatoire* si `operationtype` est défini sur "envoyer un message" et que `opmessage_usr` n'est pas défini

Propriété	Type	Description
opmessage_usr	array	Utilisateurs auxquels envoyer des messages. Les utilisateurs doivent avoir la propriété `userid` définie. Comportement de la propriété : - *obligatoire* si `operationtype` est défini sur ”envoyer un message” et que `opmessage_grp` n’est pas défini

Opération de mise à jour d’action

L’objet d’opération de mise à jour d’action définit une opération qui sera effectuée lorsqu’un problème est mis à jour (ajout d’un commentaire, acquittement, modification de la gravité ou fermeture manuelle). Les opérations de mise à jour sont possibles **uniquement** pour les actions de déclencheur et de service. Il possède les propriétés suivantes.

Propriété	Type	Description
operationtype	integer	Type d’opération. Valeurs possibles : 0 - envoyer un message ; 1 - script global ; 12 - notifier toutes les parties impliquées. Comportement de la propriété : - *obligatoire*
opcommand	object	Script global à exécuter. Le script global doit avoir la propriété `scriptid` définie. Comportement de la propriété : - *obligatoire* si `operationtype` est défini sur ”script global”
opcommand_grp	array	Groupes d’hôtes sur lesquels exécuter des scripts globaux. Les groupes d’hôtes doivent avoir la propriété `groupid` définie. Comportement de la propriété : - *obligatoire* si `eventsource` de l’objet Action est défini sur ”événement créé par un déclencheur”, que `operationtype` est défini sur ”script global” et que `opcommand_hst` n’est pas défini
opcommand_hst	array	Hôte sur lequel exécuter des scripts globaux. Les hôtes doivent avoir la propriété `hostid` définie. Comportement de la propriété : - *obligatoire* si `eventsource` de l’objet Action est défini sur ”événement créé par un déclencheur”, que `operationtype` est défini sur ”script global” et que `opcommand_grp` n’est pas défini
opmessage	object	Objet contenant les données du message envoyé par l’opération de mise à jour. L’objet de message d’opération est décrit en détail ci-dessus.
opmessage_grp	array	Groupes d’utilisateurs auxquels envoyer des messages. Les groupes d’utilisateurs doivent avoir la propriété `usrgrpid` définie. Comportement de la propriété : - *obligatoire* si `operationtype` est défini sur ”envoyer un message” et que `opmessage_usr` n’est pas défini

Propriété	Type	Description
opmessage_usr	array	Utilisateurs auxquels envoyer des messages. Les utilisateurs doivent avoir la propriété `userid` définie. Comportement de la propriété : - *obligatoire* si `operationtype` est défini sur ”envoyer un message” et que `opmessage_grp` n’est pas défini

Filtre d’action

L’objet de filtre d’action définit un ensemble de conditions qui doivent être remplies pour exécuter les opérations d’action configurées. Il possède les propriétés suivantes.

Propriété	Type	Description
conditions	array	Ensemble de conditions de filtre à utiliser pour filtrer les résultats. Les conditions seront triées dans l’ordre de leur placement dans la formule. Comportement de la propriété: - *obligatoire*
evaltype	integer	Méthode d’évaluation des conditions du filtre. Valeurs possibles : 0 - Et/Ou ; 1 - Et ; 2 - Ou ; 3 - Expression personnalisée. Comportement de la propriété: - *obligatoire*
eval_formula	string	Expression générée qui sera utilisée pour évaluer les conditions du filtre. L’expression contient des ID qui font référence à des conditions de filtre spécifiques par leur `formulaid`. La valeur de `eval_formula` est égale à la valeur de `formula` pour les filtres avec une expression personnalisée. Comportement de la propriété: - *lecture seule*
formula	string	Expression définie par l’utilisateur à utiliser pour évaluer les conditions des filtres avec une expression personnalisée. L’expression doit contenir des ID qui font référence à des conditions de filtre spécifiques par leur `formulaid`. Les ID utilisés dans l’expression doivent correspondre exactement à ceux définis dans les conditions du filtre : aucune condition ne peut rester inutilisée ou être omise. Comportement de la propriété: - *obligatoire* si `evaltype` est défini sur ”expression personnalisée”

Condition de filtre d’action

L’objet de condition de filtre d’action définit une condition spécifique qui doit être vérifiée avant l’exécution des opérations de l’action.

Propriété	Type	Description
conditiontype	integer	Type de condition. Valeurs possibles si `eventsource` de l'objet Action est défini sur « événement créé par un déclencheur » : 0 - groupe de hôtes ; 1 - hôte ; 2 - déclencheur ; 3 - nom de l'événement ; 4 - gravité du déclencheur ; 6 - période ; 13 - modèle d'hôte ; 16 - problème supprimé ; 25 - balise d'événement ; 26 - valeur de balise d'événement. Valeurs possibles si `eventsource` de l'objet Action est défini sur « événement créé par une règle de découverte » : 7 - adresse IP de l'hôte ; 8 - type de service découvert ; 9 - port du service découvert ; 10 - état de la découverte ; 11 - durée de disponibilité ou d'indisponibilité ; 12 - valeur reçue ; 18 - règle de découverte ; 19 - vérification de découverte ; 20 - proxy ; 21 - objet découvert ; 29 - groupe de proxies. Valeurs possibles si `eventsource` de l'objet Action est défini sur « événement créé par l'enregistrement automatique d'un agent actif » : 20 - proxy ; 22 - nom de l'hôte ; 24 - métadonnées de l'hôte ; 29 - groupe de proxies. Valeurs possibles si `eventsource` de l'objet Action est défini sur « événement interne » : 0 - groupe d'hôtes ; 1 - hôte ; 13 - modèle d'hôte ; 23 - type d'événement ; 25 - balise d'événement ; 26 - valeur de balise d'événement. Valeurs possibles si `eventsource` de l'objet Action est défini sur « événement créé lors de la mise à jour de l'état d'un service » : 25 - balise d'événement ; 26 - valeur de balise d'événement ; 27 - service ; 28 - nom du service. Comportement de la propriété : - *obligatoire*
value	string	Valeur à comparer. Comportement de la propriété : - *obligatoire*

Propriété	Type	Description
value2	string	Valeur secondaire à comparer. Comportement de la propriété : - *obligatoire* si `eventsource` de l'objet Action est défini sur « événement créé par un déclencheur », que `conditiontype` est défini sur une valeur possible pour les actions de déclencheur et que le type de condition (voir ci-dessous) est « 26 » - *obligatoire* si `eventsource` de l'objet Action est défini sur « événement interne », que `conditiontype` est défini sur une valeur possible pour les actions internes et que le type de condition (voir ci-dessous) est « 26 » - *obligatoire* si `eventsource` de l'objet Action est défini sur « événement créé lors de la mise à jour de l'état d'un service », que `conditiontype` est défini sur une valeur possible pour les actions de service et que le type de condition (voir ci-dessous) est « 26 »
formulaid	string	ID unique arbitraire utilisé pour référencer la condition depuis une expression personnalisée. Ne peut contenir que des lettres majuscules. L'ID doit être défini par l'utilisateur lors de la modification des conditions de filtre, mais sera généré à nouveau lors de leur demande ultérieure.
operator	integer	Opérateur de la condition. Valeurs possibles : 0 - *(par défaut)* est égal à ; 1 - n'est pas égal à ; 2 - contient ; 3 - ne contient pas ; 4 - est dans ; 5 - est supérieur ou égal à ; 6 - est inférieur ou égal à ; 7 - n'est pas dans ; 8 - correspond à ; 9 - ne correspond pas à ; 10 - Oui ; 11 - Non.

Note:
Pour mieux comprendre comment utiliser les filtres avec différents types d'expressions, consultez les exemples des pages des méthodes action.get et action.create.

Les opérateurs et valeurs suivants sont pris en charge pour chaque type de condition.

Condition	Nom de la condition	Opérateurs pris en charge	Valeur attendue
0	Groupe d'hôtes	est égal à, n'est pas égal à	ID du groupe d'hôtes.
1	Hôte	est égal à, n'est pas égal à	ID de l'hôte.
2	Déclencheur	est égal à, n'est pas égal à	ID du déclencheur.
3	Nom de l'événement	contient, ne contient pas	Nom de l'événement.
4	Gravité du déclencheur	est égal à, n'est pas égal à, est supérieur ou égal à, est inférieur ou égal à	Gravité du déclencheur. Consultez la propriété `severity` du déclencheur pour obtenir la liste des gravités de déclencheur prises en charge.

Condition	Nom de la condition	Opérateurs pris en charge	Valeur attendue
5	Valeur du déclencheur	est égal à	Valeur du déclencheur. Consultez la propriété `value` du déclencheur pour obtenir la liste des valeurs de déclencheur prises en charge.
6	Période	est dans, n'est pas dans	Moment où l'événement a été déclenché, sous la forme d'une période.
7	Adresse IP de l'hôte	est égal à, n'est pas égal à	Une ou plusieurs plages d'adresses IP à vérifier, séparées par des virgules. Consultez la section configuration de la découverte réseau pour plus d'informations sur les formats de plages d'adresses IP pris en charge.
8	Type de service découvert	est égal à, n'est pas égal à	Type du service découvert. Le type de service correspond au type de la vérification de découverte utilisée pour détecter le service. Consultez la propriété `type` de la vérification de découverte pour obtenir la liste des types pris en charge.
9	Port du service découvert	est égal à, n'est pas égal à	Une ou plusieurs plages de ports, séparées par des virgules.
10	État de la découverte	est égal à	État d'un objet découvert. Valeurs possibles : 0 - hôte ou service disponible ; 1 - hôte ou service indisponible ; 2 - hôte ou service découvert ; 3 - hôte ou service perdu.
11	Durée de disponibilité ou d'indisponibilité	est supérieur ou égal à, est inférieur ou égal à	Durée, en secondes, pendant laquelle l'objet découvert se trouve dans son état actuel.
12	Valeurs reçues	est égal à, n'est pas égal à, est supérieur ou égal à, est inférieur ou égal à, contient, ne contient pas	Valeur renvoyée lors de l'exécution d'une vérification de découverte Zabbix agent, SNMPv1, SNMPv2 ou SNMPv3.
13	Modèle d'hôte	est égal à, n'est pas égal à	ID du modèle lié.
16	Problème supprimé	Oui, Non	Aucune valeur requise : l'utilisation de l'opérateur « Oui » signifie que le problème doit être supprimé, « Non » qu'il ne doit pas l'être.
18	Règle de découverte	est égal à, n'est pas égal à	ID de la règle de découverte.
19	Vérification de découverte	est égal à, n'est pas égal à	ID de la vérification de découverte.
20	Proxy	est égal à, n'est pas égal à	ID du proxy.
21	Objet découvert	est égal à	Type d'objet ayant déclenché l'événement de découverte. Valeurs possibles : 1 - hôte découvert ; 2 - service découvert.
22	Nom de l'hôte	contient, ne contient pas, correspond à, ne correspond pas à	Nom de l'hôte. L'utilisation d'une expression régulière est prise en charge pour les opérateurs *correspond à* et *ne correspond pas à* dans les conditions d'enregistrement automatique.

Condition	Nom de la condition	Opérateurs pris en charge	Valeur attendue
23	Type d'événement	est égal à	Événement interne spécifique. Valeurs possibles : 0 - élément dans l'état « non pris en charge » ; 1 - élément dans l'état « normal » ; 2 - règle LLD dans l'état « non pris en charge » ; 3 - règle LLD dans l'état « normal » ; 4 - déclencheur dans l'état « inconnu » ; 5 - déclencheur dans l'état « normal ».
24	Métadonnées de l'hôte	contient, ne contient pas, correspond à, ne correspond pas à	Métadonnées de l'hôte enregistré automatiquement. L'utilisation d'une expression régulière est prise en charge pour les opérateurs *correspond à* et *ne correspond pas à*.
25	Balise	est égal à, n'est pas égal à, contient, ne contient pas	Balise d'événement.
26	Valeur de balise	est égal à, n'est pas égal à, contient, ne contient pas	Valeur de la balise d'événement.
27	Service	est égal à, n'est pas égal à	ID du service.
28	Nom du service	est égal à, n'est pas égal à	Nom du service.
29	Groupe de proxies	est égal à, n'est pas égal à	ID du groupe de proxies.

action.create

Description

`object action.create(object/array actions)`

Cette méthode vous permet de créer de nouvelles actions.

Note:
Cette méthode est uniquement disponible pour les types d'utilisateur *Admin* et *Super admin*. Les autorisations d'appel de cette méthode peuvent être révoquées dans les paramètres des rôles utilisateur. Consultez Rôles utilisateur pour plus d'informations.

Paramètres

`(object/array)` Actions à créer.

En plus des propriétés d'action standard, la méthode accepte les paramètres suivants.

Paramètre	Type	Description
filter	object	Objet filtre d'action pour l'action.
operations	array	Opérations d'action à créer pour l'action.
recovery_operations	array	Opérations de récupération d'action à créer pour l'action.
update_operations	array	Opérations de mise à jour d'action à créer pour l'action.

Valeurs retournées

`(object)` Retourne un objet contenant les identifiants des actions créées sous la propriété `actionids`. L'ordre des ID retournés correspond à l'ordre des actions transmises.

Exemples

Créer une action de déclenchement

Créez une action de déclencheur qui commencera une fois qu'un déclencheur (avec le mot "mémoire" dans son nom) de l'hôte "10084" passera à l'état PROBLÈME. L'action aura 4 opérations configurées. La première opération immédiate enverra un message à tous les utilisateurs du groupe d'utilisateurs "7" via le type de média "1". Si l'événement n'est pas résolu en 30 minutes, la deuxième opération s'exécutera script "5" (script avec la portée "Action operation") sur tous les hôtes du groupe "2". Si l'événement est résolu, une opération de récupération informera tous les utilisateurs qui ont reçu des messages concernant le problème. Si l'événement est mis à jour, une opération d'accusé de réception/de mise à jour notifiera (avec un objet et un message personnalisés) tous les utilisateurs qui ont reçu des messages concernant le problème.

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "action.create",
    "params": {
        "name": "Trigger action",
        "eventsource": 0,
        "esc_period": "30m",
        "filter": {
            "evaltype": 0,
            "conditions": [
                {
                    "conditiontype": 1,
                    "operator": 0,
                    "value": "10084"
                },
                {
                    "conditiontype": 3,
                    "operator": 2,
                    "value": "memory"
                }
            ]
        },
        "operations": [
            {
                "operationtype": 0,
                "esc_step_from": 1,
                "esc_step_to": 1,
                "opmessage_grp": [
                    {
                        "usrgrpid": "7"
                    }
                ],
                "opmessage": {
                    "default_msg": 1,
                    "mediatypeid": "1"
                }
            },
            {
                "operationtype": 1,
                "esc_step_from": 2,
                "esc_step_to": 2,
                "opconditions": [
                    {
                        "conditiontype": 14,
                        "operator": 0,
                        "value": "0"
                    }
                ],
                "opcommand_grp": [
                    {
                        "groupid": "2"
                    }
                ],
                "opcommand": {
```

```
                    "scriptid": "5"
                }
            }
        ],
        "recovery_operations": [
            {
                "operationtype": "11",
                "opmessage": {
                    "default_msg": 1
                }
            }
        ],
        "update_operations": [
            {
                "operationtype": "12",
                "opmessage": {
                    "default_msg": 0,
                    "message": "Custom update operation message body",
                    "subject": "Custom update operation message subject"
                }
            }
        ]
    },
    "auth": "038e1d7b1735c6a5436ee9eae095879e",
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "actionids": [
            "17"
        ]
    },
    "id": 1
}
```

Créer une action de découverte

Créer une action qui liera les hôtes découverts au modèle ”10001”.

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "action.create",
    "params": {
        "name": "Discovery action",
        "eventsource": 1,
        "filter": {
            "evaltype": 0,
            "conditions": [
                {
                    "conditiontype": 21,
                    "operator": 0,
                    "value": "1"
                },
                {
                    "conditiontype": 10,
                    "operator": 0,
                    "value": "2"
                }
            ]
```

```
        },
        "operations": [
            {
                "operationtype": 6,
                "optemplate": [
                    {
                        "templateid": "10001"
                    }
                ]
            }
        ]
    },
    "auth": "038e1d7b1735c6a5436ee9eae095879e",
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "actionids": [
            "18"
        ]
    },
    "id": 1
}
```

Utilisation d'un filtre d'expression personnalisé

Créez une action de déclencheur qui utilise une expression personnalisée - « A and (B or C) » - pour évaluer les conditions de l'action. Lorsqu'un déclencheur avec une sévérité supérieure ou égale à « Warning » provenant de l'hôte « 10084 » ou de l'hôte « 10106 » passe à l'état PROBLEM, l'action enverra un message à tous les utilisateurs du groupe d'utilisateurs « 7 » via le type de média « 1 ». Les identifiants de formule « A », « B » et « C » ont été choisis arbitrairement.

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "action.create",
    "params": {
        "name": "Trigger action",
        "eventsource": 0,
        "esc_period": "15m",
        "filter": {
            "evaltype": 3,
            "formula": "A and (B or C)",
            "conditions": [
                {
                    "conditiontype": 4,
                    "operator": 5,
                    "value": "2",
                    "formulaid": "A"
                },
                {
                    "conditiontype": 1,
                    "operator": 0,
                    "value": "10084",
                    "formulaid": "B"
                },
                {
                    "conditiontype": 1,
                    "operator": 0,
                    "value": "10106",
                    "formulaid": "C"
```

```
                }
            ]
        },
        "operations": [
            {
                "operationtype": 0,
                "esc_step_from": 1,
                "esc_step_to": 1,
                "opmessage_grp": [
                    {
                        "usrgrpid": "7"
                    }
                ],
                "opmessage": {
                    "default_msg": 1,
                    "mediatypeid": "1"
                }
            }
        ]
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "actionids": [
            "18"
        ]
    },
    "id": 1
}
```

Créer une règle d'enregistrement automatique des agents

Ajoutez un hôte au groupe d'hôtes "Linux servers" lorsque le nom d'hôte contient "SRV" ou que les métadonnées contiennent "AlmaLinux".

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "action.create",
    "params": {
        "name": "Register Linux servers",
        "eventsource": "2",
        "status": "0",
        "filter": {
            "evaltype": "2",
            "conditions": [
                {
                    "conditiontype": "22",
                    "operator": "2",
                    "value": "SRV"
                },
                {
                    "conditiontype": "24",
                    "operator": "2",
                    "value": "AlmaLinux"
                }
            ]
        },
        "operations": [
```

```
            {
                "operationtype": "4",
                "opgroup": [
                    {
                        "groupid": "2"
                    }
                ]
            }
        ]
    },
    "auth": "038e1d7b1735c6a5436ee9eae095879e",
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "actionids": [
            19
        ]
    },
    "id": 1
}
```

Créer une règle d'autoréenregistrement d'agent avec des tags d'hôte

Créez une action d'autoréenregistrement qui ajoute un hôte au groupe d'hôtes « 2 » et ajoute deux tags d'hôte.

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "action.create",
    "params": {
        "name": "Register Linux servers with tags",
        "eventsource": "2",
        "operations": [
            {
                "operationtype": "4",
                "opgroup": [
                    {
                        "groupid": "2"
                    }
                ]
            },
            {
                "operationtype": "13",
                "optag": [
                    {
                        "tag": "location",
                        "value": "office"
                    },
                    {
                        "tag": "city",
                        "value": "Riga"
                    }
                ]
            }
        ]
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "actionids": [
            20
        ]
    },
    "id": 1
}
```

Voir aussi

- Filtre d'action
- Opération d'action
- Script

Source

CAction::create() dans *ui/include/classes/api/services/CAction.php*.

action.delete

Description

`object action.delete(array actionIds)`

Cette méthode vous permet de supprimer des actions.

> **Note:**
> Cette méthode est uniquement disponible pour les types d'utilisateur *Admin* et *Super admin*. Les autorisations permettant d'appeler cette méthode peuvent être révoquées dans les paramètres des rôles utilisateur. Consultez la page Rôles utilisateur pour plus d'informations.
 L'utilisateur ne pourra pas supprimer une action qui contient un proxy ou un groupe de proxys inaccessible dans ses conditions de filtrage. Consultez la page configuration des groupes d'utilisateurs (*Liste d'accès aux proxys*) pour plus d'informations.

Paramètres

`(array)` IDs des actions à supprimer.

Valeurs de retour

`(object)` Renvoie un objet contenant les IDs des actions supprimées sous la propriété `actionids`.

Exemples

Suppression de plusieurs actions

Supprimer deux actions.

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "action.delete",
    "params": [
        "17",
        "18"
    ],
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "actionids": [
            "17",
            "18"
```

```
        ]
    },
    "id": 1
}
```

Source

CAction::delete() dans *ui/include/classes/api/services/CAction.php*.

action.get

Description

`integer/array action.get(object parameters)`

La méthode permet de récupérer des actions selon les paramètres fournis.

> **Note:**
> Cette méthode est disponible pour les utilisateurs de tout type. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres des rôles utilisateur. Consultez Rôles utilisateur pour plus d'informations.

Paramètres

`(object)` Paramètres définissant la sortie souhaitée.

La méthode prend en charge les paramètres suivants.

Paramètre	Type	Description
actionids	ID/array	Retourner uniquement les actions avec les ID donnés.
groupids	ID/array	Retourner uniquement les actions qui utilisent les groupes d'hôtes donnés dans les conditions d'action.
hostids	ID/array	Retourner uniquement les actions qui utilisent les hôtes donnés dans les conditions d'action.
triggerids	ID/array	Retourner uniquement les actions qui utilisent les déclencheurs donnés dans les conditions d'action.
mediatypeids	ID/array	Retourner uniquement les actions qui utilisent les types de média donnés pour envoyer des messages.
usrgrpids	ID/array	Retourner uniquement les actions configurées pour envoyer des messages aux groupes d'utilisateurs donnés.
userids	ID/array	Retourner uniquement les actions configurées pour envoyer des messages aux utilisateurs donnés.
scriptids	ID/array	Retourner uniquement les actions configurées pour exécuter les scripts donnés.
selectFilter	query	Retourner une propriété `filter` avec le filtre de condition d'action.
selectOperations	query	Retourner une propriété `operations` avec les opérations d'action.
selectRecoveryOperations	query	Retourner une propriété `recovery_operations` avec les opérations de récupération d'action.
selectUpdateOperations	query	Retourner une propriété `update_operations` avec les opérations de mise à jour d'action.
sortfield	string/array	Trier le résultat selon les propriétés données. Valeurs possibles : `actionid`, `name`, `status`.
countOutput	boolean	Ces paramètres sont décrits dans le commentaire de référence.
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Valeurs retournées

`(integer/array)` Retourne soit :

- un tableau d'objets ;
- le nombre d'objets récupérés, si le paramètre `countOutput` a été utilisé.

Exemples

Récupérer les actions de déclencheur

Récupérez toutes les actions de déclencheur configurées ainsi que les conditions et opérations d'action.

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "action.get",
    "params": {
        "output": "extend",
        "selectOperations": "extend",
        "selectRecoveryOperations": "extend",
        "selectUpdateOperations": "extend",
        "selectFilter": "extend",
        "filter": {
            "eventsource": 0
        }
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "actionid": "3",
            "name": "Signaler les problèmes aux administrateurs Zabbix",
            "eventsource": "0",
            "status": "1",
            "esc_period": "1h",
            "pause_suppressed": "1",
            "filter": {
                "evaltype": "0",
                "formula": "",
                "conditions": [],
                "eval_formula": ""
            },
            "operations": [
                {
                    "operationid": "3",
                    "actionid": "3",
                    "operationtype": "0",
                    "esc_period": "0",
                    "esc_step_from": "1",
                    "esc_step_to": "1",
                    "evaltype": "0",
                    "opconditions": [],
                    "opmessage": [
                        {
                            "default_msg": "1",
                            "subject": "",
                            "message": "",
                            "mediatypeid" => "0"
                        }
                    ],
```

```
                "opmessage_grp": [
                    {
                        "usrgrpid": "7"
                    }
                ]
            }
        ],
        "recovery_operations": [
            {
                "operationid": "7",
                "actionid": "3",
                "operationtype": "11",
                "evaltype": "0",
                "opconditions": [],
                "opmessage": {
                    "default_msg": "0",
                    "subject": "{TRIGGER.STATUS}: {TRIGGER.NAME}",
                    "message": "Déclencheur : {TRIGGER.NAME}\r\nÉtat du déclencheur : {TRIGGER.STATUS}
                    "mediatypeid": "0"
                }
            }
        ],
        "update_operations": [
            {
                "operationid": "31",
                "operationtype": "12",
                "evaltype": "0",
                "opmessage": {
                    "default_msg": "1",
                    "subject": "",
                    "message": "",
                    "mediatypeid": "0"
                }
            },
            {
                "operationid": "32",
                "operationtype": "0",
                "evaltype": "0",
                "opmessage": {
                    "default_msg": "0",
                    "subject": "Mis à jour : {TRIGGER.NAME}",
                    "message": "{USER.FULLNAME} a mis à jour le problème le {EVENT.UPDATE.DATE} à {EVE
                    "mediatypeid": "1"
                },
                "opmessage_grp": [
                    {
                        "usrgrpid": "7"
                    }
                ],
                "opmessage_usr": []
            },
            {
                "operationid": "33",
                "operationtype": "1",
                "evaltype": "0",
                "opcommand": {
                    "scriptid": "3"
                },
                "opcommand_hst": [
                    {
                        "hostid": "10084"
                    }
```

```
                    ],
                    "opcommand_grp": []
                }
            ]
        }
    ],
    "id": 1
}
```

Récupération des actions de découverte

Récupérer toutes les actions de découverte configurées avec les conditions d'action et les opérations. Le filtre utilise le type d'évaluation "and", la propriété `formula` est donc vide et la propriété `eval_formula` est générée automatiquement.

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "action.get",
    "params": {
        "output": "extend",
        "selectOperations": "extend",
        "selectFilter": "extend",
        "filter": {
            "eventsource": 1
        }
    },
    "auth": "038e1d7b1735c6a5436ee9eae095879e",
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "actionid": "2",
            "name": "Auto discovery. Linux servers.",
            "eventsource": "1",
            "status": "1",
            "esc_period": "0s",
            "pause_suppressed": "1",
            "filter": {
                "evaltype": "0",
                "formula": "",
                "conditions": [
                    {
                        "conditiontype": "10",
                        "operator": "0",
                        "value": "0",
                        "value2": "",
                        "formulaid": "B"
                    },
                    {
                        "conditiontype": "8",
                        "operator": "0",
                        "value": "9",
                        "value2": "",
                        "formulaid": "C"
                    },
                    {
                        "conditiontype": "12",
                        "operator": "2",
                        "value": "Linux",
```

```
                        "value2": "",
                        "formulaid": "A"
                    }
                ],
                "eval_formula": "A and B and C"
            },
            "operations": [
                {
                    "operationid": "1",
                    "actionid": "2",
                    "operationtype": "6",
                    "esc_period": "0s",
                    "esc_step_from": "1",
                    "esc_step_to": "1",
                    "evaltype": "0",
                    "opconditions": [],
                    "optemplate": [
                        {
                            "templateid": "10001"
                        }
                    ]
                },
                {
                    "operationid": "2",
                    "actionid": "2",
                    "operationtype": "4",
                    "esc_period": "0s",
                    "esc_step_from": "1",
                    "esc_step_to": "1",
                    "evaltype": "0",
                    "opconditions": [],
                    "opgroup": [
                        {
                            "groupid": "2"
                        }
                    ]
                }
            ]
        }
    ],
    "id": 1
}
```


Voir aussi

- Filtre d'action
- Opération d'action

Source

CAction::get() dans *ui/include/classes/api/services/CAction.php*.

action.update

Description

`object action.update(object/array actions)`

Cette méthode vous permet de mettre à jour des actions existantes.

> **Note:**
> Cette méthode est disponible uniquement pour les types d'utilisateur *Admin* et *Super admin*. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres du rôle utilisateur. Consultez Rôles utilisateur pour plus d'informations.

Paramètres

`(object/array)` Propriétés d'action à mettre à jour.

La propriété `actionid` doit être définie pour chaque action, toutes les autres propriétés sont facultatives. Seules les propriétés transmises seront mises à jour, toutes les autres resteront inchangées.

En plus des propriétés d'action standard, la méthode accepte les paramètres suivants.

Paramètre	Type	Description
filter	object	Objet filtre d'action pour remplacer le filtre actuel.
operations	array	Opérations d'action pour remplacer les opérations existantes.
recovery_operations	array	Opérations de récupération d'action pour remplacer les opérations de récupération existantes. Comportement du paramètre : - *pris en charge* si `eventsource` de l'objet action est défini sur "événement créé par un déclencheur", "événement interne" ou "événement créé lors de la mise à jour de l'état du service"
update_operations	array	Opérations de mise à jour d'action pour remplacer les opérations de mise à jour existantes. Comportement du paramètre : - *pris en charge* si `eventsource` de l'objet action est défini sur "événement créé par un déclencheur" ou "événement créé lors de la mise à jour de l'état du service"

Valeurs retournées

`(object)` Retourne un objet contenant les identifiants des actions mises à jour sous la propriété `actionids`.

Exemples

Désactiver l'action

Désactive l'action, c'est-à-dire définir son statut sur "1".

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "action.update",
    "params": {
        "actionid": "2",
        "status": "1"
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "actionids": [
            "2"
        ]
    },
    "id": 1
}
```

Voir aussi

- Filtre d'action
- Opération d'action

Source

CAction::update() in *ui/include/classes/api/services/CAction.php*.

Alerte

Cette classe est conçue pour fonctionner avec des alertes.

Références d'objet :

- Alerte

Méthodes disponibles :

- alert.get - récupère les alertes

Objet d'alerte

Les objets suivants sont directement liés à l'API `alert`.

Alerte

> **Note:**
> Les alertes sont créées par le serveur Zabbix et ne peuvent pas être modifiées via l'API.

L'objet alerte contient des informations indiquant si certaines opérations d'action ont été exécutées avec succès. Il possède les propriétés suivantes.

Propriété	Type	Description
alertid	ID	ID de l'alerte.
actionid	ID	ID de l'action qui a généré l'alerte.
alerttype	integer	Type d'alerte. Valeurs possibles : 0 - message ; 1 - commande distante.
clock	timestamp	Heure à laquelle l'alerte a été générée.
error	string	Texte d'erreur en cas de problème lors de l'envoi d'un message ou de l'exécution d'une commande.
esc_step	integer	Étape d'escalade de l'action pendant laquelle l'alerte a été générée.
eventid	ID	ID de l'événement qui a déclenché l'action.
mediatypeid	ID	ID du type de média utilisé pour envoyer le message.
message	text	Texte du message. Comportement de la propriété : - *pris en charge* si `alerttype` est défini sur "message"
retries	integer	Nombre de fois que Zabbix a tenté d'envoyer le message.
sendto	string	Adresse, nom d'utilisateur ou autre identifiant du destinataire. Comportement de la propriété : - *pris en charge* si `alerttype` est défini sur "message"
status	integer	Statut indiquant si l'opération d'action a été exécutée avec succès. Valeurs possibles si `alerttype` est défini sur "message" : 0 - message non envoyé ; 1 - message envoyé ; 2 - échec après plusieurs tentatives ; 3 - la nouvelle alerte n'est pas encore traitée par le gestionnaire d'alertes. Valeurs possibles si `alerttype` est défini sur "remote command" : 0 - commande non exécutée ; 1 - commande exécutée ; 2 - tentative d'exécution de la commande sur le Zabbix agent, mais celui-ci n'était pas disponible.

Propriété	Type	Description
subject	string	Sujet du message. Comportement de la propriété : - *pris en charge* si `alerttype` est défini sur "message"
userid	ID	ID de l'utilisateur auquel le message a été envoyé.
p_eventid	ID	ID de l'événement de problème qui a généré l'alerte.
acknowledgeid	ID	ID de l'accusé de réception qui a généré l'alerte.

alert.get

Description

`integer/array alert.get(object parameters)`

La méthode permet de récupérer les alertes selon les paramètres fournis.

Note:
Cette méthode est disponible pour les utilisateurs de tout type. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres des rôles utilisateur. Consultez Rôles utilisateur pour plus d'informations.

Paramètres

`(object)` Paramètres définissant la sortie souhaitée.

La méthode prend en charge les paramètres suivants.

Paramètre	Type	Description
alertids	ID/array	Retourne uniquement les alertes avec les ID donnés.
actionids	ID/array	Retourne uniquement les alertes générées par les actions données.
eventids	ID/array	Retourne uniquement les alertes générées par les événements donnés.
groupids	ID/array	Retourne uniquement les alertes générées par des objets des groupes d'hôtes donnés.
hostids	ID/array	Retourne uniquement les alertes générées par des objets des hôtes donnés.
mediatypeids	ID/array	Retourne uniquement les alertes de message qui ont utilisé les types de média donnés.
objectids	ID/array	Retourne uniquement les alertes générées par les objets donnés
userids	ID/array	Retourne uniquement les alertes de message qui ont été envoyées aux utilisateurs donnés.
eventobject	integer	Retourne uniquement les alertes générées par des événements liés à des objets du type donné. Voir l'`object` d'événement pour la liste des types d'objet pris en charge. Par défaut : 0 - déclencheur.
eventsource	integer	Retourne uniquement les alertes générées par des événements du type donné. Voir la `source` d'événement pour la liste des types d'événement pris en charge. Par défaut : 0 - événements de déclencheur.
time_from	timestamp	Retourne uniquement les alertes qui ont été générées après l'heure donnée.
time_till	timestamp	Retourne uniquement les alertes qui ont été générées avant l'heure donnée.
selectHosts	query	Retourne une propriété `hosts` avec les données des hôtes qui ont déclenché l'opération d'action.

Paramètre	Type	Description
selectMediatypes	query	Retourne une propriété `mediatypes` avec un tableau des types de média qui ont été utilisés pour l'alerte de message. Voir `mediatype.get` pour les restrictions selon le type d'utilisateur.
selectUsers	query	Retourne une propriété `users` avec un tableau des utilisateurs auxquels le message était adressé. Voir `user.get` pour les restrictions selon le type d'utilisateur.
sortfield	string/array	Trie le résultat selon les propriétés données. Valeurs possibles : `alertid`, `clock`, `eventid`, `mediatypeid`, `sendto`, `status`.
countOutput	boolean	Ces paramètres sont décrits dans les commentaires de référence.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Valeurs retournées

`(integer/array)` Retourne soit :

- un tableau d'objets ;
- le nombre d'objets récupérés, si le paramètre countOutput' a été utilisé.

Exemples

Récupérer les alertes par ID d'action

Récupérez toutes les alertes générées par l'action « 3 ».

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "alert.get",
    "params": {
        "output": "extend",
        "actionids": "3"
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "alertid": "1",
            "actionid": "3",
            "eventid": "21243",
            "userid": "1",
            "clock": "1362128008",
            "mediatypeid": "1",
            "sendto": "support@company.com",
            "subject": "PROBLEM: Zabbix agent on Linux server is unreachable for 5 minutes: ",
            "message": "Trigger: Zabbix agent on Linux server is unreachable for 5 minutes: \nTrigger stat
```

```
            "status": "0",
            "retries": "3",
            "error": "",
            "esc_step": "1",
            "alerttype": "0",
            "p_eventid": "0",
            "acknowledgeid": "0"
        }
    ],
    "id": 1
}
```

Voir aussi

- Hôte
- Type de média
- Utilisateur

Source

CAlert::get() in *ui/include/classes/api/services/CAlert.php*.

API info

Cette classe est conçue pour récupérer les informations relatives à l'API.

Méthodes disponibles :

- apiinfo.version - récupère la version de l'API Zabbix.

apiinfo.version

Description

`string apiinfo.version(array)`

Cette méthode vous permet de récupérer la version de l'API Zabbix.

> **Attention:**
> Cette méthode est uniquement disponible pour les utilisateurs non authentifiés.

Paramètres

`(array)` La méthode accepte un tableau vide.

Valeurs retournées

`(string)` Renvoie la version de l'API Zabbix.

> **Note:**
> A partir de Zabbix 2.0.4, la version de l'API correspond à la version de Zabbix.

Exemples

Récupération de la version de l'API

Récupérez la version de l'API Zabbix.

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "apiinfo.version",
    "params": [],
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": "8.0.0",
    "id": 1
}
```

Source

CAPIInfo::version() dans *ui/include/classes/api/services/CAPIInfo.php*.

Appareil

Cette classe est conçue pour fonctionner avec des appareils mobiles utilisés avec l'application mobile Zabbix.

Références d'objet :

- Device

Méthodes disponibles :

- device.get - récupérer des appareils
- device.offboard - supprimer (retirer) des appareils

Objet Device

L'objet Device représente un appareil mobile à utiliser avec l'application mobile Zabbix.

Property	Type	Description
deviceid	ID	ID de l'appareil (interne, par exemple 1, 2). Comportement de la propriété : - Lecture seule
userid	ID	ID de l'utilisateur de l'appareil. Comportement de la propriété : - Constant
uuid	string	Identifiant universel unique de l'appareil (UUIDv7) (externe, par exemple `019f930e-0ea2-7c1f-9dcc-0fbb60c40664`). Comportement de la propriété : - Lecture seule
name	string	Nom de l'appareil. Comportement de la propriété : - Lecture seule
status	integer	État de l'appareil. Valeurs possibles : 0 - Nouveau 1 - Actif 2 - Non attribué Comportement de la propriété : - Lecture seule
activated_at	timestamp	Date et heure auxquelles l'appareil a été ajouté dans l'application mobile, ou 0 s'il n'a pas encore été ajouté. Comportement de la propriété : - Lecture seule

Property	Type	Description
lastaccess	timestamp	Date et heure auxquelles l'appareil a été authentifié pour la dernière fois dans l'application mobile, ou 0 s'il n'a pas encore été authentifié. Comportement de la propriété : - Lecture seule

device.get

Cette méthode permet de récupérer les appareils mobiles à utiliser avec l'application mobile Zabbix.

La méthode est disponible pour les utilisateurs *Utilisateur*, *Administrateur* et *Super administrateur* si `EnableMobileDevices` est activé et si l'utilisateur dispose d'un accès aux appareils. L'accès aux méthodes de l'API peut être révoqué dans la configuration du rôle utilisateur.

Les utilisateurs *Utilisateur* et *Administrateur* peuvent uniquement récupérer leurs propres appareils.

Signature de la méthode : `integer/array device.get(object parameters)`

Source de la méthode : `CDevice::get()` dans `ui/include/classes/api/services/CDevice.php`

Paramètres

Parameter	Type	Description
deviceids	ID/array	Retourne uniquement les appareils avec les ID d'appareil indiqués.
userids	ID/array	Retourne uniquement les appareils appartenant aux utilisateurs avec les ID utilisateur indiqués.
roleids	ID/array	Retourne uniquement les appareils appartenant aux utilisateurs avec les ID de rôle utilisateur indiqués.
usrgrpids	ID/array	Retourne uniquement les appareils appartenant aux utilisateurs avec les ID de groupe d'utilisateurs indiqués.
sortfield	string/array	Trie le résultat selon les propriétés indiquées. Valeurs possibles : `deviceid`, `name`.
countOutput	boolean	Ces paramètres sont décrits dans le commentaire de référence.
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Valeurs de retour

Renvoie soit un tableau d'objets, soit le nombre d'objets récupérés (si `countOutput` est utilisé).

Exemples

Récupérer des appareils

Récupérer les appareils appartenant aux utilisateurs dont les ID sont les suivants : 1, 2, 3.
 Notez que l'utilisateur 2 (invité) ne possède aucun appareil et que l'appareil 3 n'a pas encore été ajouté ou authentifié dans l'application mobile Zabbix.

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "device.get",
    "params": {
        "output": "extend",
        "userids": [1, 2, 3]
```

```
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "deviceid": "1",
            "userid": "1",
            "uuid": "019f930e-0ea2-7c1f-9dcc-0fbb60c40664",
            "name": "Google Pixel 10",
            "status": "1",
            "activated_at": "1784533122",
            "lastaccess": "1784879933"
        },
        {
            "deviceid": "2",
            "userid": "1",
            "uuid": "019d00fb-9e87-7ad5-bb4e-978467254aff",
            "name": "Apple iPhone 15 Pro",
            "status": "1",
            "activated_at": "1784533225",
            "lastaccess": "0"
        },
        {
            "deviceid": "3",
            "userid": "3",
            "uuid": "019d00fb-9e87-7ad5-bb4e-6ed88a2297ee",
            "name": "",
            "status": "0",
            "activated_at": "0",
            "lastaccess": "0"
        }
    ],
    "id": 1
}
```

device.offboard

Cette méthode permet de supprimer (désenregistrer) des appareils mobiles utilisés avec l'application mobile Zabbix.

La méthode est disponible pour les utilisateurs de type *Utilisateur*, *Administrateur* et *Super administrateur* si `EnableMobileDevices` est activé et si l'utilisateur a accès aux appareils. L'accès aux méthodes de l'API peut être révoqué dans la configuration du rôle utilisateur.

Les utilisateurs de type *Utilisateur* et *Administrateur* peuvent uniquement supprimer leurs propres appareils.

Signature de la méthode : `object device.offboard(object parameters)`

Source de la méthode : `CDevice::offboard()` dans `ui/include/classes/api/services/CDevice.php`

Paramètres

Parameter	Type	Description
uuid	string	Identifiant universel unique du périphérique (UUIDv7). Correspond à `deviceid`.
force	boolean	Définissez `true` pour ignorer l'erreur « Périphérique introuvable » et supprimer de force le périphérique. Valeur par défaut : `false`

Valeurs de retour

Renvoie un objet contenant l'UUID du périphérique supprimé.

Exemples

Supprimer un appareil

Supprime un appareil avec l'UUID indiqué, en ignorant l'erreur « Appareil introuvable ».

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "device.offboard",
    "params": {
        "uuid": "019d0106-5eef-7154-a159-bb29e35d38e5",
        "force": true
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "uuid": "019d0106-5eef-7154-a159-bb29e35d38e5"
    },
    "id": 1
}
```

Authentification

Cette classe est conçue pour fonctionner avec les paramètres d'authentification.

Références d'objet :

- Authentification

Méthodes disponibles :

- authentication.get - récupère les paramètres d'authentification
- authentication.update - met à jour les paramètres d'authentification

Objet d'authentification

Les objets suivants sont directement liés à l'API « d'authentification ».

Authentification

L'objet d'authentification possède les propriétés suivantes.

Property	Type	Description
authentication_type	integer	Authentification par défaut. Valeurs possibles : 0 - *(par défaut)* Interne ; 1 - LDAP.
http_auth_enabled	integer	Authentification HTTP. Valeurs possibles : 0 - *(par défaut)* Désactivée ; 1 - Activée. Comportement de la propriété : - *prise en charge* si `$ZBX_FEATURE_FLAGS['http_auth_enabled']` est défini sur `true` dans le fichier de configuration de l'interface.

Property	Type	Description
http_login_form	integer	Formulaire de connexion par défaut. Valeurs possibles : 0 - *(par défaut)* Formulaire de connexion Zabbix ; 1 - Formulaire de connexion HTTP. Comportement de la propriété : - *prise en charge* si `$ZBX_FEATURE_FLAGS['http_auth_enabled']` est défini sur `true` dans le fichier de configuration de l'interface.
http_strip_domains	string	Nom de domaine à supprimer. Comportement de la propriété : - *prise en charge* si `$ZBX_FEATURE_FLAGS['http_auth_enabled']` est défini sur `true` dans le fichier de configuration de l'interface.
http_case_sensitive	integer	Connexion HTTP sensible à la casse. Valeurs possibles : 0 - Désactivée ; 1 - *(par défaut)* Activée. Comportement de la propriété : - *prise en charge* si `$ZBX_FEATURE_FLAGS['http_auth_enabled']` est défini sur `true` dans le fichier de configuration de l'interface.
ldap_auth_enabled	integer	Authentification LDAP. Valeurs possibles : 0 - *(par défaut)* Désactivée ; 1 - Activée.
ldap_case_sensitive	integer	Connexion LDAP sensible à la casse. Valeurs possibles : 0 - Désactivée ; 1 - *(par défaut)* Activée.
ldap_userdirectoryid	ID	ID de l'annuaire utilisateur par défaut pour l'authentification LDAP. Utilisé pour les groupes d'utilisateurs dont `gui_access` est défini sur LDAP ou Système par défaut. Comportement de la propriété : - *requis* si `ldap_auth_enabled` est défini sur « Activée »
saml_auth_enabled	integer	Authentification SAML. Valeurs possibles : 0 - *(par défaut)* Désactivée ; 1 - Activée.
saml_case_sensitive	integer	Connexion SAML sensible à la casse. Valeurs possibles : 0 - Désactivée ; 1 - *(par défaut)* Activée.
passwd_min_length	integer	Exigence de longueur minimale du mot de passe. Les valeurs possibles vont de 1 à 70. Valeur par défaut : 8.

Property	Type	Description
passwd_check_rules	integer	Règles de vérification du mot de passe. Valeurs bitmap possibles : 0 - Vérifier la longueur du mot de passe ; 1 - Vérifier si le mot de passe utilise des lettres latines majuscules et minuscules ; 2 - Vérifier si le mot de passe utilise des chiffres ; 4 - Vérifier si le mot de passe utilise des caractères spéciaux ; 8 - *(par défaut)* Vérifier que le mot de passe ne figure pas dans la liste des mots de passe couramment utilisés et qu'il ne contient pas de dérivés du mot « Zabbix », ni du prénom, du nom ou du nom d'utilisateur de l'utilisateur. Il s'agit d'un champ de masque de bits ; toute somme de valeurs bitmap possibles est acceptée (par exemple, 15 pour vérifier toutes les règles).
ldap_jit_status	integer	État du provisionnement LDAP. Valeurs possibles : 0 - Désactivé pour les IdP LDAP configurés ; 1 - Activé pour les IdP LDAP configurés.
saml_jit_status	integer	État du provisionnement SAML. Valeurs possibles : 0 - Désactivé pour les IdP SAML configurés ; 1 - Activé pour les IdP SAML configurés.
jit_provision_interval	string	Intervalle de temps entre les demandes de provisionnement JIT pour l'utilisateur connecté. Accepte un nombre de secondes (par exemple, 3600) ou une valeur avec un suffixe temporel (par exemple, 3600s, 60m, 1h, 1d, 1M, 1y), avec la prise en charge des mois et des années. Valeur minimale : 1h. Valeur par défaut : 1h.
disabled_usrgrpid	ID	Disponible uniquement pour le provisionnement LDAP. ID du groupe d'utilisateurs auquel affecter l'utilisateur déprovisionné. Le groupe d'utilisateurs doit être désactivé et ne peut pas être activé ou supprimé lorsqu'il est configuré. Comportement de la propriété : - *requis* si `ldap_jit_status` est défini sur « Activé pour les IdP LDAP configurés » ou si `saml_jit_status` est défini sur « Activé pour les IdP SAML configurés »
mfa_status	integer	Authentification multifacteur. Valeurs possibles : 0 - Désactivée (pour toutes les méthodes MFA configurées) ; 1 - Activée (pour toutes les méthodes MFA configurées).
mfaid	ID	Méthode MFA par défaut pour les groupes d'utilisateurs avec l'authentification MFA activée. Comportement de la propriété : - *requis* si `mfa_status` est défini sur « Activée »

authentication.get

Description

`object authentication.get(object parameters)`

Cette méthode permet de récupérer un objet d'authentification selon les paramètres fournis.

Note:
Cette méthode est disponible uniquement pour le type d'utilisateur *Super admin*. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres du rôle utilisateur. Consultez User roles pour plus d'informations.

Paramètres

`(object)` Paramètres définissant la sortie souhaitée.

La méthode ne prend en charge qu'un seul paramètre.

Paramètre	Type	Description
output	query	Ce paramètre est décrit dans le commentaire de référence.

Valeurs retournées

`(object)` Renvoie l'objet d'authentification.

Exemples

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "authentication.get",
    "params": {
        "output": "extend"
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "authentication_type": "0",
        "http_auth_enabled": "0",
        "http_login_form": "0",
        "http_strip_domains": "",
        "http_case_sensitive": "1",
        "ldap_auth_enabled": "0",
        "ldap_case_sensitive": "1",
        "ldap_userdirectoryid": "0",
        "saml_auth_enabled": "0",
        "saml_case_sensitive": "0",
        "passwd_min_length": "8",
        "passwd_check_rules": "15",
        "jit_provision_interval": "1h",
        "saml_jit_status": "0",
        "ldap_jit_status": "0",
        "disabled_usrgrpid": "9",
        "mfa_status": "0",
        "mfaid": "0"
    },
    "id": 1
}
```

Source

CAuthentication::get() dans *ui/include/classes/api/services/CAuthentication.php*.

authentication.update

Description

`object authentication.update(object authentication)`

Cette méthode vous permet de mettre à jour des paramètres d'authentification existants.

> **Note:**
> Cette méthode est disponible uniquement pour le type d'utilisateur *Super admin*. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres des rôles utilisateur. Consultez User roles pour plus d'informations.

Paramètres

`(object)` Propriétés d'authentification à mettre à jour.

Valeurs retournées

`(array)` Retourne un tableau avec les noms des paramètres mis à jour.

Exemples

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "authentification.update",
    "params": {
        "http_auth_enabled": 1,
        "http_case_sensitive": 0,
        "http_login_form": 1
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": [
        "http_auth_enabled",
        "http_case_sensitive",
        "http_login_form"
    ],
    "id": 1
}
```

Source

CAuthentication::update() dans *ui/include/classes/api/services/CAuthentication.php*.

Carte

Cette classe est conçue pour fonctionner avec des cartes.

Références d'objet :

- Carte
- Élément de carte
  - Élément de carte Hôte
  - Élément de carte Groupe d'hôtes
  - Élément de carte Carte
  - Élément de carte Déclencheur
  - Élément de carte Tag
  - Élément de carte URL
- Lien de carte
  - Déclencheur de lien de carte
  - Indicateurs de lien de carte
- URL de carte
- Utilisateur de carte
- Groupe d'utilisateurs de carte
- Formes de carte
- Lignes de carte

Méthodes disponibles :

- map.create - créer de nouvelles cartes
- map.delete - supprimer des cartes
- map.get - récupérer des cartes
- map.update - mettre à jour des cartes

Objet de carte

Les objets suivants sont directement liés à l'API `map`.

Carte

L'objet carte possède les propriétés suivantes.

Propriété	Type	Description
sysmapid	ID	ID de la carte. Comportement de la propriété: - *lecture seule* - *obligatoire* pour les opérations de mise à jour
height	integer	Hauteur de la carte en pixels. Comportement de la propriété: - *obligatoire* pour les opérations de création
name	string	Nom de la carte. Comportement de la propriété: - *obligatoire* pour les opérations de création
width	integer	Largeur de la carte en pixels. Comportement de la propriété: - *obligatoire* pour les opérations de création
backgroundid	ID	ID de l'image utilisée comme arrière-plan de la carte.
background_scale	integer	Indique s'il faut activer la mise à l'échelle proportionnelle de l'image d'arrière-plan. Valeurs possibles : 0 - désactiver la mise à l'échelle ; 1 - *(par défaut)* activer la mise à l'échelle de l'image.
expand_macros	integer	Indique s'il faut développer les macros dans les libellés lors de la configuration de la carte. Valeurs possibles : 0 - *(par défaut)* ne pas développer les macros ; 1 - développer les macros.
expandproblem	integer	Indique si le déclencheur de problème sera affiché pour les éléments ayant un seul problème. Valeurs possibles : 0 - toujours afficher le nombre de problèmes ; 1 - *(par défaut)* afficher le déclencheur de problème s'il n'y a qu'un seul problème.
grid_align	integer	Indique s'il faut activer l'alignement sur la grille. Valeurs possibles : 0 - désactiver l'alignement sur la grille ; 1 - *(par défaut)* activer l'alignement sur la grille.
grid_show	integer	Indique s'il faut afficher la grille sur la carte. Valeurs possibles : 0 - ne pas afficher la grille ; 1 - *(par défaut)* afficher la grille.

Propriété	Type	Description
grid_size	integer	Taille de la grille de la carte en pixels. Valeurs prises en charge : 20, 40, 50, 75 et 100. Par défaut : 50.
highlight	integer	Indique si la mise en surbrillance des icônes est activée. Valeurs possibles : 0 - mise en surbrillance désactivée ; 1 - *(par défaut)* mise en surbrillance activée.
iconmapid	ID	ID de la carte d'icônes utilisée sur la carte.
label_format	integer	Indique s'il faut activer les libellés avancés. Valeurs possibles : 0 - *(par défaut)* désactiver les libellés avancés ; 1 - activer les libellés avancés.
label_location	integer	Emplacement du libellé de l'élément de la carte. Valeurs possibles : 0 - *(par défaut)* en bas ; 1 - à gauche ; 2 - à droite ; 3 - en haut.
label_string_host	string	Libellé personnalisé pour les éléments hôte. Comportement de la propriété: - *obligatoire* si `label_type_host` est défini sur "custom"
label_string_hostgroup	string	Libellé personnalisé pour les éléments de groupe d'hôtes. Comportement de la propriété: - *obligatoire* si `label_type_hostgroup` est défini sur "custom"
label_string_image	string	Libellé personnalisé pour les éléments image. Comportement de la propriété: - *obligatoire* si `label_type_image` est défini sur "custom"
label_string_map	string	Libellé personnalisé pour les éléments carte. Comportement de la propriété: - *obligatoire* si `label_type_map` est défini sur "custom"
label_string_trigger	string	Libellé personnalisé pour les éléments déclencheur. Comportement de la propriété: - *obligatoire* si `label_type_trigger` est défini sur "custom"
label_type	integer	Type de libellé de l'élément de la carte. Valeurs possibles : 0 - libellé ; 1 - adresse IP ; 2 - *(par défaut)* nom de l'élément ; 3 - statut uniquement ; 4 - rien.
label_type_host	integer	Type de libellé pour les éléments hôte. Valeurs possibles : 0 - libellé ; 1 - adresse IP ; 2 - *(par défaut)* nom de l'élément ; 3 - statut uniquement ; 4 - rien ; 5 - personnalisé.

Propriété	Type	Description
label_type_hostgroup	integer	Type de libellé pour les éléments de groupe d'hôtes. Valeurs possibles : 0 - libellé ; 2 - *(par défaut)* nom de l'élément ; 3 - statut uniquement ; 4 - rien ; 5 - personnalisé.
label_type_image	integer	Type de libellé pour les éléments de groupe d'hôtes. Valeurs possibles : 0 - libellé ; 2 - *(par défaut)* nom de l'élément ; 4 - rien ; 5 - personnalisé.
label_type_map	integer	Type de libellé pour les éléments carte. Valeurs possibles : 0 - libellé ; 2 - *(par défaut)* nom de l'élément ; 3 - statut uniquement ; 4 - rien ; 5 - personnalisé.
label_type_trigger	integer	Type de libellé pour les éléments déclencheur. Valeurs possibles : 0 - libellé ; 2 - *(par défaut)* nom de l'élément ; 3 - statut uniquement ; 4 - rien ; 5 - personnalisé.
markelements	integer	Indique s'il faut mettre en surbrillance les éléments de la carte dont le statut a récemment changé. Valeurs possibles : 0 - *(par défaut)* ne pas mettre les éléments en surbrillance ; 1 - mettre les éléments en surbrillance.
severity_min	integer	Gravité minimale des déclencheurs qui seront affichés sur la carte. Consultez la propriété `severity` du déclencheur pour obtenir la liste des gravités de déclencheur prises en charge.
show_element_label	integer	Comment afficher les libellés des éléments par défaut. Valeurs possibles : 0 - toujours afficher ; 1 - *(par défaut)* masquer automatiquement.
show_link_label	integer	Comment afficher les libellés des liens par défaut. Valeurs possibles : 0 - toujours afficher ; 1 - *(par défaut)* masquer automatiquement.
show_unack	integer	Comment les problèmes doivent être affichés. Valeurs possibles : 0 - *(par défaut)* afficher le nombre de tous les problèmes ; 1 - afficher uniquement le nombre de problèmes non acquittés ; 2 - afficher séparément le nombre de problèmes acquittés et non acquittés.
userid	ID	ID de l'utilisateur propriétaire de la carte.

Propriété	Type	Description
private	integer	Type de partage de la carte. Valeurs possibles : 0 - carte publique ; 1 - *(par défaut)* carte privée.
show_suppressed	integer	Indique si les problèmes supprimés sont affichés. Valeurs possibles : 0 - *(par défaut)* masquer les problèmes supprimés ; 1 - afficher les problèmes supprimés.

Élément de carte

L'objet d'élément de carte définit un objet affiché sur une carte. Il possède les propriétés suivantes.

Propriété	Type	Description
selementid	ID	ID de l'élément de carte. Comportement de la propriété: - *lecture seule*
elements	array	Objet de données Element. Comportement de la propriété: - *obligatoire* si `elementtype` est défini sur "host", "map", "trigger" ou "host group"
elementtype	integer	Type d'élément de carte. Valeurs possibles : 0 - hôte ; 1 - carte ; 2 - déclencheur ; 3 - groupe d'hôtes ; 4 - image. Comportement de la propriété: - *obligatoire*
iconid_off	ID	ID de l'image utilisée pour afficher l'élément dans son état par défaut. Comportement de la propriété: - *obligatoire*
areatype	integer	Comment les hôtes d'un groupe d'hôtes séparé doivent être affichés. Valeurs possibles : 0 - *(par défaut)* l'élément de groupe d'hôtes occupera toute la carte ; 1 - l'élément de groupe d'hôtes aura une taille fixe.
elementsubtype	integer	Comment un élément de groupe d'hôtes doit être affiché sur une carte. Valeurs possibles : 0 - *(par défaut)* afficher le groupe d'hôtes comme un seul élément ; 1 - afficher séparément chaque hôte du groupe.
evaltype	integer	Condition de filtrage des tags de l'élément de carte, méthode d'évaluation. Valeurs possibles : 0 - *(par défaut)* And/Or ; 2 - Or.
height	integer	Hauteur, en pixels, de l'élément de groupe d'hôtes à taille fixe. Par défaut : 200.

Propriété	Type	Description
iconid_disabled	ID	ID de l'image utilisée pour afficher les éléments de carte désactivés. Comportement de la propriété: - *pris en charge* si `elementtype` est défini sur "host", "map", "trigger" ou "host group"
iconid_maintenance	ID	ID de l'image utilisée pour afficher les éléments de carte en maintenance. Comportement de la propriété: - *pris en charge* si `elementtype` est défini sur "host", "map", "trigger" ou "host group"
iconid_on	ID	ID de l'image utilisée pour afficher les éléments de carte présentant des problèmes. Comportement de la propriété: - *pris en charge* si `elementtype` est défini sur "host", "map", "trigger" ou "host group"
label	string	Libellé de l'élément.
label_location	integer	Emplacement du libellé de l'élément de carte. Valeurs possibles : -1 - *(par défaut)* emplacement par défaut ; 0 - bas ; 1 - gauche ; 2 - droite ; 3 - haut.
permission	integer	Type de niveau d'autorisation. Valeurs possibles : -1 - aucun ; 2 - lecture seule ; 3 - lecture-écriture.
show_label	integer	Comment afficher le libellé de l'élément. Valeurs possibles : -1 - *(par défaut)* valeur par défaut de la carte ; 0 - toujours afficher ; 1 - masquer automatiquement.
sysmapid	ID	ID de la carte à laquelle l'élément appartient. Comportement de la propriété: - *lecture seule*
urls	array	URL de l'élément de carte. L'objet URL d'élément de carte est décrit en détail ci-dessous.
use_iconmap	integer	Indique si le mappage d'icônes doit être utilisé pour les éléments d'hôte. Valeurs possibles : 0 - ne pas utiliser le mappage d'icônes ; 1 - *(par défaut)* utiliser le mappage d'icônes.
viewtype	integer	Algorithme de placement de l'élément de groupe d'hôtes. Valeurs possibles : 0 - *(par défaut)* grille.
width	integer	Largeur, en pixels, de l'élément de groupe d'hôtes à taille fixe. Par défaut : 200.
x	integer	Coordonnées X de l'élément en pixels. Par défaut : 0.

Propriété	Type	Description
y	integer	Coordonnées Y de l'élément en pixels. Par défaut : 0.
zindex	integer	Valeur utilisée pour ordonner les éléments de carte (z-index). Par défaut : 0.

Élément de carte Hôte

L'objet d'élément de carte Hôte définit un élément d'hôte.

Propriété	Type	Description
hostid	ID	ID de l'hôte.

Élément de carte Groupe d'hôtes

L'objet de groupe d'hôtes d'un élément de carte définit un élément de groupe d'hôtes.

Property	Type	Description
groupid	ID	ID du groupe d'hôtes.

Élément de carte Map

L'objet d'élément de carte Map définit un élément de carte.

Propriété	Type	Description
sysmapid	ID	ID de la carte.

Élément de carte Déclencheur

L'objet de l'élément de carte Déclencheur définit un ou plusieurs éléments de déclencheur.

Propriété	Type	Description
triggerid	ID	ID du déclencheur.

Balise d'élément de carte

L'objet Balise d'élément de carte possède les propriétés suivantes.

Propriété	Type	Description
tag	string	Nom de la balise de l'élément de carte. Comportement de la propriété : - *obligatoire*
operator	integer	Opérateur de condition de la balise de l'élément de carte. Valeurs possibles : 0 - *(par défaut)* Contient ; 1 - Égal à ; 2 - Ne contient pas ; 3 - Différent de ; 4 - Existe ; 5 - N'existe pas.
value	string	Valeur de la balise de l'élément de carte.

URL de l'élément de carte

L’objet URL de l’élément de carte définit un lien cliquable qui sera disponible pour un élément de carte spécifique. Il possède les propriétés suivantes :

Propriété	Type	Description
sysmapelementurlid	ID	ID de l’URL de l’élément de carte. Comportement de la propriété : - *lecture seule*
name	string	Légende du lien. Comportement de la propriété : - *obligatoire*
url	string	URL du lien. Comportement de la propriété : - *obligatoire*
selementid	ID	ID de l’élément de carte auquel l’URL appartient.

Lien de carte

L’objet de lien de carte définit un lien entre deux éléments de carte. Il possède les propriétés suivantes.

Propriété	Type	Description
linkid	ID	ID du lien de carte. Comportement de la propriété : - *lecture seule*
sysmapid	ID	ID de la carte à laquelle le lien appartient.
selementid1	ID	ID du premier élément de carte lié à une extrémité. Comportement de la propriété : - *requis*
selementid2	ID	ID du premier élément de carte lié à l’autre extrémité. Comportement de la propriété : - *requis*
drawtype	integer	Style de tracé de la ligne du lien. Valeurs possibles : 0 - *(par défaut)* ligne ; 2 - ligne en gras ; 3 - ligne pointillée ; 4 - ligne en tirets.
color	string	Couleur de la ligne sous forme de code couleur hexadécimal. Par défaut : 000000.
label	string	Libellé du lien.
show_label	integer	Comment afficher le libellé du lien. Valeurs possibles : -1 - *(par défaut)* valeur par défaut de la carte ; 0 - toujours afficher ; 1 - masquer automatiquement.
indicator_type	integer	Sélectionnez le type d’indicateur du lien. Valeurs possibles : 0 - *(par défaut)* lien statique ; 1 - déclencheur ; 2 - valeur d’élément.

Propriété	Type	Description
linktriggers	array	Déclencheurs de lien de carte à utiliser comme indicateurs d'état du lien. L'objet déclencheur de lien de carte est décrit en détail ci-dessous. Comportement de la propriété : - *requis* si `indicator_type` est défini sur "trigger" (1).
itemid	ID	ID de l'élément. Comportement de la propriété : - *requis* si `indicator_type` est défini sur "item value" (2).
highlights	array	Mises en évidence de lien de carte à utiliser comme indicateurs d'état du lien. L'objet des indicateurs de lien de carte est décrit en détail ci-dessous. Comportement de la propriété : - *requis* si `indicator_type` est défini sur "item value" (2).
thresholds	array	Seuils de lien de carte à utiliser comme indicateurs d'état du lien. L'objet des indicateurs de lien de carte est décrit en détail ci-dessous. Comportement de la propriété : - *requis* si `indicator_type` est défini sur "item value" (2).
permission	integer	Type de niveau d'autorisation. Valeurs possibles : -1 - aucun ; 2 - lecture seule ; 3 - lecture-écriture.

Déclencheur de lien de carte

L'objet de déclencheur de lien de carte définit un indicateur d'état de lien de carte en fonction de l'état d'un déclencheur. Il possède les propriétés suivantes :

Propriété	Type	Description
triggerid	ID	ID du déclencheur utilisé comme indicateur de lien. Comportement de la propriété : - *obligatoire*
color	string	Couleur de l'indicateur sous forme de code couleur hexadécimal. Par défaut : `DD0000`.
drawtype	integer	Style de dessin de l'indicateur. Valeurs possibles : 0 - *(par défaut)* ligne ; 2 - ligne en gras ; 3 - ligne pointillée ; 4 - ligne en tirets.

Indicateurs de lien de carte

L'objet des indicateurs de lien de carte définit un indicateur d'état de lien de carte en fonction de la valeur de l'élément. Il possède les propriétés suivantes :

Propriété	Type	Description
pattern	string	Expression régulière à comparer. Disponible uniquement pour les surbrillances.
sortorder	integer	Utilisé pour définir l'ordre de tri des surbrillances. Disponible uniquement pour les surbrillances.
threshold	string	Valeur numérique à comparer. Disponible uniquement pour les seuils.
drawtype	integer	Style de dessin de l'indicateur. Valeurs possibles : 0 - *(par défaut)* ligne ; 2 - ligne en gras ; 3 - ligne pointillée ; 4 - ligne en tirets.
color	string	Couleur de l'indicateur sous forme de code couleur hexadécimal. Par défaut : DD0000. Comportement de la propriété : - *obligatoire*

URL de carte

L'objet URL de carte définit un lien cliquable qui sera disponible pour tous les éléments d'un type spécifique sur la carte. Il possède les propriétés suivantes :

Propriété	Type	Description
sysmapurlid	ID	ID de l'URL de carte. Comportement de la propriété : - *lecture seule*
name	string	Légende du lien. Comportement de la propriété : - *obligatoire*
url	string	URL du lien. Comportement de la propriété : - *obligatoire*
elementtype	integer	Type d'élément de carte pour lequel l'URL sera disponible. Consultez la propriété `type` de l'élément de carte pour obtenir la liste des types pris en charge. Par défaut : 0.
sysmapid	ID	ID de la carte à laquelle l'URL appartient.

Utilisateur de carte

Liste des autorisations de carte basées sur les utilisateurs. Elle possède les propriétés suivantes :

Propriété	Type	Description
sysmapuserid	ID	ID de l'utilisateur de la carte. Comportement de la propriété : - *lecture seule*
userid	ID	ID de l'utilisateur. Comportement de la propriété : - *obligatoire*

Propriété	Type	Description
permission	integer	Type de niveau d'autorisation. Valeurs possibles : 2 - lecture seule ; 3 - lecture-écriture. Comportement de la propriété : - *obligatoire*

Groupe d'utilisateurs de carte

Liste des permissions de carte basées sur les groupes d'utilisateurs. Il possède les propriétés suivantes :

Propriété	Type	Description
sysmapusrgrpid	ID	ID du groupe d'utilisateurs de carte. Comportement de la propriété : - *lecture seule*
usrgrpid	ID	ID du groupe d'utilisateurs. Comportement de la propriété : - *obligatoire*
permission	integer	Type de niveau de permission. Valeurs possibles : 2 - lecture seule ; 3 - lecture-écriture. Comportement de la propriété : - *obligatoire*

Formes de carte

L'objet de forme de carte définit une forme géométrique (avec ou sans texte) affichée sur une carte. Il possède les propriétés suivantes :

Propriété	Type	Description
sysmap_shapeid	ID	ID de l'élément de forme de carte. Comportement de la propriété : - *lecture seule*
type	integer	Type de l'élément de forme de carte. Valeurs possibles : 0 - rectangle ; 1 - ellipse. La propriété est requise lors de la création de nouvelles formes. Comportement de la propriété : - *obligatoire*
x	integer	Coordonnées X de la forme en pixels. Par défaut : 0.
y	integer	Coordonnées Y de la forme en pixels. Par défaut : 0.
width	integer	Largeur de la forme en pixels. Par défaut : 200.

Propriété	Type	Description
height	integer	Hauteur de la forme en pixels. Par défaut : 200.
text	string	Texte de la forme.
font	integer	Police du texte dans la forme. Valeurs possibles : 0 - Georgia, serif 1 - ”Palatino Linotype”, ”Book Antiqua”, Palatino, serif 2 - ”Times New Roman”, Times, serif 3 - Arial, Helvetica, sans-serif 4 - ”Arial Black”, Gadget, sans-serif 5 - ”Comic Sans MS”, cursive, sans-serif 6 - Impact, Charcoal, sans-serif 7 - ”Lucida Sans Unicode”, ”Lucida Grande”, sans-serif 8 - Tahoma, Geneva, sans-serif 9 - ”Trebuchet MS”, Helvetica, sans-serif 10 - Verdana, Geneva, sans-serif 11 - ”Courier New”, Courier, monospace 12 - ”Lucida Console”, Monaco, monospace Par défaut : 9.
font_size	integer	Taille de la police en pixels. Par défaut : 11.
font_color	string	Couleur de la police. Par défaut : 000000.
text_halign	integer	Alignement horizontal du texte. Valeurs possibles : 0 - centré ; 1 - gauche ; 2 - droite. Par défaut : 0.
text_valign	integer	Alignement vertical du texte. Valeurs possibles : 0 - milieu ; 1 - haut ; 2 - bas. Par défaut : 0.
border_type	integer	Type de bordure. Valeurs possibles : 0 - aucune ; 1 - ——————— ; 2 - ┅ ; 3 - – – –. Par défaut : 0.
border_width	integer	Largeur de la bordure en pixels. Par défaut : 0.
border_color	string	Couleur de la bordure. Par défaut : 000000.
background_color	string	Couleur d’arrière-plan (couleur de remplissage). Par défaut : `(vide)`.

Propriété	Type	Description
zindex	integer	Valeur utilisée pour ordonner toutes les formes et lignes (z-index). Par défaut : 0.

Lignes de carte

L'objet de ligne de carte définit une ligne affichée sur une carte. Il possède les propriétés suivantes :

Propriété	Type	Description
sysmap_shapeid	ID	ID de l'élément de forme de la carte. Comportement de la propriété: *- lecture seule*
x1	integer	Coordonnées X du point 1 de la ligne en pixels. Par défaut : 0.
y1	integer	Coordonnées Y du point 1 de la ligne en pixels. Par défaut : 0.
x2	integer	Coordonnées X du point 2 de la ligne en pixels. Par défaut : 200.
y2	integer	Coordonnées Y du point 2 de la ligne en pixels. Par défaut : 200.
line_type	integer	Type des lignes. Valeurs possibles : 0 - aucune ; 1 - ——————— ; 2 - ┅┅ ; 3 - − − −.
line_width	integer	Largeur des lignes en pixels. Par défaut : 0.
line_color	string	Couleur de la ligne. Par défaut : 000000.
zindex	integer	Valeur utilisée pour ordonner toutes les formes et lignes (z-index). Par défaut : 0.

map.create

Description

`object map.create(object/array maps)`

Cette méthode vous permet de créer de nouvelles cartes.

> **Note:**
> Cette méthode est disponible pour les utilisateurs de tout type. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres des rôles utilisateur. Voir Rôles utilisateur pour plus d'informations.

Paramètres

`(object/array)` Cartes à créer.

En plus des propriétés de carte standard, la méthode accepte les paramètres suivants.

Paramètre	Type	Description
links	array	Liens de carte à créer sur la carte.
selements	array	Éléments de carte à créer sur la carte.
urls	array	URL de carte à créer sur la carte.
users	array	Partages d'utilisateur de carte à créer sur la carte.
userGroups	array	Partages de groupe d'utilisateurs de carte à créer sur la carte.
shapes	array	Formes de carte à créer sur la carte.
lines	array	Lignes de carte à créer sur la carte.

Note:
Pour créer des liens de carte, vous devez définir `selementid` d'un élément de carte sur une valeur arbitraire, puis utiliser cette valeur pour référencer cet élément dans les propriétés de lien `selementid1` ou `selementid2`. Lorsque l'élément est créé, cette valeur sera remplacée par l'ID correct généré par Zabbix. Voir l'exemple.

Valeurs retournées

`(object)` Retourne un objet contenant les identifiants des cartes créées sous la propriété `sysmapids`. L'ordre des ID retournés correspond à l'ordre des cartes transmises.

Exemples

Créer une carte vide

Créez une carte sans éléments.

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "map.create",
    "params": {
        "name": "Map",
        "width": 600,
        "height": 600
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "sysmapids": [
            "8"
        ]
    },
    "id": 1
}
```

Créer une carte d'hôtes

Créez une carte avec deux éléments d'hôte et un lien entre eux. Notez l'utilisation des valeurs temporaires « selementid1 » et « selementid2 » dans l'objet de lien de carte pour faire référence aux éléments de la carte.

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "map.create",
    "params": {
        "name": "Host map",
        "width": 600,
        "height": 600,
        "selements": [
            {
```

```
                "selementid": "1",
                "elements": [
                    {"hostid": "1033"}
                ],
                "elementtype": 0,
                "iconid_off": "2"
            },

            {
                "selementid": "2",
                "elements": [
                    {"hostid": "1037"}
                ],
                "elementtype": 0,
                "iconid_off": "2"
            }
        ],
        "links": [
            {
                "selementid1": "1",
                "selementid2": "2"
            }
        ]
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "sysmapids": [
            "9"
        ]
    },
    "id": 1
}
```

Créer une carte de déclencheurs

Créez une carte avec un élément de déclencheur, qui contient deux déclencheurs.

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "map.create",
    "params": {
        "name": "Trigger map",
        "width": 600,
        "height": 600,
        "selements": [
            {
                "elements": [
                    {"triggerid": "12345"},
                    {"triggerid": "67890"}
                ],
                "elementtype": 2,
                "iconid_off": "2"
            }
        ]
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "sysmapids": [
            "10"
        ]
    },
    "id": 1
}
```

Partage de carte

Créez une carte avec deux types de partage (utilisateur et groupe d'utilisateurs).

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "map.create",
    "params": {
        "name": "Map sharing",
        "width": 600,
        "height": 600,
        "users": [
            {
                "userid": "4",
                "permission": "3"
            }
        ],
        "userGroups": [
            {
                "usrgrpid": "7",
                "permission": "2"
            }
        ]
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "sysmapids": [
            "9"
        ]
    },
    "id": 1
}
```

Formes de carte

Créez une carte avec le titre du nom de la carte.

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "map.create",
    "params": {
        "name": "Host map",
        "width": 600,
        "height": 600,
        "shapes": [
            {
```

```
                "type": 0,
                "x": 0,
                "y": 0,
                "width": 600,
                "height": 11,
                "text": "{MAP.NAME}"
            }
        ]
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "sysmapids": [
            "10"
        ]
    },
    "id": 1
}
```

Lignes de carte

Créez une ligne de carte.

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "map.create",
    "params": {
        "name": "Map API lines",
        "width": 500,
        "height": 500,
        "lines": [
            {
                "x1": 30,
                "y1": 10,
                "x2": 100,
                "y2": 50,
                "line_type": 1,
                "line_width": 10,
                "line_color": "009900"
            }
        ]
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "sysmapids": [
            "11"
        ]
    },
    "id": 1
}
```

Voir aussi

- Élément de carte

- Lien de carte
- URL de carte
- Utilisateur de carte
- Groupe d'utilisateurs de carte
- Forme de carte
- Ligne de carte

Source

CMap::create() dans *ui/include/classes/api/services/CMap.php*.

map.delete

Description

`object map.delete(array mapIds)`

Cette méthode vous permet de supprimer des cartes.

> **Note:**
> Cette méthode est disponible pour les utilisateurs de tout type. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres du rôle utilisateur. Consultez Rôles utilisateur pour plus d'informations.

Paramètres

`(array)` IDs des cartes à supprimer.

Valeurs retournées

`(object)` Retourne un objet contenant les identifiants des cartes supprimées sous la propriété `sysmapids`.

Exemples

Suppression de plusieurs cartes

Supprimer deux cartes.

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "map.delete",
    "params": [
        "12",
        "34"
    ],
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "sysmapids": [
            "12",
            "34"
        ]
    },
    "id": 1
}
```

Source

CMap::delete() dans *ui/include/classes/api/services/CMap.php*.

map.get

Description

`integer/array map.get(object parameters)`

La méthode permet de récupérer des cartes selon les paramètres fournis.

> **Note:**
> Cette méthode est disponible pour les utilisateurs de tout type. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres du rôle utilisateur. Consultez Rôles d'utilisateur pour plus d'informations.

Paramètres

`(object)` Paramètres définissant la sortie souhaitée.

La méthode prend en charge les paramètres suivants.

Paramètre	Type	Description
sysmapids	ID/array	Renvoie uniquement les cartes avec les ID donnés.
userids	ID/array	Renvoie uniquement les cartes qui appartiennent aux ID d'utilisateur donnés.
expandUrls	flag	Ajoute les URL globales de la carte aux éléments de carte correspondants et développe les macros dans toutes les URL des éléments de carte.
selectIconMap	query	Renvoie une propriété `iconmap` avec le plan d'icônes utilisé sur la carte.
selectLinks	query	Renvoie une propriété `links` avec les liens de la carte entre les éléments.
selectSelements	query	Renvoie une propriété `selements` avec les éléments de la carte.
selectUrls	query	Renvoie une propriété `urls` avec les URL de la carte.
selectUsers	query	Renvoie une propriété `users` avec les utilisateurs avec lesquels la carte est partagée.
selectUserGroups	query	Renvoie une propriété `userGroups` avec les groupes d'utilisateurs avec lesquels la carte est partagée.
selectShapes	query	Renvoie une propriété `shapes` avec les formes de la carte.
selectLines	query	Renvoie une propriété `lines` avec les lignes de la carte.
sortfield	string/array	Trie le résultat selon les propriétés données. Valeurs possibles : `name`, `width`, `height`.
countOutput	boolean	Ces paramètres sont décrits dans le commentaire de référence.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Valeurs retournées

`(integer/array)` Retourne soit :

- un tableau d'objets ;
- le nombre d'objets récupérés, si le paramètre `countOutput` a été utilisé.

Exemples

Récupérer une carte

Récupérez toutes les données concernant la carte « 3 ».

Requête :

```
{
    "jsonrpc": "2.0",
```

```
    "method": "map.get",
    "params": {
        "output": "extend",
        "selectSelements": "extend",
        "selectLinks": "extend",
        "selectUsers": "extend",
        "selectUserGroups": "extend",
        "selectShapes": "extend",
        "selectLines": "extend",
        "sysmapids": "3"
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "selements": [
                {
                    "selementid": "10",
                    "sysmapid": "3",
                    "elementtype": "4",
                    "evaltype": "0",
                    "iconid_off": "1",
                    "iconid_on": "0",
                    "label": "Zabbix server",
                    "label_location": "3",
                    "x": "11",
                    "y": "141",
                    "iconid_disabled": "0",
                    "iconid_maintenance": "0",
                    "elementsubtype": "0",
                    "areatype": "0",
                    "width": "200",
                    "height": "200",
                    "viewtype": "0",
                    "use_iconmap": "1",
                    "show_label": "-1",
                    "zindex":"0",
                    "elements": [],
                    "urls": [],
                    "tags": [
                        {
                            "tag": "service",
                            "value": "mysqld",
                            "operator": "0"
                        }
                    ]
                },
                {
                    "selementid": "11",
                    "sysmapid": "3",
                    "elementtype": "4",
                    "evaltype": "0",
                    "iconid_off": "1",
                    "iconid_on": "0",
                    "label": "Web server",
                    "label_location": "3",
                    "x": "211",
                    "y": "191",
```

```
                "iconid_disabled": "0",
                "iconid_maintenance": "0",
                "elementsubtype": "0",
                "areatype": "0",
                "width": "200",
                "height": "200",
                "viewtype": "0",
                "use_iconmap": "1",
                "show_label": "0",
                "zindex":"0",
                "elements": [],
                "urls": [],
                "tags": []
            },
            {
                "selementid": "12",
                "sysmapid": "3",
                "elementtype": "0",
                "evaltype": "0",
                "iconid_off": "185",
                "iconid_on": "0",
                "label": "{HOST.NAME}\r\n{HOST.CONN}",
                "label_location": "0",
                "x": "111",
                "y": "61",
                "iconid_disabled": "0",
                "iconid_maintenance": "0",
                "elementsubtype": "0",
                "areatype": "0",
                "width": "200",
                "height": "200",
                "viewtype": "0",
                "use_iconmap": "0",
                "show_label": "1",
                "zindex":"0",
                "elements": [
                    {
                        "hostid": "10084"
                    }
                ],
                "urls": [],
                "tags": []
            }
        ],
        "links": [
            {
                "linkid": "23",
                "sysmapid": "3",
                "selementid1": "10",
                "selementid2": "11",
                "drawtype": "0",
                "color": "00CC00",
                "label": "",
                "show_label": "1",
                "indicator_type": "0",
                "itemid": "0",
                "linktriggers": [],
                "thresholds": [],
                "highlights": []
            }
        ],
        "users": [
```

```
            {
                "sysmapuserid": "1",
                "userid": "2",
                "permission": "2"
            }
        ],
        "userGroups": [
            {
                "sysmapusrgrpid": "1",
                "usrgrpid": "7",
                "permission": "2"
            }
        ],
        "shapes":[
            {
                "sysmap_shapeid":"1",
                "type":"0",
                "x":"0",
                "y":"0",
                "width":"680",
                "height":"15",
                "text":"{MAP.NAME}",
                "font":"9",
                "font_size":"11",
                "font_color":"000000",
                "text_halign":"0",
                "text_valign":"0",
                "border_type":"0",
                "border_width":"0",
                "border_color":"000000",
                "background_color":"",
                "zindex":"0"
            }
        ],
        "lines":[
            {
                "sysmap_shapeid":"2",
                "x1": 30,
                "y1": 10,
                "x2": 100,
                "y2": 50,
                "line_type": 1,
                "line_width": 10,
                "line_color": "009900",
                "zindex":"1"
            }
        ],
        "sysmapid": "3",
        "name": "Local network",
        "width": "400",
        "height": "400",
        "backgroundid": "0",
        "background_scale": "1",
        "label_type": "2",
        "label_location": "3",
        "show_element_label": "0",
        "show_link_label": "1",
        "highlight": "1",
        "expandproblem": "1",
        "markelements": "0",
        "show_unack": "0",
        "grid_size": "50",
```

```
            "grid_show": "1",
            "grid_align": "1",
            "label_format": "0",
            "label_type_host": "2",
            "label_type_hostgroup": "2",
            "label_type_trigger": "2",
            "label_type_map": "2",
            "label_type_image": "2",
            "label_string_host": "",
            "label_string_hostgroup": "",
            "label_string_trigger": "",
            "label_string_map": "",
            "label_string_image": "",
            "iconmapid": "0",
            "expand_macros": "0",
            "severity_min": "0",
            "userid": "1",
            "private": "1",
            "show_suppressed": "1"
        }
    ],
    "id": 1
}
```


Voir aussi

- Carte d'icônes
- Élément de carte
- Lien de carte
- URL de carte
- Utilisateur de carte
- Groupe d'utilisateurs de carte
- Formes de carte
- Lignes de carte

Source

CMap::get() dans *ui/include/classes/api/services/CMap.php*.

map.update

Description

`object map.update(object/array maps)`

Cette méthode vous permet de mettre à jour des cartes existantes.

> **Note:**
> Cette méthode est disponible pour les utilisateurs de tout type. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres des rôles utilisateur. Consultez Rôles utilisateur pour plus d'informations.

Paramètres

`(object/array)` Associe les propriétés à mettre à jour.

La propriété `mapid` doit être définie pour chaque carte, toutes les autres propriétés sont facultatives. Seules les propriétés transmises seront mises à jour, toutes les autres resteront inchangées.

En plus des propriétés de carte standard, la méthode accepte les paramètres suivants.

Paramètre	Type	Description
links	array	Liens de carte pour remplacer les liens existants.
selements	array	Éléments de carte pour remplacer les éléments existants.
urls	array	URL de carte pour remplacer les URL existantes.
users	array	Partages d'utilisateur de carte pour remplacer les éléments existants.

Paramètre	Type	Description
userGroups	array	Partages de groupes d'utilisateurs de carte pour remplacer les éléments existants.
shapes	array	Formes de carte pour remplacer les formes existantes.
lines	array	Lignes de carte pour remplacer les lignes existantes.

Note:
Pour créer des liens de carte entre de nouveaux éléments de carte, vous devrez définir le `selementid` d'un élément sur une valeur arbitraire, puis utiliser cette valeur pour référencer cet élément dans les propriétés `selementid1` ou `selementid2` des liens. Lorsque l'élément est créé, cette valeur sera remplacée par l'ID correct généré par Zabbix. Voir l'exemple pour map.create.

Valeurs retournées

`(object)` Retourne un objet contenant les identifiants des cartes mises à jour sous la propriété `sysmapids`.

Exemples

Redimensionner une carte

Modifiez la taille de la carte à 1200x1200 pixels.

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "map.update",
    "params": {
        "sysmapid": "8",
        "width": 1200,
        "height": 1200
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "sysmapids": [
            "8"
        ]
    },
    "id": 1
}
```

Changer le propriétaire de la carte

Disponible uniquement pour les administrateurs et les super administrateurs.

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "map.update",
    "params": {
        "sysmapid": "9",
        "userid": "1"
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
```

```
    "result": {
        "sysmapids": [
            "9"
        ]
    },
    "id": 1
}
```

Voir aussi

- Élément de carte
- Lien de carte
- URL de carte
- Utilisateur de carte
- Groupe d'utilisateurs de carte
- Formes de carte
- Lignes de carte

Source

CMap::update() dans *ui/include/classes/api/services/CMap.php*.

Carte d'icônes

Cette classe est conçue pour fonctionner avec les cartes d'icônes.

Références d'objet :

- Carte d'icônes
- Correspondance d'icônes

Méthodes disponibles :

- iconmap.create - créer de nouvelles cartes d'icônes
- iconmap.delete - supprimer des cartes d'icônes
- iconmap.get - récupérer des cartes d'icônes
- iconmap.update - mettre à jour des cartes d'icônes

Objet de mappage d'icônes

Les objets suivants sont directement liés à l'API `iconmap`.

Mappage d'icônes

L'objet de mappage d'icônes possède les propriétés suivantes.

Propriété	Type	Description
iconmapid	ID	ID du mappage d'icônes. Comportement de la propriété : - *lecture seule* - *obligatoire* pour les opérations de mise à jour
default_iconid	ID	ID de l'icône par défaut. Comportement de la propriété : - *obligatoire* pour les opérations de création
name	string	Nom du mappage d'icônes. Comportement de la propriété : - *obligatoire* pour les opérations de création

Correspondance des icônes

L'objet de correspondance des icônes définit une icône spécifique à utiliser pour les hôtes ayant une certaine valeur de champ d'inventaire. Il possède les propriétés suivantes.

Propriété	Type	Description
iconid	ID	ID de l'icône utilisée par la correspondance des icônes. Comportement de la propriété : - *obligatoire*
expression	string	Expression utilisée pour faire correspondre le champ d'inventaire. Comportement de la propriété : - *obligatoire*
inventory_link	integer	ID du champ d'inventaire de l'hôte. Consultez l'objet d'inventaire de l'hôte pour obtenir la liste des champs d'inventaire pris en charge. Comportement de la propriété : - *obligatoire*
sortorder	integer	Position de la correspondance des icônes dans la carte d'icônes. Comportement de la propriété : - *lecture seule*

iconmap.create

Description

`object iconmap.create(object/array iconMaps)`

Cette méthode vous permet de créer de nouvelles cartes d'icônes.

Note:
Cette méthode est disponible uniquement pour le type d'utilisateur *Super admin*. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres des rôles utilisateur. Consultez User roles pour plus d'informations.

Paramètres

`(object/array)` Correspondances d'icônes à créer.

En plus des propriétés standard des correspondances d'icônes, la méthode accepte les paramètres suivants.

Paramètre	Type	Description
mappings	array	Correspondances d'icônes à créer pour la correspondance d'icônes. Comportement du paramètre: - *obligatoire*

Valeurs retournées

`(object)` Renvoie un objet contenant les ID des tables de correspondance d'icônes créées sous la propriété `iconmapids`. L'ordre des ID renvoyés correspond à l'ordre des tables de correspondance d'icônes transmises.

Exemples

Créer une carte d'icônes

Créez une carte d'icônes pour afficher des hôtes de différents types.

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "iconmap.create",
    "params": {
        "name": "Icônes de type",
        "default_iconid": "2",
```

```
        "mappings": [
            {
                "inventory_link": 1,
                "expression": "server",
                "iconid": "3"
            },
            {
                "inventory_link": 1,
                "expression": "switch",
                "iconid": "4"
            }
        ]
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "iconmapids": [
            "2"
        ]
    },
    "id": 1
}
```

Voir aussi

- Correspondance des icônes

Source

CIconMap::create() dans *ui/include/classes/api/services/CIconMap.php*.

iconmap.delete

Description

`object iconmap.delete(array iconMapIds)`

Cette méthode vous permet de supprimer des cartes d'icônes.

Note:
Cette méthode est disponible uniquement pour le type d'utilisateur *Super admin*. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres des rôles utilisateur. Consultez User roles pour plus d'informations.

Paramètres

`(array)` ID des plans d'icônes à supprimer.

Valeurs de retour

`(object)` Renvoie un objet contenant les identifiants des mappages d'icônes supprimés dans la propriété `iconmapids`.

Exemples

Supprimer plusieurs cartes d'icônes

Supprimez deux cartes d'icônes.

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "iconmap.delete",
    "params": [
        "2",
```

```
        "5"
    ],
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "iconmapids": [
            "2",
            "5"
        ]
    },
    "id": 1
}
```

Source

CIconMap::delete() dans *ui/include/classes/api/services/CIconMap.php*.

iconmap.get

Description

`integer/array iconmap.get(object parameters)`

La méthode permet de récupérer les mappages d'icônes selon les paramètres fournis.

> **Note:**
> Cette méthode est disponible pour les utilisateurs de tout type. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres des rôles utilisateur. Consultez Rôles utilisateur pour plus d'informations.

Paramètres

`(object)` Paramètres définissant la sortie souhaitée.

La méthode prend en charge les paramètres suivants.

Paramètre	Type	Description
iconmapids	ID/array	Retourner uniquement les plans d'icônes avec les ID indiqués.
sysmapids	ID/array	Retourner uniquement les plans d'icônes utilisés dans les cartes indiquées.
selectMappings	query	Retourner une propriété `mappings` avec les correspondances d'icônes utilisées.
sortfield	string/array	Trier le résultat selon les propriétés indiquées. Valeurs possibles : `iconmapid`, `name`.
countOutput	boolean	Ces paramètres sont décrits dans le commentaire de référence.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Valeurs retournées

`(integer/array)` Retourne soit :

- un tableau d'objets ;
- le nombre d'objets récupérés, si le paramètre `countOutput` a été utilisé.

Exemples

Récupérer une carte d'icônes

Récupérez toutes les données de la carte d'icônes « 3 ».

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "iconmap.get",
    "params": {
        "iconmapids": "3",
        "output": "extend",
        "selectMappings": "extend"
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "mappings": [
                {
                    "iconmappingid": "3",
                    "iconmapid": "3",
                    "iconid": "6",
                    "inventory_link": "1",
                    "expression": "server",
                    "sortorder": "0"
                },
                {
                    "iconmappingid": "4",
                    "iconmapid": "3",
                    "iconid": "10",
                    "inventory_link": "1",
                    "expression": "switch",
                    "sortorder": "1"
                }
            ],
            "iconmapid": "3",
            "name": "Host type icons",
            "default_iconid": "2"
        }
    ],
    "id": 1
}
```

Voir aussi

- Mappage des icônes

Source

CIconMap::get() dans *ui/include/classes/api/services/CIconMap.php*.

iconmap.update

Description

`object iconmap.update(object/array iconMaps)`

Cette méthode permet de mettre à jour des mappages d'icônes existants.

Note:
Cette méthode est disponible uniquement pour le type d'utilisateur *Super admin*. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres des rôles utilisateur. Consultez User roles pour plus d'informations.

Paramètres

`(object/array)` Propriétés des cartes d'icônes à mettre à jour.

La propriété `iconmapid` doit être définie pour chaque carte d'icônes, toutes les autres propriétés sont facultatives. Seules les propriétés transmises seront mises à jour, toutes les autres resteront inchangées.

En plus des propriétés standard des cartes d'icônes, la méthode accepte les paramètres suivants.

Paramètre	Type	Description
mappings	array	Correspondances d'icônes pour remplacer les correspondances d'icônes existantes.

Valeurs retournées

`(object)` Renvoie un objet contenant les ID des tables de correspondance d'icônes mises à jour sous la propriété `iconmapids`.

Exemples

Renommer une carte d'icônes

Renommez une carte d'icônes en « OS icons ».

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "iconmap.update",
    "params": {
        "iconmapid": "1",
        "name": "OS icons"
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "iconmapids": [
            "1"
        ]
    },
    "id": 1
}
```

Voir aussi

- Correspondance des icônes

Source

CIconMap::update() dans *ui/include/classes/api/services/CIconMap.php*.

Complex event processing

This class is designed to work with complex event processing.

Object references:

- CEP rule
- CEP rule filter
  - CEP rule filter condition

- CEP rule time window
- CEP rule operation
- CEP rule operation filter
- CEP rule operation filter condition

Available methods:

- ceprule.create - create new CEP rules
- ceprule.delete - delete CEP rules
- ceprule.get - retrieve CEP rules
- ceprule.update - update CEP rules
- ceprule.resettimewindows - reset time windows for CEP rule

CEP rule object

The following objects are directly related to the `ceprule` API.

CEP rule

The CEP rule object has the following properties.

Property	Type	Description
cep_ruleid	ID	ID of the CEP rule. Property behavior: - *read-only* - *required* for update operations
name	string	Name of the CEP rule. Property behavior: - *required* for create operations
window_type	integer	Time window type: 0 - (default) None 1 - Simple (multiple); 2 - Cause and symptom grouping; 3 - Simple (once); 4 - Event pattern match.
stop	integer	Possible values: 0 - (default) Continue with the next CEP rule according to sort order; 1 - Stop after executing this rule.
sortorder	integer	Execution order of the CEP rule.
description	string	Description of the CEP rule.
status	integer	Whether the CEP rule is enabled or disabled. Possible values: 0 - *(default)* enabled; 1 - disabled.
error	string	Error text if the CEP rule processing failed on rule level. Property behavior: - *read-only*

CEP rule filter

The CEP rule filter object defines a set of conditions that must be met for events to be processed by CEP rule. It has the following properties.

Property	Type	Description
conditions	array	Set of filter conditions to use for filtering results. The conditions will be sorted in the order of their placement in the formula. Property behavior: - *required*
evaltype	integer	Filter condition evaluation method. Possible values: 0 - And/Or; 1 - And; 2 - Or; 3 - Custom expression. Property behavior: - *required*
eval_formula	string	Generated expression that will be used for evaluating filter conditions. The expression contains IDs that reference specific filter conditions by its `formulaid`. The value of `eval_formula` is equal to the value of `formula` for filters with a custom expression. Property behavior: - *read-only*
formula	string	User-defined expression to be used for evaluating conditions of filters with a custom expression. The expression must contain IDs that reference specific filter conditions by its `formulaid`. The IDs used in the expression must exactly match the ones defined in the filter conditions: no condition can remain unused or omitted. Property behavior: - *required* if `evaltype` is set to "custom expression"

CEP rule filter condition

The CEP rule filter condition object defines a specific condition that must be checked before processing events by the CEP rule.

Property	Type	Description
type	integer	Type of condition. Possible values: 3 - event name; 4 - severity; 6 - time period; 25 - tag name; 26 - tag value; 36 - host visible name; 37 - host group name. Property behavior: - *required*

Property	Type	Description
operator	integer	Condition operator. Possible values: 0 - equals (default); 1 - does not equal; 2 - contains; 3 - does not contain; 4 - in; 5 - is greater than or equals; 6 - is less than or equals; 7 - not in. Property behavior: - *required*.
event_name	string	Event name for a specific operation type. Property behavior: - *required* if `type` is set to ”event name”
severity	string	Event severity for a specific operation type. Property behavior: - *required* if `type` is set to ”severity”
time_period	string	Time period for a specific operation type. Property behavior: - *required* if `type` is set to ”time period”
tag	string	Tag name for a specific operation type. Property behavior: - *required* if `type` is set to ”tag name” or ”tag value”
tag_value	string	Tag value for a specific operation type. Property behavior: - *required* if `type` is set to ”tag value”
event_severity	string	Event severity for a specific operation type. Property behavior: - *required* if `type` is set to ”severity”
host_name	string	Host name for a specific operation type. Property behavior: - *required* if `type` is set to ”host visible name”
host_group_name	string	Host group name for a specific operation type. Property behavior: - *required* if `type` is set to ”host group name”
formulaid	string	Arbitrary unique ID that is used to reference the condition from a custom expression. Can only contain capital-case letters. The ID must be defined by the user when modifying filter conditions, but will be generated anew when requesting them afterward. Property behavior: - *required* if CEP rule filter `evaltype` is set to ”custom expression”

CEP rule time window

The CEP rule time window object defines the time window of the CEP rule. It has the following properties.

Property	Type	Description
duration	string	Duration of the CEP rule window (365 days maximum). Accepts seconds and time units with suffix. Default: `10m`
capacity	integer	Capacity of the CEP rule window. Supports user macros. Default: 0 (unlimited)
group_by_host_group	integer	Group by flag for host group. Possible values: 0 - no (default); 1 - yes.
group_by_host	integer	Group by flag for host. Possible values: 0 - no (default); 1 - yes.
group_by_tags	integer	Group by flag for tag. Possible values: 0 - no (default); 1 - yes.
tags	string	Tag names to use for grouping. Property behavior: - *required* if `group_by_tag` is set to ”1”
event_count_tag	string	Tag name for displaying the count of affected events. Property behavior: - *supported* if CEP rule `window_type` is set to ”2”; not allowed otherwise
script	string	Script to execute. Property behavior: - *supported* if CEP rule `window_type` is set to ”4”; not allowed otherwise

CEP rule operation

The CEP rule operation object defines an operation that will be performed when a CEP rule is executed. It has the following properties.

Property	Type	Description
sortorder	integer	Execution order (ascending) of the operation. Property behavior: - *required*
execute_when	integer	Type of condition for operation. Possible values: 0 - event occurred; 1 - event added; 2 - event evicted; 3 - window closed; 4 - event pattern matched; 5 - event closed. Property behavior: - *required*

Property	Type	Description
scope	integer	Scope of the operation. Possible values: 0 - current event (default); 1 - other events; 2 - all events. Possible values when `execute_when` is ”Window closed” or ”Pattern matched”: 2 - all events. Possible values when `execute_when` is ”Event ocurred”: 0 - current event.
filter	object	Filter object.
scope_filter	object	Scope filter object. scope_filter is editable only if the `scope` field contains ”Other events” or ”All events” value. Otherwise scope_filter accepts only object with default values.
type	integer	CEP operation type. Possible values: 1 - set event name; 2 - close event; 3 - discard event; 4 - set event severity; 5 - increase event severity; 6 - decrease event severity; 7 - suppress event; 8 - unsuppress event; 9 - copy first event as new; 10 - copy last event as new; 11 - add tag; 12 - set tag; 13 - set tag value; 14 - increase tag value; 15 - decrease tag value; 16 - rename tag; 17 - remove tag; 18 - close window. Property behavior: - *required* - *supported*: 1, 2, 3, 4, 5, 6, 7, 8, 11, 12, 13, 14, 15, 16, 17 - supported if ”Event occurred” (see `execute_when`) 1, 2, 4, 5, 6, 7, 8, 11, 12, 13, 14, 15, 16, 17, 18 - supported if ”Event added” (see `execute_when`) 1, 2, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18 - supported if ”Event evicted” (see `execute_when`) 1, 2, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18 - supported if ”Event closed” (see `execute_when`) 1, 2, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17 - supported if ”Window closed” (see `execute_when`) 1, 2, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18 - supported if ”Event pattern matched” (see `execute_when`)
event_name	string	Event name for a specific operation type. Property behavior: - *required* if `type` is set to ”Set event name”

Property	Type	Description
tag	string	Tag name for specific operation types. Property behavior: - *required* if `type` is set to ”Add tag”, ”Set tag”, ”Set tag value”, ”Increase tag value”, ”Decrease tag value”, ”Rename tag”, ”Remove tag”
new_tag	string	New tag name for a specific operation type. Property behavior: - *required* if `type` is set to ”Rename tag”
tag_value	string	Tag value for specific operation types. Property behavior: - *required* if `type` is set to ”Add tag”, ”Set tag”, ”Set tag value”, ”Increase tag value”, ”Decrease tag value”, ”Rename tag”, ”Remove tag”
severity	integer	Event severity for a specific operation type. Property behavior: - *required* if `type` is set to ”Set event severity”
suppress_duration	string	Duration of the event suppression. Accepts seconds and time unit with suffix. Property behavior: - *required* if `type` is set to ”Suppress event”

CEP rule operation filter

The CEP rule operation filter object defines a set of conditions that must be met to perform the configured CEP operations. It has the following properties.

Property	Type	Description
conditions	array	Set of operation filter conditions to use for filtering results. The conditions will be sorted in the order of their placement in the formula. Property behavior: - *required*
evaltype	integer	Filter condition evaluation method. Possible values: 0 - And/Or; 1 - And; 2 - Or; 3 - Custom expression. Property behavior: - *required*
eval_formula	string	Generated expression that will be used for evaluating filter conditions. The expression contains IDs that reference specific filter conditions by its `formulaid`. The value of `eval_formula` is equal to the value of `formula` for filters with a custom expression. Property behavior: - *read-only*

Property	Type	Description
formula	string	Arbitrary unique ID that is used to reference the condition from a custom expression. Can only contain capital-case letters. The ID must be defined by the user when modifying filter conditions, but will be generated anew when requesting them afterward. Property behavior: - *required*

CEP rule operation filter condition

The CEP rule operation filter condition object defines a specific condition that must be checked before running the CEP rule operations.

Property	Type	Description
type	integer	Type of operation condition. Possible values: 25 - tag name; 26 - tag value; 30 - problem opened; 31 - first; 32 - last; 33 - symptom; 34 - cloned; 35 - suppressed. Property behavior: - *required* - *supported*: 25, 26, 30, 33, 34, 35 - supported if "Event occurred" (see `execute_when`)
operator	integer	Condition operator. Possible values: 0 - equals (default); 1 - does not equal; 2 - contains; 3 - does not contain; 5 - is greater than or equals; 6 - is less than or equals; 10 - true; 11 - false. Property behavior: - *required* - *supported*: 0, 1, 2, 3 - supported values if `type` set to "Tag name" 0, 1, 2, 3, 5, 6 - supported values if `type` set to "Tag value" 10, 11 - supported values if `type` set to "Problem opened", "First", "Last", "Symptom", "Cloned", "Suppressed".
tag	string	Tag name of operation condition. Property behavior: - *required* if `type` is set to "Tag name" or "Tag value"
tag_value	string	Value to compare with. Property behavior: - *required* if `type` is set to "Tag value"

CEP rule operation scope filter

The CEP rule operation scope filter object defines a set of conditions that must be met to apply the configured CEP operations to events. It has the following properties.

Property	Type	Description
conditions	array	Set of operation scope filter conditions to use for filtering results. The conditions will be sorted in the order of their placement in the formula. Property behavior: - *required*
evaltype	integer	Filter condition evaluation method. Possible values: 0 - And/Or; 1 - And; 2 - Or; 3 - Custom expression. Property behavior: - *required*
eval_formula	string	Generated expression that will be used for evaluating filter conditions. The expression contains IDs that reference specific filter conditions by its `formulaid`. The value of `eval_formula` is equal to the value of `formula` for filters with a custom expression. Property behavior: - *read-only*
formula	string	Arbitrary unique ID that is used to reference the condition from a custom expression. Can only contain capital-case letters. The ID must be defined by the user when modifying filter conditions, but will be generated anew when requesting them afterward. Property behavior: - *required*

CEP rule operation scope filter condition

The CEP rule operation scope filter condition object defines a specific condition that must be checked before applying the CEP rule operations to events.

Property	Type	Description
type	integer	Type of operation condition. Possible values: 25 - tag name; 26 - tag value; 30 - problem opened; 31 - first; 32 - last; 33 - symptom; 34 - cloned; 35 - suppressed. Property behavior: - *required* - *supported*: 25, 26, 30, 33, 34, 35 - supported if ”Event occurred” (see `execute_when`)

Property	Type	Description
operator	integer	Condition operator. Possible values: 0 - equals (default); 1 - does not equal; 2 - contains; 3 - does not contain; 5 - is greater than or equals; 6 - is less than or equals; 10 - true; 11 - false. Property behavior: - *required* - *supported*: 0, 1, 2, 3 - supported values if `type` set to ”Tag name” 0, 1, 2, 3, 5, 6 - supported values if `type` set to ”Tag value” 10, 11 - supported values if `type` set to ”Problem opened”, ”First”, ”Last”, ”Symptom”, ”Cloned”, ”Suppressed”.
tag	string	Tag name of operation condition. Property behavior: - *required* if `type` is set to ”Tag name” or ”Tag value”
tag_value	string	Value to compare with. Property behavior: - *required* if `type` is set to ”Tag value”

ceprule.create

Description

`object ceprule.create(object/array CEP rules)`

This method allows you to create new CEP rules.

Note:
This method is only available to *Super admin* user type. Permissions to call the method can be revoked in user role settings. See User roles for more information.

Parameters

`(object/array)` CEP rules to create.

Additionally to the standard CEP properties, the method accepts the following parameters.

Parameter	Type	Description
operations	array	CEP rule operations to create for the CEP rule. Parameter behavior: - *required*
filter	object	CEP rule filter object for the CEP rule. Parameter behavior: - *required*

Return values

`(object)` Returns an object containing the IDs of the created CEP rules under the `cep_ruleids` property. The order of the returned IDs matches the order of the passed CEP rules.

Examples

Create a new CEP rule

Create a CEP rule using evaluation method `AND/OR` with one condition and one window. By default, the rule will be enabled.

Request:

```
{
    "jsonrpc": "2.0",
    "method": "ceprule.create",
    "params": {
      "name": "Group events by cause",
      "stop": "1",
      "sortorder": "0",
      "description": "",
      "status": "0",
      "window_type": "2",
      "filter": {
        "evaltype": "0",
        "formula": "",
        "conditions": [
          {
            "type": "3",
            "operator": "2",
            "event_name": "Printer",
            "tag": "",
            "tag_value": "",
            "severity": "0",
            "host_name": "",
            "host_group_name": "",
            "time_period": "",
            "formulaid": ""
          }
        ]
      },
      "window": {
        "duration": "1d",
        "capacity": "0",
        "group_by_host_group": "0",
        "group_by_host": "1",
        "group_by_tags": "1",
        "tags": [
          "cause"
        ],
        "event_count_tag": "",
        "script": ""
      },
      "operations": []
    },
    "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "cep_ruleids": [
      "1"
    ]
  },
  "id": 1
}
```

See also

- CEP rule

Source

CCepRule::create() in *ui/include/classes/api/services/CCepRule.php*.

ceprule.delete

Description

`object ceprule.delete(array cep_ruleids)`

This method allows you to delete CEP rules.

> **Note:**
> This method is only available to *Super admin* user type. Permissions to call the method can be revoked in user role settings. See User roles for more information.

Parameters

`(array)` IDs of the CEP rules to delete.

Return values

`(object)` Returns an object containing the IDs of the deleted CEP rules under the `cep_ruleids` property.

Example

Delete multiple CEP rules

Delete two CEP rules.

Request:

```
{
    "jsonrpc": "2.0",
    "method": "ceprule.delete",
    "params": [
        "3",
        "4"
    ],
    "id": 1
}
```

Response:

```
{
    "jsonrpc": "2.0",
    "result": {
        "cep_ruleids": [
            "3",
            "4"
        ]
    },
    "id": 1
}
```

Source

CCepRule::delete() in *ui/include/classes/api/services/CCepRule.php*.

ceprule.get

Description

`integer/array ceprule.get(object parameters)`

The method allows you to retrieve CEP rules according to the given parameters.

Note:
This method is available to users of any type. Permissions to call the method can be revoked in user role settings. See User roles for more information.

Parameters

`(object)` Parameters defining the desired output.

The method supports the following parameters.

Parameter	Type	Description
cep_ruleids	ID/array	Return only CEP rules with the given IDs.
selectFilter	query	Return a `filter` property with the CEP rule conditions.
selectOperations	query	Return an `operations` property with the CEP rule operations.
selectWindow	query	Return a `[window]`(/manual/api/reference/ceprule/object#cep-rule-time-window) property with the CEP rule time window.
sortfield	string/array	Sort the result by the given properties. Possible values: `cep_ruleid`, `name`, `stop`, `sortorder`, `status`.
countOutput	boolean	These parameters are described in the reference commentary.
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Return values

`(integer/array)` Returns either:

- an array of objects;
- the count of retrieved objects, if the `countOutput` parameter has been used.

Examples

Retrieve CEP rules

Retrieve enabled CEP rules, including conditions, windows, and operations.

Request:

```
{
    "jsonrpc": "2.0",
    "method": "ceprule.get",
    "params": {
        "output": "extend",
        "filter": {"status": 0},
        "selectFilter": "extend",
        "selectWindow": "extend",
        "selectOperations": "extend"
    },
    "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": [
    {
      "cep_ruleid": "2",
```

```
      "name": "Group events by cause",
      "stop": "0",
      "sortorder": "1",
      "description": "",
      "status": "0",
      "window_type": "2",
      "error": "",
      "filter": {
        "evaltype": "0",
        "eval_formula": "A",
        "formula": "",
        "conditions": [
          {
            "type": "3",
            "operator": "2",
            "event_name": "Printer",
            "tag": "",
            "tag_value": "",
            "severity": "0",
            "host_name": "",
            "host_group_name": "",
            "time_period": "",
            "formulaid": "A"
          }
        ]
      },
      "window": {
        "duration": "1d",
        "capacity": "0",
        "group_by_host_group": "0",
        "group_by_host": "1",
        "group_by_tags": "1",
        "tags": [
          "cause"
        ],
        "event_count_tag": "symptoms",
        "script": ""
      },
      "operations": []
    },
    {
      "cep_ruleid": "10",
      "name": "Suppress consecutive events",
      "stop": "1",
      "sortorder": "2",
      "description": "",
      "status": "0",
      "window_type": "3",
      "error": "",
      "filter": {
        "evaltype": "0",
        "eval_formula": "A",
        "formula": "",
        "conditions": [
          {
            "type": "3",
            "operator": "2",
            "event_name": "Printer",
            "tag": "",
            "tag_value": "",
            "severity": "0",
            "host_name": "",
```

```
          "host_group_name": "",
          "time_period": "",
          "formulaid": "A"
        }
      ]
    },
    "window": {
      "duration": "3m",
      "capacity": "0",
      "group_by_host_group": "0",
      "group_by_host": "1",
      "group_by_tags": "1",
      "tags": [
        "cause"
      ],
      "event_count_tag": "",
      "script": ""
    },
    "operations": [
      {
        "sortorder": "1",
        "execute_when": "1",
        "type": "7",
        "event_name": "",
        "severity": "0",
        "suppress_duration": "6m",
        "tag": "",
        "new_tag": "",
        "tag_value": "",
        "scope": "0",
        "filter": {
          "evaltype": "0",
          "eval_formula": "A",
          "formula": "",
          "conditions": [
            {
              "type": "33",
              "operator": "10",
              "tag": "",
              "tag_value": "",
              "formulaid": "A"
            }
          ]
        },
        "scope_filter": {
          "evaltype": "0",
          "eval_formula": "",
          "formula": "",
          "conditions": []
        }
      },
      {
        "sortorder": "2",
        "execute_when": "2",
        "type": "8",
        "event_name": "",
        "severity": "0",
        "suppress_duration": "0",
        "tag": "",
        "new_tag": "",
        "tag_value": "",
        "scope": "0",
```

```
          "filter": {
            "evaltype": "0",
            "eval_formula": "A",
            "formula": "",
            "conditions": [
              {
                "type": "35",
                "operator": "10",
                "tag": "",
                "tag_value": "",
                "formulaid": "A"
              }
            ]
          },
          "scope_filter": {
            "evaltype": "0",
            "eval_formula": "",
            "formula": "",
            "conditions": []
          }
        }
      ]
    }
  ],
  "id": 1
}
```

See also

- CEP rule

Source

CCepRule::get() in *ui/include/classes/api/services/CCepRule.php*.

ceprule.resettimewindows

Description

`object ceprule.resettimewindows(object/array cep_ruleid)`

This method allows you to reset time windows for an existing CEP rule.

> **Note:**
> This method is only available to *Super admin* user type. Permissions to call the method can be revoked in user role settings. See User roles for more information.

Parameters

`(object)` CEP rule object must have a `cep_ruleid` property defined.

Return values

`(object)` Returns an object containing the ID of the CEP rule whose time windows have been reset under the `cep_ruleid` property.

Examples

Reset time windows for CEP rule

Request:

```
{
    "jsonrpc": "2.0",
    "method": "ceprule.resettimewindows",
    "params": {
        "cep_ruleid": "1"
    },
```

```
    "id": 1
}
```

Response:

```
{
    "jsonrpc": "2.0",
    "result": {
      "cep_ruleid": "1"
    },
    "id": 1
}
```

See also

- CEP rule

Source

CCepRule::resettimewindows() in *ui/include/classes/api/services/CCepRule.php*.

ceprule.update

Description

`object ceprule.update(object/array CEP rules)`

This method allows you to update existing CEP rules.

Note:
This method is only available to *Super admin* user type. Permissions to call the method can be revoked in user role settings. See User roles for more information.

Parameters

`(object/array)` CEP rule properties to be updated.

The `cep_ruleid` property must be defined for each CEP rule, all other properties are optional. Only the passed properties will be updated, all others will remain unchanged.

Additionally to the standard CEP rule properties, the method accepts the following parameters.

Parameter	Type	Description
filter	object	CEP rule filter object to replace the current filter.
window	array	CEP rule window to be updated. Property behavior: - *required* if `window_type` is changed to "Cause and symptom grouping" and "group_by_host_group", "group_by_host", "group_by_tags" are all set to NO; - *required* if `window_type` is changed to "Event pattern match"; - *supported* if `window_type` is "Simple (multiple)" or "Simple (once)"
operations	array	CEP rule operations to replace existing operations. Property behavior: - *required* if `window_type` is changed to "None" (0), "Simple (multiple)" (1), "Simple (once)" (3) or "Event pattern match" (4).

Return values

`(object)` Returns an object containing the IDs of the updated CEP rules under the `cep_ruleids` property.

Examples

Disable CEP rule

Request:

```
{
    "jsonrpc": "2.0",
    "method": "ceprule.update",
    "params": {
        "cep_ruleid": "1",
        "status": "1"
    },
    "id": 1
}
```

Response:

```
{
    "jsonrpc": "2.0",
    "result": {
        "cep_ruleids": [
            "1"
        ]
    },
    "id": 1
}
```

Replace condition

Request:

```
{
    "jsonrpc": "2.0",
    "method": "ceprule.update",
    "params": {
        "cep_ruleid": "3",
        "filter": {
        "evaltype": "0",
        "formula": "",
            "conditions": [
                {
                    "type": "3",
                    "operator": "2",
                    "event_name": "Inaccessible port",
                    "severity": "0"
                }
            ]
        }
    },
    "id": 1
}
```

Response:

```
{
    "jsonrpc": "2.0",
    "result": {
        "cep_ruleids": [
            "1"
        ]
    },
    "id": 1
}
```

See also

- CEP rule

Source

CCepRule::update() in *ui/include/classes/api/services/CCepRule.php*.

Configuration

Cette classe est conçue pour exporter et importer des données de configuration Zabbix.

Méthodes disponibles:

- configuration.export - exporter la configuration
- configuration.import - importer la configuration
- configuration.importcompare - comparaison du fichier d'importation avec les éléments du système actuel

configuration.export

Description

`string configuration.export(object parameters)`

Cette méthode permet d'exporter les données de configuration sous forme de chaîne sérialisée.

> **Note:**
> Cette méthode est disponible pour les utilisateurs de tout type. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres des rôles utilisateur. Consultez User roles pour plus d'informations.

Paramètres

`(object)` Paramètres définissant les objets à exporter et le format à utiliser.

Parameter	Type	Description
format	string	Format dans lequel les données doivent être exportées. Valeurs possibles : `yaml` - YAML; `xml` - XML; `json` - JSON; `raw` - tableau PHP non traité. Comportement du paramètre : - *required*
prettyprint	boolean	Rendre la sortie plus lisible pour l'utilisateur en ajoutant une indentation. Valeurs possibles : `true` - ajouter une indentation; `false` - *(par défaut)* ne pas ajouter d'indentation.

Parameter	Type	Description
options	object	Objets à exporter. L'objet `options` contient les paramètres suivants : `host_groups` - `(array)` IDs des groupes d'hôtes à exporter; `hosts` - `(array)` IDs des hôtes à exporter; `images` - `(array)` IDs des images à exporter; `maps` - `(array)` IDs des cartes à exporter; `mediaTypes` - `(array)` IDs des types de média à exporter; `template_groups` - `(array)` IDs des groupes de modèles à exporter; `templates` - `(array)` IDs des modèles à exporter; `dashboards` - `(array)` IDs des tableaux de bord à exporter; `global_regexes` - `(array)` IDs des expressions régulières globales à exporter. Les utilisateurs de type *Admin* et *User* peuvent exporter uniquement les objets pour lesquels ils disposent d'une permission en *lecture seule* ou en *lecture-écriture*, ainsi que les images, mais pas les types de média. Comportement du paramètre : - *required*.

Valeurs retournées

`(string)` Renvoie une chaîne sérialisée contenant les données de configuration demandées.

Exemples

Exportation d'un modèle

Exportez la configuration du modèle « 10571 » sous forme de chaîne XML.

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "configuration.export",
    "params": {
        "options": {
            "templates": [
                "10571"
            ]
        },
        "format": "xml"
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": "<?xml version=\"1.0\" encoding=\"UTF-8\"?>\n<zabbix_export><version>8.0</version><template_
    "id": 1
}
```

Source

CConfiguration::export() dans *ui/include/classes/api/services/CConfiguration.php*.

configuration.import

Description

`boolean configuration.import(object parameters)`

Cette méthode permet d'importer des données de configuration à partir d'une chaîne sérialisée.

Note:
Cette méthode est disponible pour les utilisateurs de tout type. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres des rôles utilisateur. Consultez User roles pour plus d'informations.

Paramètres

`(object)` Paramètres contenant les données à importer et les règles de traitement de ces données.

Parameter	Type	Description
format	string	Format de la chaîne sérialisée. Valeurs possibles : `yaml` - YAML; `xml` - XML; `json` - JSON. Comportement du paramètre : - *required*
source	string	Chaîne sérialisée contenant les données de configuration. Comportement du paramètre : - *required*
rules	object	Règles de traitement des nouveaux objets et des objets existants lors de l'import. Les utilisateurs de type *Admin* peuvent importer uniquement les objets pour lesquels ils disposent d'une permission *read-write*, ainsi que les cartes. Par exemple, un hôte et ses entités (éléments, déclencheurs, graphiques, etc.) ne peuvent être importés que si le groupe d'utilisateurs de l'utilisateur dispose d'une permission sur le groupe d'hôtes auquel l'hôte importé appartiendra. Les images et les types de média ne peuvent pas être importés par les utilisateurs de type *Admin*. Le paramètre `rules` est décrit en détail dans le tableau ci-dessous. Comportement du paramètre : - *required*
returnMissingObjects	boolean	Règles indiquant si l'objet de réponse doit contenir des informations sur les objets manquants ; valeur par défaut : `false`.

Attention:
Si aucune règle n'est fournie, la configuration ne sera pas mise à jour.

L'objet `rules` prend en charge les paramètres suivants.

Parameter	Type	Description
dashboards	object	Règles d'import des tableaux de bord. Paramètres pris en charge : `createMissing` - `(boolean)` si défini sur `true`, de nouveaux tableaux de bord seront créés ; valeur par défaut : `false`; `updateExisting` - `(boolean)` si défini sur `true` et qu'un tableau de bord importable particulier est présent dans le système, il sera entièrement mis à jour. Les pages et les widgets seront entièrement remplacés par les nouveaux ; valeur par défaut : `false`.

Parameter	Type	Description
discoveryRules	object	Règles d'import des règles de découverte LLD. Paramètres pris en charge : `createMissing` - `(boolean)` si défini sur `true`, de nouvelles règles LLD seront créées ; valeur par défaut : `false`; `updateExisting` - `(boolean)` si défini sur `true`, les règles LLD existantes seront mises à jour ; valeur par défaut : `false`; `deleteMissing` - `(boolean)` si défini sur `true`, les règles LLD absentes des données importées seront supprimées de la base de données ; valeur par défaut : `false`.
global_regexes	object	Règles d'import des expressions régulières globales. Paramètres pris en charge : `createMissing` - `(boolean)` si défini sur `true`, de nouvelles expressions régulières globales seront créées ; valeur par défaut : `false`; `updateExisting` - `(boolean)` si défini sur `true`, les expressions régulières globales existantes seront mises à jour, avec leurs sous-expressions entièrement remplacées ; valeur par défaut : `false`.
graphs	object	Règles d'import des graphiques. Paramètres pris en charge : `createMissing` - `(boolean)` si défini sur `true`, de nouveaux graphiques seront créés ; valeur par défaut : `false`; `updateExisting` - `(boolean)` si défini sur `true`, les graphiques existants seront mis à jour ; valeur par défaut : `false`; `deleteMissing` - `(boolean)` si défini sur `true`, les graphiques absents des données importées seront supprimés de la base de données ; valeur par défaut : `false`.
host_groups	object	Règles d'import des groupes d'hôtes. Paramètres pris en charge : `createMissing` - `(boolean)` si défini sur `true`, de nouveaux groupes d'hôtes seront créés ; valeur par défaut : `false`; `updateExisting` - `(boolean)` si défini sur `true`, les groupes d'hôtes existants seront mis à jour ; valeur par défaut : `false`.
hosts	object	Règles d'import des hôtes. Paramètres pris en charge : `createMissing` - `(boolean)` si défini sur `true`, de nouveaux hôtes seront créés ; valeur par défaut : `false`; `updateExisting` - `(boolean)` si défini sur `true`, les hôtes existants seront mis à jour ; valeur par défaut : `false`.
httptests	object	Règles d'import des scénarios web. Paramètres pris en charge : `createMissing` - `(boolean)` si défini sur `true`, de nouveaux scénarios web seront créés ; valeur par défaut : `false`; `updateExisting` - `(boolean)` si défini sur `true`, les scénarios web existants seront mis à jour ; valeur par défaut : `false`; `deleteMissing` - `(boolean)` si défini sur `true`, les scénarios web absents des données importées seront supprimés de la base de données ; valeur par défaut : `false`.
images	object	Règles d'import des images. Paramètres pris en charge : `createMissing` - `(boolean)` si défini sur `true`, de nouvelles images seront créées ; valeur par défaut : `false`; `updateExisting` - `(boolean)` si défini sur `true`, les images existantes seront mises à jour ; valeur par défaut : `false`.

Parameter	Type	Description
items	object	Règles d'import des éléments. Paramètres pris en charge : `createMissing` - `(boolean)` si défini sur `true`, de nouveaux éléments seront créés ; valeur par défaut : `false`; `updateExisting` - `(boolean)` si défini sur `true`, les éléments existants seront mis à jour ; valeur par défaut : `false`; `deleteMissing` - `(boolean)` si défini sur `true`, les éléments absents des données importées seront supprimés de la base de données ; valeur par défaut : `false`.
maps	object	Règles d'import des cartes. Paramètres pris en charge : `createMissing` - `(boolean)` si défini sur `true`, de nouvelles cartes seront créées ; valeur par défaut : `false`; `updateExisting` - `(boolean)` si défini sur `true`, les cartes existantes seront mises à jour ; valeur par défaut : `false`.
mediaTypes	object	Règles d'import des types de média. Paramètres pris en charge : `createMissing` - `(boolean)` si défini sur `true`, de nouveaux types de média seront créés ; valeur par défaut : `false`; `updateExisting` - `(boolean)` si défini sur `true`, les types de média existants seront mis à jour ; valeur par défaut : `false`.
templateDashboards	object	Règles d'import des tableaux de bord de modèle. Paramètres pris en charge : `createMissing` - `(boolean)` si défini sur `true`, de nouveaux tableaux de bord de modèle seront créés ; valeur par défaut : `false`; `updateExisting` - `(boolean)` si défini sur `true`, les tableaux de bord de modèle existants seront mis à jour ; valeur par défaut : `false`; `deleteMissing` - `(boolean)` si défini sur `true`, les tableaux de bord de modèle absents des données importées seront supprimés de la base de données ; valeur par défaut : `false`.
template_groups	object	Règles d'import des groupes de modèles. Paramètres pris en charge : `createMissing` - `(boolean)` si défini sur `true`, de nouveaux groupes de modèles seront créés ; valeur par défaut : `false`; `updateExisting` - `(boolean)` si défini sur `true`, les groupes de modèles existants seront mis à jour ; valeur par défaut : `false`.
templateLinkage	object	Règles d'import des liens de modèles. Paramètres pris en charge : `createMissing` - `(boolean)` si défini sur `true`, les modèles qui ne sont pas liés à l'hôte ou au modèle importé, mais qui sont présents dans les données importées, seront liés ; valeur par défaut : `false`; `deleteMissing` - `(boolean)` si défini sur `true`, les modèles liés à l'hôte ou au modèle importé, mais absents des données importées, seront dissociés sans supprimer les entités (éléments, déclencheurs, etc.) héritées des modèles dissociés ; valeur par défaut : `false`.
templates	object	Règles d'import des modèles. Paramètres pris en charge : `createMissing` - `(boolean)` si défini sur `true`, de nouveaux modèles seront créés ; valeur par défaut : `false`; `updateExisting` - `(boolean)` si défini sur `true`, les modèles existants seront mis à jour ; valeur par défaut : `false`.

Parameter	Type	Description
triggers	object	Règles d'import des déclencheurs. Paramètres pris en charge : `createMissing` - `(boolean)` si défini sur `true`, de nouveaux déclencheurs seront créés ; valeur par défaut : `false`; `updateExisting` - `(boolean)` si défini sur `true`, les déclencheurs existants seront mis à jour ; valeur par défaut : `false`; `deleteMissing` - `(boolean)` si défini sur `true`, les déclencheurs absents des données importées seront supprimés de la base de données ; valeur par défaut : `false`.
valueMaps	object	Règles d'import des mappages de valeurs d'hôte ou de modèle. Paramètres pris en charge : `createMissing` - `(boolean)` si défini sur `true`, de nouveaux mappages de valeurs seront créés ; valeur par défaut : `false`; `updateExisting` - `(boolean)` si défini sur `true`, les mappages de valeurs existants seront mis à jour ; valeur par défaut : `false`; `deleteMissing` - `(boolean)` si défini sur `true`, les mappages de valeurs absents des données importées seront supprimés de la base de données ; valeur par défaut : `false`.

Valeurs retournées

`(boolean)` Retourne `true` si l'import est réussi.

Exemples

Importation d'un modèle

Importez la configuration du modèle contenue dans la chaîne XML. Si des éléments ou des déclencheurs dans la chaîne XML sont manquants, ils seront supprimés de la base de données, et tout le reste restera inchangé.

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "configuration.import",
    "params": {
        "format": "xml",
        "rules": {
            "templates": {
                "createMissing": true,
                "updateExisting": true
            },
            "items": {
                "createMissing": true,
                "updateExisting": true,
                "deleteMissing": true
            },
            "triggers": {
                "createMissing": true,
                "updateExisting": true,
                "deleteMissing": true
            },
            "valueMaps": {
                "createMissing": true,
                "updateExisting": false
            }
        },
        "source": "<?xml version=\"1.0\" encoding=\"UTF-8\"?>\n<zabbix_export><version>8.0</version><templ
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": true,
    "id": 1
}
```

Source

CConfiguration::import() dans *ui/include/classes/api/services/CConfiguration.php*.

configuration.importcompare

Description

`array configuration.importcompare(object parameters)`

Cette méthode permet de comparer le fichier d'import avec les éléments actuels du système et affiche ce qui sera modifié si ce fichier d'import est importé.

> **Note:**
> Cette méthode est disponible pour les utilisateurs de tout type. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres du rôle utilisateur. Consultez User roles pour plus d'informations.

Paramètres

`(object)` Paramètres contenant les données possibles à importer et les règles de traitement de ces données.

Parameter	Type	Description
format	string	Format de la chaîne sérialisée. Valeurs possibles : `yaml` - YAML; `xml` - XML; `json` - JSON. Comportement du paramètre : - *required*
source	string	Chaîne sérialisée contenant les données de configuration. Comportement du paramètre : - *required*
rules	object	Règles indiquant comment comparer les objets nouveaux et existants. Les utilisateurs de type *Admin* et *User* peuvent comparer les nouveaux objets uniquement avec les objets existants pour lesquels ils disposent d'une permission en *lecture seule* ou en *lecture-écriture*. Le paramètre `rules` est décrit en détail dans le tableau ci-dessous. Comportement du paramètre : - *required*
returnMissingObjects	boolean	Règles indiquant si l'objet de réponse doit contenir des informations sur les objets manquants ; valeur par défaut : `false`.

> **Attention:**
> Si aucune règle n'est fournie, il n'y aura rien à mettre à jour et le résultat sera vide.

> **Note:**
> La comparaison sera effectuée uniquement pour les groupes d'hôtes et les modèles. Les déclencheurs et les graphiques seront comparés uniquement pour les modèles importés ; tout autre élément sera considéré comme "new".

L'objet `rules` prend en charge les paramètres suivants.

Parameter	Type	Description
dashboards	object	Règles d'importation des tableaux de bord. Paramètres pris en charge : `createMissing` - `(boolean)` si défini sur `true`, de nouveaux tableaux de bord seront créés ; valeur par défaut : `false`; `updateExisting` - `(boolean)` si défini sur `true` et qu'un tableau de bord importable particulier est présent dans le système, il sera entièrement mis à jour. Les pages et les widgets seront entièrement remplacés par les nouveaux ; valeur par défaut : `false`.
discoveryRules	object	Règles d'importation des règles de découverte LLD. Paramètres pris en charge : `createMissing` - `(boolean)` si défini sur `true`, de nouvelles règles LLD seront créées ; valeur par défaut : `false`; `updateExisting` - `(boolean)` si défini sur `true`, les règles LLD existantes seront mises à jour ; valeur par défaut : `false`; `deleteMissing` - `(boolean)` si défini sur `true`, les règles LLD absentes des données importées seront supprimées de la base de données ; valeur par défaut : `false`.
global_regexes	object	Règles d'importation des expressions régulières globales. Paramètres pris en charge : `createMissing` - `(boolean)` si défini sur `true`, de nouvelles expressions régulières globales seront créées ; valeur par défaut : `false`; `updateExisting` - `(boolean)` si défini sur `true`, les expressions régulières globales existantes seront mises à jour, avec leurs sous-expressions entièrement remplacées ; valeur par défaut : `false`.
graphs	object	Règles d'importation des graphiques. Paramètres pris en charge : `createMissing` - `(boolean)` si défini sur `true`, de nouveaux graphiques seront créés ; valeur par défaut : `false`; `updateExisting` - `(boolean)` si défini sur `true`, les graphiques existants seront mis à jour ; valeur par défaut : `false`; `deleteMissing` - `(boolean)` si défini sur `true`, les graphiques absents des données importées seront supprimés de la base de données ; valeur par défaut : `false`.
host_groups	object	Règles d'importation des groupes d'hôtes. Paramètres pris en charge : `createMissing` - `(boolean)` si défini sur `true`, de nouveaux groupes d'hôtes seront créés ; valeur par défaut : `false`; `updateExisting` - `(boolean)` si défini sur `true`, les groupes d'hôtes existants seront mis à jour ; valeur par défaut : `false`.
httptests	object	Règles d'importation des scénarios web. Paramètres pris en charge : `createMissing` - `(boolean)` si défini sur `true`, de nouveaux scénarios web seront créés ; valeur par défaut : `false`; `updateExisting` - `(boolean)` si défini sur `true`, les scénarios web existants seront mis à jour ; valeur par défaut : `false`; `deleteMissing` - `(boolean)` si défini sur `true`, les scénarios web absents des données importées seront supprimés de la base de données ; valeur par défaut : `false`.

Parameter	Type	Description
images	object	Règles d'importation des images. Paramètres pris en charge : `createMissing` - `(boolean)` si défini sur `true`, de nouvelles images seront créées ; valeur par défaut : `false`; `updateExisting` - `(boolean)` si défini sur `true`, les images existantes seront mises à jour ; valeur par défaut : `false`. Ce paramètre n'aura aucun effet sur la sortie. Il est autorisé uniquement pour assurer la cohérence avec `configuration.import`.
items	object	Règles d'importation des éléments. Paramètres pris en charge : `createMissing` - `(boolean)` si défini sur `true`, de nouveaux éléments seront créés ; valeur par défaut : `false`; `updateExisting` - `(boolean)` si défini sur `true`, les éléments existants seront mis à jour ; valeur par défaut : `false`; `deleteMissing` - `(boolean)` si défini sur `true`, les éléments absents des données importées seront supprimés de la base de données ; valeur par défaut : `false`.
maps	object	Règles d'importation des cartes. Paramètres pris en charge : `createMissing` - `(boolean)` si défini sur `true`, de nouvelles cartes seront créées ; valeur par défaut : `false`; `updateExisting` - `(boolean)` si défini sur `true`, les cartes existantes seront mises à jour ; valeur par défaut : `false`. Ce paramètre n'aura aucun effet sur la sortie. Il est autorisé uniquement pour assurer la cohérence avec `configuration.import`.
mediaTypes	object	Règles d'importation des types de média. Paramètres pris en charge : `createMissing` - `(boolean)` si défini sur `true`, de nouveaux types de média seront créés ; valeur par défaut : `false`; `updateExisting` - `(boolean)` si défini sur `true`, les types de média existants seront mis à jour ; valeur par défaut : `false`. Ce paramètre n'aura aucun effet sur la sortie. Il est autorisé uniquement pour assurer la cohérence avec `configuration.import`.
templateDashboards	object	Règles d'importation des tableaux de bord de modèle. Paramètres pris en charge : `createMissing` - `(boolean)` si défini sur `true`, de nouveaux tableaux de bord de modèle seront créés ; valeur par défaut : `false`; `updateExisting` - `(boolean)` si défini sur `true`, les tableaux de bord de modèle existants seront mis à jour ; valeur par défaut : `false`; `deleteMissing` - `(boolean)` si défini sur `true`, les tableaux de bord de modèle absents des données importées seront supprimés de la base de données ; valeur par défaut : `false`.
template_groups	object	Règles d'importation des groupes de modèles. Paramètres pris en charge : `createMissing` - `(boolean)` si défini sur `true`, de nouveaux groupes de modèles seront créés ; valeur par défaut : `false`; `updateExisting` - `(boolean)` si défini sur `true`, les groupes de modèles existants seront mis à jour ; valeur par défaut : `false`.

Parameter	Type	Description
templateLinkage	object	Règles d'importation des liens de modèles. Paramètres pris en charge : `createMissing` - `(boolean)` si défini sur `true`, les modèles qui ne sont pas liés à l'hôte ou au modèle en cours d'importation, mais qui sont présents dans les données importées, seront liés ; valeur par défaut : `false`; `deleteMissing` - `(boolean)` si défini sur `true`, les modèles liés à l'hôte ou au modèle en cours d'importation, mais absents des données importées, seront dissociés sans supprimer les entités (éléments, déclencheurs, etc.) héritées des modèles dissociés ; valeur par défaut : `false`.
templates	object	Règles d'importation des modèles. Paramètres pris en charge : `createMissing` - `(boolean)` si défini sur `true`, de nouveaux modèles seront créés ; valeur par défaut : `false`; `updateExisting` - `(boolean)` si défini sur `true`, les modèles existants seront mis à jour ; valeur par défaut : `false`.
triggers	object	Règles d'importation des déclencheurs. Paramètres pris en charge : `createMissing` - `(boolean)` si défini sur `true`, de nouveaux déclencheurs seront créés ; valeur par défaut : `false`; `updateExisting` - `(boolean)` si défini sur `true`, les déclencheurs existants seront mis à jour ; valeur par défaut : `false`; `deleteMissing` - `(boolean)` si défini sur `true`, les déclencheurs absents des données importées seront supprimés de la base de données ; valeur par défaut : `false`.
valueMaps	object	Règles d'importation des mappages de valeurs de l'hôte ou du modèle. Paramètres pris en charge : `createMissing` - `(boolean)` si défini sur `true`, de nouveaux mappages de valeurs seront créés ; valeur par défaut : `false`; `updateExisting` - `(boolean)` si défini sur `true`, les mappages de valeurs existants seront mis à jour ; valeur par défaut : `false`; `deleteMissing` - `(boolean)` si défini sur `true`, les mappages de valeurs absents des données importées seront supprimés de la base de données ; valeur par défaut : `false`.

Valeurs retournées

`(array)` Renvoie un tableau avec les changements de configuration qui seront effectués.

Exemples

Comparaison de l'importation d'un modèle

Comparez le modèle contenu dans la chaîne XML aux éléments actuels du système, et affichez ce qui sera modifié si ce modèle est importé.

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "configuration.importcompare",
    "params": {
        "format": "xml",
        "rules": {
            "discoveryRules": {
                "createMissing": true,
                "updateExisting": true,
                "deleteMissing": true
            },
```

```
            "graphs": {
                "createMissing": true,
                "updateExisting": true,
                "deleteMissing": true
            },
            "host_groups": {
                "createMissing": true,
                "updateExisting": true
            },
            "template_groups": {
                "createMissing": true,
                "updateExisting": true
            },
            "httptests": {
                "createMissing": true,
                "updateExisting": true,
                "deleteMissing": true
            },
            "items": {
                "createMissing": true,
                "updateExisting": true,
                "deleteMissing": true
            },
            "templateLinkage": {
                "createMissing": true,
                "deleteMissing": true
            },
            "templates": {
                "createMissing": true,
                "updateExisting": true
            },
            "templateDashboards": {
                "createMissing": true,
                "updateExisting": true,
                "deleteMissing": true
            },
            "triggers": {
                "createMissing": true,
                "updateExisting": true,
                "deleteMissing": true
            },
            "valueMaps": {
                "createMissing": true,
                "updateExisting": true,
                "deleteMissing": true
            }
        },
        "source": "<?xml version=\"1.0\" encoding=\"UTF-8\"?>\n<zabbix_export><version>8.0</version><templ
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "templates": {
            "updated": [
                {
                    "before": {
                        "uuid": "5aef0444a82a4d8cb7a95dc4c0c85330",
                        "template": "New template",
```

```
            "name": "New template",
            "groups": [
                {
                    "name": "Templates"
                }
            ]
        },
        "after": {
            "uuid": "5aef0444a82a4d8cb7a95dc4c0c85330",
            "template": "New template",
            "name": "New template",
            "groups": [
                {
                    "name": "Templates"
                }
            ]
        },
        "items": {
            "added": [
                {
                    "after": {
                        "uuid": "648006da5971424ead0c47ddbbf1ea2e",
                        "name": "CPU utilization",
                        "key": "system.cpu.util",
                        "value_type": "FLOAT",
                        "units": "%"
                    },
                    "triggers": {
                        "added": [
                            {
                                "after": {
                                    "uuid": "736225012c534ec480c2a66a91322ce0",
                                    "expression": "avg(/New template/system.cpu.util,3m)>70",
                                    "name": "CPU utilization too high on 'New host' for 3 minu
                                    "priority": "WARNING"
                                }
                            }
                        ]
                    }
                }
            ],
            "removed": [
                {
                    "before": {
                        "uuid": "6805d4c39a624a8bab2cc8ab63df1ab3",
                        "name": "CPU load",
                        "key": "system.cpu.load",
                        "value_type": "FLOAT"
                    },
                    "triggers": {
                        "removed": [
                            {
                                "before": {
                                    "uuid": "ab4c2526c2bc42e48a633082255ebcb3",
                                    "expression": "avg(/New template/system.cpu.load,3m)>2",
                                    "name": "CPU load too high on 'New host' for 3 minutes",
                                    "priority": "WARNING"
                                }
                            }
                        ]
                    }
                }
```

```
                        ],
                        "updated": [
                            {
                                "before": {
                                    "uuid": "7f1e6f1e48aa4a128e5b6a958a5d11c3",
                                    "name": "Zabbix agent ping",
                                    "key": "agent.ping"
                                },
                                "after": {
                                    "uuid": "7f1e6f1e48aa4a128e5b6a958a5d11c3",
                                    "name": "Zabbix agent ping",
                                    "key": "agent.ping",
                                    "delay": "3m"
                                }
                            }
                        ]
                    }
                }
            ]
        }
    },
    "id": 1
}
```


Source

CConfiguration::importcompare() dans *ui/include/classes/api/services/CConfiguration.php*.

Connecteur

Cette classe est conçue pour fonctionner avec des connecteurs.

Références d'objet :

- Connecteur
- Filtre de tags

Méthodes disponibles :

- connector.create - créer de nouveaux connecteurs
- connector.delete - supprimer des connecteurs
- connector.get - récupérer des connecteurs
- connector.update - mettre à jour des connecteurs

Objet connecteur

Les objets suivants sont directement liés à l'API `connector`.

Connecteur

L'objet connecteur possède les propriétés suivantes.

Property	Type	Description
connectorid	ID	ID du connecteur. Comportement de la propriété : - *lecture seule* - *requis* pour les opérations de mise à jour
name	string	Nom du connecteur. Comportement de la propriété : - *requis* pour les opérations de création

Property	Type	Description
url	string	URL du point de terminaison, c'est-à-dire l'URL du récepteur. Les macros utilisateur sont prises en charge. Comportement de la propriété : - *requis* pour les opérations de création
protocol	integer	Protocole de communication. Valeurs possibles : 0 - *(par défaut)* Zabbix Streaming Protocol v1.0.
data_type	integer	Type de données. Valeurs possibles : 0 - *(par défaut)* Valeurs des éléments ; 1 - Événements.
item_value_type	integer	Somme des types de valeurs des éléments à envoyer. Valeurs possibles : 1 - Numérique (flottant) ; 2 - Caractère ; 4 - Journal ; 8 - Numérique (non signé) ; 16 - Texte ; 32 - Binaire ; 64 - JSON. Valeur par défaut : 31 - Tous les types d'éléments (à l'exception des types binaire et JSON). Comportement de la propriété : - *pris en charge* si `data_type` est défini sur « Item values ».
max_records	integer	Nombre maximal d'événements ou d'éléments pouvant être envoyés dans un même message. Valeurs possibles : 0-2147483647 (valeur maximale d'un entier signé sur 32 bits). Valeur par défaut : 0 - Illimité.
max_senders	integer	Nombre de processus d'envoi à exécuter pour ce connecteur. Valeurs possibles : 1-100. Valeur par défaut : 1.
max_attempts	integer	Nombre de tentatives. Valeurs possibles : 1-5. Valeur par défaut : 1.
attempt_interval	string	Intervalle entre les tentatives de nouvelle exécution. Accepte les secondes. Valeurs possibles : 0s-10s. Valeur par défaut : 5s. Comportement de la propriété : - *pris en charge* si `max_attempts` est supérieur à 1.

Property	Type	Description
timeout	string	Délai d'expiration. Les suffixes de durée sont pris en charge (par exemple, 30s, 1m). Les macros utilisateur sont prises en charge. Valeurs possibles : 1s-60s. Valeur par défaut : 5s.
http_proxy	string	Chaîne de connexion au proxy HTTP(S), indiquée sous la forme *[protocol]://[username[:password]@]proxy.example.com[:port]*. Les macros utilisateur sont prises en charge.
authtype	integer	Méthode d'authentification HTTP. Valeurs possibles : 0 - *(par défaut)* Aucune ; 1 - Basic ; 2 - NTLM ; 3 - Kerberos ; 4 - Digest ; 5 - Bearer.
username	string	Nom d'utilisateur. Les macros utilisateur sont prises en charge. Comportement de la propriété : - *pris en charge* si `authtype` est défini sur « Basic », « NTLM », « Kerberos » ou « Digest »
password	string	Mot de passe. Les macros utilisateur sont prises en charge. Comportement de la propriété : - *pris en charge* si `authtype` est défini sur « Basic », « NTLM », « Kerberos » ou « Digest »
token	string	Jeton Bearer. Les macros utilisateur sont prises en charge. Comportement de la propriété : - *requis* si `authtype` est défini sur « Bearer »
verify_peer	integer	Indique s'il faut vérifier que le certificat du hôte est authentique. Valeurs possibles : 0 - Ne pas vérifier ; 1 - *(par défaut)* Vérifier.
verify_host	integer	Indique s'il faut vérifier que le nom du hôte utilisé pour la connexion correspond à celui indiqué dans le certificat du hôte. Valeurs possibles : 0 - Ne pas vérifier ; 1 - *(par défaut)* Vérifier.
ssl_cert_file	string	Chemin du fichier de clé SSL publique. Les macros utilisateur sont prises en charge.
ssl_key_file	string	Chemin du fichier de clé SSL privée. Les macros utilisateur sont prises en charge.
ssl_key_password	string	Mot de passe du fichier de clé SSL. Les macros utilisateur sont prises en charge.
description	text	Description du connecteur.
status	integer	Indique si le connecteur est activé. Valeurs possibles : 0 - Désactivé ; 1 - *(par défaut)* Activé.

Property	Type	Description
tags_evaltype	integer	Méthode d'évaluation des tags. Valeurs possibles : 0 - *(par défaut)* Et/Ou ; 2 - Ou.

Filtre de balises

Le filtre de balises vous permet d'exporter uniquement les valeurs d'élément ou les événements correspondants. S'il n'est pas défini, tout sera exporté. L'objet de filtre de balises possède les propriétés suivantes.

Property	Type	Description
tag	string	Nom de la balise. Property behavior: - *required*
operator	integer	Condition operator. Possible values: 0 - *(default)* Equals; 1 - Does not equal; 2 - Contains; 3 - Does not contain; 12 - Exists; 13 - Does not exist.
value	string	Valeur de la balise.

connector.create

Description

`object connector.create(object/array connectors)`

Cette méthode vous permet de créer de nouveaux objets connecteur.

Note:
Cette méthode est disponible uniquement pour le type d'utilisateur *Super admin*. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres des rôles utilisateur. Consultez User roles pour plus d'informations.

Paramètres

`(object/array)` Objets connecteur à créer.

En plus des propriétés standard du connecteur, la méthode accepte les paramètres suivants.

Paramètre	Type	Description
tags	array	Filtre de tags du connecteur.

Valeurs de retour

`(object)` Renvoie un objet contenant les IDs des connecteurs créés sous la propriété `connectorids`. L'ordre des IDs renvoyés correspond à l'ordre des connecteurs transmis.

Exemples

Création d'un connecteur

Créez un connecteur pour exporter les événements de déclencheur avec un filtre de balises. L'authentification HTTP sera effectuée à l'aide d'un jeton Bearer.

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "connector.create",
    "params": [
        {
            "name": "Export of events",
            "data_type": 1,
            "url": "{$DATA_EXPORT_URL}",
            "authtype": 5,
            "token": "{$DATA_EXPORT_BEARER_TOKEN}",
            "tags": [
                {
                    "tag": "service",
                    "operator": 0,
                    "value": "mysqld"
                }
            ]
        }
    ],
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "connectorid": [
            "3"
        ]
    },
    "id": 1
}
```

Source

CConnector::create() dans *ui/include/classes/api/services/CConnector.php*.

connector.delete

Description

`object connector.delete(array connectorids)`

Cette méthode vous permet de supprimer des entrées de connecteur.

> **Note:**
> Cette méthode est disponible uniquement pour le type d'utilisateur *Super admin*. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres des rôles utilisateur. Consultez User roles pour plus d'informations.

Paramètres

`(array)` ID des connecteurs à supprimer.

Valeurs de retour

`(object)` Renvoie un objet contenant les IDs des connecteurs supprimés sous la propriété `connectorids`.

Exemples

Suppression de plusieurs connecteurs

Supprimez deux entrées de connecteur.

Requête :

```
{
    "jsonrpc": "2.0",
```

```
    "method": "connector.delete",
    "params": [
        3,
        5
    ],
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "connectorids": [
            "3",
            "5"
        ]
    },
    "id": 1
}
```

Source

CConnector::delete() dans *ui/include/classes/api/services/CConnector.php*.

connector.get

Description

`integer/array connector.get(object parameters)`

La méthode permet de récupérer des objets connector selon les paramètres fournis.

Note:
Cette méthode est disponible uniquement pour le type d'utilisateur *Super admin*. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres du rôle utilisateur. Consultez Rôles utilisateur pour plus d'informations.

Paramètres

`(object)` Paramètres définissant la sortie souhaitée.

La méthode prend en charge les paramètres suivants.

Paramètre	Type	Description
connectorids	ID/array	Retourner uniquement les connecteurs avec les ID donnés.
selectTags	query	Retourner une propriété `tags` avec le filtre de tags du connecteur. Prend en charge `count`.
sortfield	string/array	Trier le résultat selon les propriétés données. Valeurs possibles : `connectorid`, `name`, `data_type`, `status`.
countOutput	boolean	Ces paramètres sont décrits dans le commentaire de référence.
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Valeurs de retour

((integer/array) Renvoie soit :

- un tableau d'objets ;
- le nombre d'objets récupérés, si le paramètre `countOutput` a été utilisé.

Exemples

Récupération de tous les connecteurs

Récupérez toutes les données sur tous les connecteurs et leurs propriétés.

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "connector.get",
    "params": {
        "output": "extend",
        "selectTags": ["tag", "operator", "value"],
        "preservekeys": true
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "connectorid": "1",
            "name": "Export of item values",
            "protocol": "0",
            "data_type": "0",
            "url": "{$DATA_EXPORT_VALUES_URL}",
            "item_value_type": "31",
            "authtype": "4",
            "username": "{$DATA_EXPORT_VALUES_USERNAME}",
            "password": "{$DATA_EXPORT_VALUES_PASSWORD}",
            "token": "",
            "max_records": "0",
            "max_senders": "4",
            "max_attempts": "2",
            "attempt_interval": "10s",
            "timeout": "10s",
            "http_proxy": "{$DATA_EXPORT_VALUES_PROXY}",
            "verify_peer": "1",
            "verify_host": "1",
            "ssl_cert_file": "{$DATA_EXPORT_VALUES_SSL_CERT_FILE}",
            "ssl_key_file": "{$DATA_EXPORT_VALUES_SSL_KEY_FILE}",
            "ssl_key_password": "",
            "description": "",
            "status": "1",
            "tags_evaltype": "0",
            "tags": [
                {
                    "tag": "component",
                    "operator": "0",
                    "value": "memory"
                }
            ]
        },
        {
            "connectorid": "2",
            "name": "Export of events",
            "protocol": "0",
            "data_type": "1",
```

```
            "url": "{$DATA_EXPORT_EVENTS_URL}",
            "item_value_type": "31",
            "authtype": "5",
            "username": "",
            "password": "",
            "token": "{$DATA_EXPORT_EVENTS_BEARER_TOKEN}",
            "max_records": "0",
            "max_senders": "2",
            "max_attempts": "1",
            "attempt_interval": "5s",
            "timeout": "5s",
            "http_proxy": "",
            "verify_peer": "1",
            "verify_host": "1",
            "ssl_cert_file": "",
            "ssl_key_file": "",
            "ssl_key_password": "",
            "description": "",
            "status": "1",
            "tags_evaltype": "0",
            "tags": [
                {
                    "tag": "scope",
                    "operator": "0",
                    "value": "performance"
                }
            ]
        }
    ],
    "id": 1
}
```

Source

CConnector:get() dans *ui/include/classes/api/services/CConnector.php*.

connector.update

Description

`object connector.update(object/array connectors)`

Cette méthode vous permet de mettre à jour des connecteurs existants.

> **Note:**
> Cette méthode est disponible uniquement pour le type d'utilisateur *Super admin*. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres des rôles utilisateur. Voir User roles pour plus d'informations.

Paramètres

`(object/array)` Propriétés du connecteur à mettre à jour.

La propriété `connectorid` doit être définie pour chaque connecteur, toutes les autres propriétés sont facultatives. Seules les propriétés transmises seront mises à jour, toutes les autres resteront inchangées.

En plus des propriétés standard du connecteur, la méthode accepte les paramètres suivants.

Paramètre	Type	Description
tags	array	Filtre de tags du connecteur pour remplacer le filtre de tags actuel.

Valeurs de retour

`(object)` Renvoie un objet contenant les ID des connecteurs mis à jour sous la propriété `connectorids`.

Exemples

Modification du type d'authentification HTTP

Modifiez le type d'authentification HTTP en Bearer pour le connecteur avec l'ID « 3 ».

Request:

```
{
    "jsonrpc": "2.0",
    "method": "connector.update",
    "params": {
        "connectorid": 3,
        "authtype": 5,
        "token": "{$DATA_EXPORT_BEARER_TOKEN}"
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "connectorids": [
            "3"
        ]
    },
    "id": 1
}
```

Mise à jour du filtre de tags

Modifiez le filtre de tags pour le connecteur avec l'ID « 5 ».

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "connector.update",
    "params": [
        {
            "connectorid": 5,
            "tags_evaltype": 2,
            "tags": [
                {
                    "tag": "service",
                    "operator": 0,
                    "value": "mysqld"
                },
                {
                    "tag": "error",
                    "operator": 12,
                    "value": ""
                }
            ]
        }
    ],
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "connectorids": [
            "5"
        ]
    },
```

```
    "id": 1
}
```

Source

CConnector::update() dans *ui/include/classes/api/services/CConnector.php*.

Correspondance de valeurs

Cette classe est conçue pour fonctionner avec les correspondances de valeurs.

Références d'objet :

- Correspondance de valeurs
  - Correspondances de valeurs

Méthodes disponibles :

- valuemap.create - créer de nouvelles correspondances de valeurs
- valuemap.delete - supprimer des correspondances de valeurs
- valuemap.get - récupérer des correspondances de valeurs
- valuemap.update - mettre à jour des correspondances de valeurs

Objet de correspondance de valeurs

Les objets suivants sont directement liés à l'API `valuemap`.

Correspondance de valeurs

L'objet de correspondance de valeurs possède les propriétés suivantes.

Propriété	Type	Description
valuemapid	ID	ID de la correspondance de valeurs. Comportement de la propriété: - *lecture seule* - *requis* pour les opérations de mise à jour
hostid	ID	ID de l'hôte ou du modèle auquel appartient la correspondance de valeurs. Comportement de la propriété: - *constant* - *requis* pour les opérations de création
name	string	Nom de la correspondance de valeurs. Comportement de la propriété: - *requis* pour les opérations de création
mappings	array	Correspondances de valeurs pour la correspondance de valeurs actuelle. L'objet de correspondance est décrit en détail ci-dessous. Comportement de la propriété: - *requis* pour les opérations de création
uuid	string	Identifiant unique universel, utilisé pour lier les correspondances de valeurs importées à celles qui existent déjà. Généré automatiquement s'il n'est pas fourni. Comportement de la propriété: - *pris en charge* si la correspondance de valeurs appartient à un modèle

Correspondances de valeurs

L'objet de correspondances de valeurs définit les correspondances de valeurs de la table de correspondance des valeurs. Il possède les propriétés suivantes.

Propriété	Type	Description
type	integer	Type de correspondance. Valeurs possibles : 0 - *(par défaut)* la correspondance sera appliquée si la valeur est égale ; 1 - la correspondance sera appliquée si la valeur est supérieure ou égale[1] ; 2 - la correspondance sera appliquée si la valeur est inférieure ou égale[1] ; 3 - la correspondance sera appliquée si la valeur est dans une plage (les plages sont inclusives ; plusieurs plages, séparées par une virgule, peuvent être définies)[1] ; 4 - la correspondance sera appliquée si la valeur correspond à une expression régulière[2] ; 5 - si aucune correspondance n'est trouvée, la correspondance ne sera pas appliquée et la valeur par défaut sera utilisée. Si `type` est défini sur "0", "1", "2", "3", "4", alors `value` ne peut pas être vide. Si `type` est défini sur "5", alors `value` doit être vide.
value	string	Valeur d'origine. Comportement de la propriété : - *obligatoire* si `type` est défini sur "1", "2", "3", "4" - *pris en charge* si `type` est défini sur "5"
newvalue	string	Valeur vers laquelle la valeur d'origine est mappée. Comportement de la propriété : - *obligatoire*

[1] pris en charge uniquement pour les éléments ayant le type de valeur "numeric unsigned", "numeric float".
 [2] pris en charge uniquement pour les éléments ayant le type de valeur "character".

valuemap.create

Description

`object valuemap.create(object/array valuemaps)`

Cette méthode vous permet de créer de nouvelles correspondances de valeurs.

> **Note:**
> Cette méthode est disponible uniquement pour le type d'utilisateur *Super admin*. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres des rôles utilisateur. Voir User roles pour plus d'informations.

Paramètres

`(object/array)` Correspondances de valeurs à créer.

La méthode accepte des correspondances de valeurs avec les propriétés standard des correspondances de valeurs.

Valeurs retournées

`(object)` Retourne un objet contenant les identifiants des tables de correspondance créées dans la propriété `valuemapids`. L'ordre des ID retournés correspond à l'ordre des tables de correspondance transmises.

Exemples

Création d'une correspondance de valeurs

Créez une correspondance de valeurs avec deux mappages.

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "valuemap.create",
    "params": {
        "hostid": "50009",
        "name": "Service state",
        "mappings": [
            {
                "type": "1",
                "value": "1",
                "newvalue": "Up"
            },
            {
                "type": "5",
                "newvalue": "Down"
            }
        ]
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "valuemapids": [
            "1"
        ]
    },
    "id": 1
}
```

Source

CValueMap::create() dans *ui/include/classes/api/services/CValueMap.php*.

valuemap.delete

Description

`object valuemap.delete(array valuemapids)`

Cette méthode vous permet de supprimer des mappages de valeurs.

> **Note:**
> Cette méthode est disponible uniquement pour le type d'utilisateur *Super admin*. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres des rôles utilisateur. Consultez User roles pour plus d'informations.

Paramètres

`(array)` ID des correspondances de valeurs à supprimer.

Valeurs de retour

`(object)` Renvoie un objet contenant les ID des mappages de valeurs supprimés sous la propriété `valuemapids`.

Exemples

Suppression de plusieurs correspondances de valeurs

Supprimez deux correspondances de valeurs.

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "valuemap.delete",
```

```
    "params": [
        "1",
        "2"
    ],
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "valuemapids": [
            "1",
            "2"
        ]
    },
    "id": 1
}
```

Source

CValueMap::delete() dans *ui/include/classes/api/services/CValueMap.php*.

valuemap.get

Description

`integer/array valuemap.get(object parameters)`

La méthode permet de récupérer des mappages de valeurs selon les paramètres fournis.

Note:
Cette méthode est disponible pour les utilisateurs de tout type. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres du rôle utilisateur. Consultez Rôles utilisateur pour plus d'informations.

Paramètres

`(object)` Paramètres définissant la sortie souhaitée.

La méthode prend en charge les paramètres suivants.

Paramètre	Type	Description
valuemapids	ID/array	Retourner uniquement les correspondances de valeurs avec les ID donnés.
selectMappings	query	Retourner les correspondances de valeurs pour la correspondance de valeurs actuelle dans la propriété `mappings`. Prend en charge `count`.
sortfield	string/array	Trier le résultat selon les propriétés données. Valeurs possibles : `valuemapid`, `name`.
countOutput	boolean	Ces paramètres sont décrits dans le commentaire de référence.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Valeurs retournées

`(integer/array)` Retourne soit :

- un tableau d'objets ;
- le nombre d'objets récupérés, si le paramètre `countOutput` a été utilisé.

Exemples

Récupération des correspondances de valeurs

Récupérez toutes les correspondances de valeurs configurées.

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "valuemap.get",
    "params": {
        "output": "extend"
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "valuemapid": "4",
            "name": "APC Battery Replacement Status"
        },
        {
            "valuemapid": "5",
            "name": "APC Battery Status"
        },
        {
            "valuemapid": "7",
            "name": "Dell Open Manage System Status"
        }
    ],
    "id": 1
}
```

Récupérez une correspondance de valeurs avec ses mappages.

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "valuemap.get",
    "params": {
        "output": "extend",
        "selectMappings": "extend",
        "valuemapids": ["4"]
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "valuemapid": "4",
            "name": "APC Battery Replacement Status",
            "mappings": [
```

```
                {
                    "type": "0",
                    "value": "1",
                    "newvalue": "unknown"
                },
                {
                    "type": "0",
                    "value": "2",
                    "newvalue": "notInstalled"
                },
                {
                    "type": "0",
                    "value": "3",
                    "newvalue": "ok"
                },
                {
                    "type": "0",
                    "value": "4",
                    "newvalue": "failed"
                },
                {
                    "type": "0",
                    "value": "5",
                    "newvalue": "highTemperature"
                },
                {
                    "type": "0",
                    "value": "6",
                    "newvalue": "replaceImmediately"
                },
                {
                    "type": "0",
                    "value": "7",
                    "newvalue": "lowCapacity"
                }
            ]
        }
    ],
    "id": 1
}
```


Source

CValueMap::get() dans *ui/include/classes/api/services/CValueMap.php*.

valuemap.update

Description

`object valuemap.update(object/array valuemaps)`

Cette méthode vous permet de mettre à jour des mappages de valeurs existants.

> **Note:**
> Cette méthode est disponible uniquement pour le type d'utilisateur *Super admin*. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres du rôle utilisateur. Voir User roles pour plus d'informations.

Paramètres

`(object/array)` Propriétés de la correspondance de valeurs à mettre à jour.

La propriété `valuemapid` doit être définie pour chaque correspondance de valeurs, toutes les autres propriétés sont facultatives. Seules les propriétés transmises seront mises à jour, toutes les autres resteront inchangées.

Valeurs retournées

(`object`) Retourne un objet contenant les ID des tables de correspondance mises à jour sous la propriété `valuemapids`.

Exemples

Modification du nom de la correspondance de valeurs

Modifiez le nom de la correspondance de valeurs en « Device status ».

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "valuemap.update",
    "params": {
        "valuemapid": "2",
        "name": "Device status"
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "valuemapids": [
            "2"
        ]
    },
    "id": 1
}
```

Modification des correspondances pour une carte de valeurs.

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "valuemap.update",
    "params": {
        "valuemapid": "2",
        "mappings": [
            {
                "type": "0",
                "value": "0",
                "newvalue": "Online"
            },
            {
                "type": "0",
                "value": "1",
                "newvalue": "Offline"
            }
        ]
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "valuemapids": [
            "2"
        ]
    },
    "id": 1
}
```

Source

CValueMap::update() dans *ui/include/classes/api/services/CValueMap.php*.

Corrélation

Cette classe est conçue pour fonctionner avec les corrélations.

Références d'objet :

- Corrélation
- Opération de corrélation
- Filtre de corrélation
  - Condition de filtre de corrélation

Méthodes disponibles :

- correlation.create - créer de nouvelles corrélations
- correlation.delete - supprimer des corrélations
- correlation.get - récupérer des corrélations
- correlation.update - mettre à jour des corrélations

Les objets suivants sont directement liés à l'API `correlation`.

Corrélation

L'objet de corrélation possède les propriétés suivantes.

Propriété	Type	Description
correlationid	ID	ID de la corrélation. Comportement de la propriété : - *lecture seule* - *obligatoire* pour les opérations de mise à jour
name	string	Nom de la corrélation. Comportement de la propriété : - *obligatoire* pour les opérations de création
description	string	Description de la corrélation.
status	integer	Indique si la corrélation est activée ou désactivée. Valeurs possibles : 0 - *(par défaut)* activée ; 1 - désactivée.

Opération de corrélation

L'objet d'opération de corrélation définit une opération qui sera effectuée lorsqu'une corrélation est exécutée. Il possède les propriétés suivantes.

Propriété	Type	Description
type	integer	Type d'opération. Valeurs possibles : 0 - fermer les anciens événements ; 1 - fermer le nouvel événement. Comportement de la propriété : - *obligatoire*

Filtre de corrélation

L'objet filtre de corrélation définit un ensemble de conditions qui doivent être remplies pour exécuter les opérations de corrélation configurées. Il possède les propriétés suivantes.

Property	Type	Description
conditions	array	Ensemble de conditions de filtre à utiliser pour filtrer les résultats. Les conditions seront triées dans l'ordre de leur placement dans la formule. Comportement de la propriété: - *required*
evaltype	integer	Méthode d'évaluation des conditions du filtre. Valeurs possibles: 0 - And/Or; 1 - And; 2 - Or; 3 - Expression personnalisée. Comportement de la propriété: - *required*
eval_formula	string	Expression générée qui sera utilisée pour évaluer les conditions du filtre. L'expression contient des ID qui font référence à des conditions de filtre spécifiques via leur `formulaid`. La valeur de `eval_formula` est égale à la valeur de `formula` pour les filtres avec une expression personnalisée. Comportement de la propriété: - *lecture seule*
formula	string	Expression définie par l'utilisateur à utiliser pour évaluer les conditions des filtres avec une expression personnalisée. L'expression doit contenir des ID qui font référence à des conditions de filtre spécifiques via leur `formulaid`. Les ID utilisés dans l'expression doivent correspondre exactement à ceux définis dans les conditions du filtre: aucune condition ne peut rester inutilisée ou omise. Comportement de la propriété: - *required* si `evaltype` est défini sur "custom expression"

Condition de filtre de corrélation

L'objet de condition de filtre de corrélation définit une condition spécifique qui doit être vérifiée avant l'exécution des opérations de corrélation.

Property	Type	Description
type	integer	Type de condition. Valeurs possibles : 0 - ancien tag d'événement ; 1 - nouveau tag d'événement ; 2 - nouveau groupe d'hôtes de l'événement ; 3 - paire de tags d'événement ; 4 - valeur de l'ancien tag d'événement ; 5 - valeur du nouveau tag d'événement. Comportement de la propriété : - *obligatoire*
tag	string	Tag d'événement (ancien ou nouveau). Comportement de la propriété : - *obligatoire* si `type` est défini sur "ancien tag d'événement", "nouveau tag d'événement", "valeur de l'ancien tag d'événement" ou "valeur du nouveau tag d'événement"

Property	Type	Description
groupid	ID	ID du groupe d'hôtes. Comportement de la propriété : - *obligatoire* si `type` est défini sur "nouveau groupe d'hôtes de l'événement"
oldtag	string	Ancien tag d'événement. Comportement de la propriété : - *obligatoire* si `type` est défini sur "paire de tags d'événement"
newtag	string	Ancien tag d'événement. Comportement de la propriété : - *obligatoire* si `type` est défini sur "paire de tags d'événement"
value	string	Valeur du tag d'événement (ancien ou nouveau). Comportement de la propriété : - *obligatoire* si `type` est défini sur "valeur de l'ancien tag d'événement" ou "valeur du nouveau tag d'événement"
formulaid	string	ID unique arbitraire utilisé pour référencer la condition depuis une expression personnalisée. Il ne peut contenir que des lettres majuscules. L'ID doit être défini par l'utilisateur lors de la modification des conditions de filtre, mais sera généré à nouveau lors de leur demande ultérieure.
operator	integer	Opérateur de condition. Comportement de la propriété : - *obligatoire* si `type` est défini sur "nouveau groupe d'hôtes de l'événement", "valeur de l'ancien tag d'événement" ou "valeur du nouveau tag d'événement"

Note:
Pour mieux comprendre comment utiliser les filtres avec différents types d'expressions, consultez les exemples sur les pages des méthodes correlation.get et correlation.create.

Les opérateurs et valeurs suivants sont pris en charge pour chaque type de condition.

Condition	Condition name	Supported operators	Expected value
2	Groupe d'hôtes	=, <>	ID du groupe d'hôtes.
4	Valeur de l'ancien tag d'événement	=, <>, like, not like	string
5	Valeur du nouveau tag d'événement	=, <>, like, not like	string

correlation.create

Description

`object correlation.create(object/array correlations)`

Cette méthode vous permet de créer de nouvelles corrélations.

Note:
Cette méthode est disponible uniquement pour le type d'utilisateur *Super admin*. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres du rôle utilisateur. Consultez User roles pour plus d'informations.

Paramètres

`(object/array)` Corrélations à créer.

En plus des propriétés de corrélation standard, la méthode accepte les paramètres suivants.

Paramètre	Type	Description
operations	array	Opérations de corrélation à créer pour la corrélation. Comportement du paramètre: - *obligatoire*
filter	object	Objet filtre de corrélation pour la corrélation. Comportement du paramètre: - *obligatoire*

Valeurs retournées

`(object)` Renvoie un objet contenant les ID des corrélations créées sous la propriété `correlationids`. L'ordre des ID renvoyés correspond à l'ordre des corrélations transmises.

Exemples

Créer une nouvelle corrélation de tags d'événement

Créez une corrélation en utilisant la méthode d'évaluation `AND/OR` avec une condition et une opération. Par défaut, la corrélation sera activée.

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "correlation.create",
    "params": {
        "name": "new event tag correlation",
        "filter": {
            "evaltype": 0,
            "conditions": [
                {
                    "type": 1,
                    "tag": "ok"
                }
            ]
        },
        "operations": [
            {
                "type": 0
            }
        ]
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "correlationids": [
            "1"
        ]
    },
    "id": 1
}
```

Utilisation d'un filtre d'expression personnalisé

Créez une corrélation qui utilisera une condition de filtre personnalisée. Les ID de formule « A » ou « B » ont été choisis arbitrairement. Le type de condition sera « Groupe d'hôtes » avec l'opérateur « <> ».

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "correlation.create",
    "params": {
        "name": "new host group correlation",
        "description": "a custom description",
        "status": 0,
        "filter": {
            "evaltype": 3,
            "formula": "A or B",
            "conditions": [
                {
                    "type": 2,
                    "operator": 1,
                    "formulaid": "A"
                },
                {
                    "type": 2,
                    "operator": 1,
                    "formulaid": "B"
                }
            ]
        },
        "operations": [
            {
                "type": 1
            }
        ]
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "correlationids": [
            "2"
        ]
    },
    "id": 1
}
```

Voir aussi

- Filtre de corrélation
- Opération de corrélation

Source

CCorrelation::create() dans *ui/include/classes/api/services/CCorrelation.php*.

correlation.delete

Description

`object correlation.delete(array correlationids)`

Cette méthode vous permet de supprimer des corrélations.

> **Note:**
> Cette méthode est disponible uniquement pour le type d'utilisateur *Super admin*. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres du rôle utilisateur. Consultez User roles pour plus d'informations.

Paramètres

`(array)` ID des corrélations à supprimer.

Valeurs retournées

`(object)` Renvoie un objet contenant les identifiants des corrélations supprimées dans la variable `correlationids`.

Exemple

Supprimer plusieurs corrélations

Supprimer deux corrélations.

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "correlation.delete",
    "params": [
        "1",
        "2"
    ],
    "auth": "343baad4f88b4106b9b5961e77437688",
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "correlaionids": [
            "1",
            "2"
        ]
    },
    "id": 1
}
```

Source

CCorrelation::delete() dans *ui/include/classes/api/services/CCorrelation.php*.

correlation.get

Description

`integer/array correlation.get(object parameters)`

Cette méthode permet de récupérer les corrélations selon les paramètres fournis.

> **Note:**
> Cette méthode est disponible pour les utilisateurs de tout type. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres du rôle utilisateur. Consultez Rôles utilisateur pour plus d'informations.

Paramètres

`(object)` Paramètres définissant la sortie souhaitée.

La méthode prend en charge les paramètres suivants.

Paramètre	Type	Description
correlationids	ID/array	Retourner uniquement les corrélations avec les ID donnés.
selectFilter	query	Retourner une propriété `filter` avec les conditions de corrélation.
selectOperations	query	Retourner une propriété `operations` avec les opérations de corrélation.

Paramètre	Type	Description
sortfield	string/array	Trier le résultat selon les propriétés indiquées. Valeurs possibles : `correlationid`, `name`, `status`.
countOutput	boolean	Ces paramètres sont décrits dans les commentaires de référence.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Valeurs retournées

`(integer/array)` Retourne soit:

- un tableau d'objets ;
- le nombre d'objets récupérés, si le paramètre `countOutput` a été utilisé.

Exemples

Récupérer les corrélations

Récupérez toutes les corrélations configurées avec les conditions et les opérations de corrélation. Le filtre utilise le type d'évaluation « and/or », donc la propriété `formula` est vide et `eval_formula` est générée automatiquement.

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "correlation.get",
    "params": {
        "output": "extend",
        "selectOperations": "extend",
        "selectFilter": "extend"
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "correlationid": "1",
            "name": "Correlation 1",
            "description": "",
            "status": "0",
            "filter": {
                "evaltype": "0",
                "formula": "",
                "conditions": [
                    {
                        "type": "3",
                        "oldtag": "error",
                        "newtag": "ok",
                        "formulaid": "A"
                    }
                ],
                "eval_formula": "A"
```

```
            },
            "operations": [
                {
                    "type": "0"
                }
            ]
        }
    ],
    "id": 1
}
```

Voir aussi

- Filtre de corrélation
- Opération de corrélation

Source

CCorrelation::get() dans *ui/include/classes/api/services/CCorrelation.php*.

correlation.update

Description

`object correlation.update(object/array correlations)`

Cette méthode vous permet de mettre à jour des corrélations existantes.

> **Note:**
> Cette méthode est disponible uniquement pour le type d'utilisateur *Super admin*. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres du rôle utilisateur. Consultez User roles pour plus d'informations.

Paramètres

`(object/array)` Propriétés de corrélation à mettre à jour.

La propriété `correlationid` doit être définie pour chaque corrélation, toutes les autres propriétés sont facultatives. Seules les propriétés transmises seront mises à jour, toutes les autres resteront inchangées.

En plus des propriétés de corrélation standard, la méthode accepte les paramètres suivants.

Paramètre	Type	Description
filter	object	Objet filtre de corrélation pour remplacer le filtre actuel.
operations	array	Opérations de corrélation pour remplacer les opérations existantes.

Valeurs retournées

`(object)` Retourne un objet contenant les identifiants des corrélations mises à jour dans la propriété `correlationids`.

Exemples

Désactiver la corrélation

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "correlation.update",
    "params": {
        "correlationid": "1",
        "status": "1"
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "correlationids": [
            "1"
        ]
    },
    "id": 1
}
```

Remplacer les conditions, mais conserver la méthode d'évaluation

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "correlation.update",
    "params": {
        "correlationid": "1",
        "filter": {
            "conditions": [
                {
                    "type": 3,
                    "oldtag": "error",
                    "newtag": "ok"
                }
            ]
        }
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "correlationids": [
            "1"
        ]
    },
    "id": 1
}
```

Voir aussi

- Filtre de corrélation
- Opération de corrélation

Source

CCorrelation::update() dans *ui/include/classes/api/services/CCorrelation.php*.

Déclencheur

Cette classe est conçue pour fonctionner avec les déclencheurs.

Références d'objet :

- Déclencheur
- Tag de déclencheur

Méthodes disponibles :

- trigger.create - créer de nouveaux déclencheurs
- trigger.delete - supprimer des déclencheurs
- trigger.get - récupérer des déclencheurs
- trigger.update - mettre à jour des déclencheurs

Objet déclencheur

Les objets suivants sont directement liés à l'API `trigger`.

Déclencheur

L'objet déclencheur possède les propriétés suivantes.

Propriété	Type	Description
triggerid	ID	ID du déclencheur. Comportement de la propriété : - *lecture seule* - *obligatoire* pour les opérations de mise à jour
description	string	Nom du déclencheur. Comportement de la propriété : - *obligatoire* pour les opérations de création
expression	string	Expression du déclencheur. Les éléments dont `value_type` est défini sur 5 (binaire) ou 6 (JSON) ne peuvent pas être utilisés dans les expressions de déclencheur. Comportement de la propriété : - *obligatoire* pour les opérations de création
event_name	string	Nom de l'événement généré par le déclencheur.
opdata	string	Données opérationnelles.
comments	string	Description supplémentaire du déclencheur.
error	string	Texte d'erreur s'il y a eu des problèmes lors de la mise à jour de l'état du déclencheur. Comportement de la propriété : - *lecture seule*
flags	integer	Origine du déclencheur. Valeurs possibles : 0 - *(par défaut)* un déclencheur simple ; 4 - un déclencheur converti à partir d'un prototype. Comportement de la propriété : - *lecture seule*
lastchange	timestamp	Moment où le déclencheur a changé d'état pour la dernière fois. Comportement de la propriété : - *lecture seule*
priority	integer	Sévérité du déclencheur. Valeurs possibles : 0 - *(par défaut)* non classé ; 1 - information ; 2 - avertissement ; 3 - moyenne ; 4 - élevée ; 5 - désastre.
state	integer	État de l'expression du déclencheur. Valeurs possibles : 0 - *(par défaut)* l'état du déclencheur est à jour ; 1 - l'état actuel du déclencheur est inconnu. Comportement de la propriété : - *lecture seule*

Propriété	Type	Description
status	integer	Indique si le déclencheur est activé ou désactivé. Valeurs possibles : 0 - *(par défaut)* activé ; 1 - désactivé.
templateid	ID	ID du déclencheur de modèle parent. Comportement de la propriété : - *lecture seule*
type	integer	Indique si le déclencheur peut générer plusieurs événements de problème. Valeurs possibles : 0 - *(par défaut)* ne pas générer plusieurs événements ; 1 - générer plusieurs événements.
url	string	URL associée au déclencheur.
url_name	string	Libellé de l'URL associée au déclencheur.
value	integer	Indique si le déclencheur est dans l'état OK ou problème. Valeurs possibles : 0 - *(par défaut)* OK ; 1 - problème. Comportement de la propriété : - *lecture seule*
recovery_mode	integer	Mode de génération de l'événement OK. Valeurs possibles : 0 - *(par défaut)* Expression ; 1 - Expression de récupération ; 2 - Aucun.
recovery_expression	string	Expression de récupération du déclencheur. Les éléments dont `value_type` est défini sur 5 (binaire) ou 6 (JSON) ne peuvent pas être utilisés dans les expressions de récupération du déclencheur.
correlation_mode	integer	Fermeture de l'événement OK. Valeurs possibles : 0 - *(par défaut)* Tous les problèmes ; 1 - Tous les problèmes si les valeurs des tags correspondent.
correlation_tag	string	Tag pour la correspondance.
manual_close	integer	Autoriser la fermeture manuelle. Valeurs possibles : 0 - *(par défaut)* Non ; 1 - Oui.
uuid	string	Identifiant unique universel, utilisé pour lier les déclencheurs importés à ceux qui existent déjà. Généré automatiquement s'il n'est pas fourni. Comportement de la propriété : - *pris en charge* si le déclencheur appartient à un modèle

Balise de déclencheur

L'objet de balise de déclencheur possède les propriétés suivantes.

Propriété	Type	Description
tag	string	Nom de la balise de déclencheur. Comportement de la propriété: - *obligatoire*
value	string	Valeur de la balise de déclencheur. Comportement de la propriété: - *lecture seule*.
object	integer	Type d'objet dont la balise a été héritée. Valeurs possibles : 0 - modèle; 1 - hôte; 3 - élément. Comportement de la propriété: - *lecture seule*.
objectid	ID	ID de l'objet dont la balise a été héritée. Comportement de la propriété: - *lecture seule*.
automatic	integer	Type d'une balise de déclencheur : Valeurs possibles : 0 - *(par défaut)* balise créée par l'utilisateur ; 1 - balise créée par LLD;Comportement de la propriété: - *lecture seule*.

trigger.create

Description

`object trigger.create(object/array triggers)`

Cette méthode permet de créer de nouveaux déclencheurs.

> **Note:**
> Cette méthode est uniquement disponible pour les types d'utilisateur *Admin* et *Super admin*. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres du rôle utilisateur. Consultez Rôles utilisateur pour plus d'informations.

Paramètres

`(object/array)` Déclencheurs à créer.

En plus des propriétés standard du déclencheur, la méthode accepte les paramètres suivants.

Paramètre	Type	Description
dependencies	array	Déclencheurs dont le déclencheur dépend. Les déclencheurs doivent avoir uniquement la propriété `triggerid` définie.
tags	array	Tags de déclencheur.

Valeurs retournées

`(object)` Retourne un objet contenant les identifiants des déclencheurs créés sous la propriété `triggerids`. L'ordre des ID retournés correspond à l'ordre des déclencheurs transmis.

Exemples

Création d'un déclencheur

Créez deux déclencheurs, chacun dépendant d'un autre déclencheur.

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "trigger.create",
    "params": [
        {
            "description": "La charge du processeur est trop élevée sur {HOST.NAME}",
            "expression": "last(/Linux server/system.cpu.load[percpu,avg1])>5",
            "dependencies": [
                {
                    "triggerid": "17367"
                }
            ]
        },
        {
            "description": "Le service s'est arrêté",
            "expression": "length(last(/Linux server/log[/var/log/system,Service .* has stopped]))<>0",
            "dependencies": [
                {
                    "triggerid": "17368"
                }
            ],
            "tags": [
                {
                    "tag": "service",
                    "value": "{{ITEM.VALUE}.regsub(\"Service (.*) has stopped\", \"\\1\")}"
                },
                {
                    "tag": "error",
                    "value": ""
                }
            ]
        }
    ],
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "triggerids": [
            "17369",
            "17370"
        ]
    },
    "id": 1
}
```

Source

CTrigger::create() dans *ui/include/classes/api/services/CTrigger.php*.

trigger.delete

Description

`object trigger.delete(array triggerIds)`

Cette méthode vous permet de supprimer des déclencheurs.

Note:
Cette méthode est disponible uniquement pour les types d'utilisateur *Admin* et *Super admin*. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres des rôles utilisateur. Consultez Rôles utilisateur pour plus d'informations.

Paramètres

`(array)` ID des déclencheurs à supprimer.

Valeurs de retour

`(object)` Renvoie un objet contenant les ID des déclencheurs supprimés dans la propriété `triggerids`.

Exemples

Supprimer plusieurs déclencheurs

Supprimez deux déclencheurs.

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "trigger.delete",
    "params": [
        "12002",
        "12003"
    ],
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "triggerids": [
            "12002",
            "12003"
        ]
    },
    "id": 1
}
```

Source

CTrigger::delete() dans *ui/include/classes/api/services/CTrigger.php*.

trigger.get

Description

`integer/array trigger.get(object parameters)`

La méthode permet de récupérer les déclencheurs selon les paramètres fournis.

Note:
Cette méthode est disponible pour les utilisateurs de tout type. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres du rôle utilisateur. Consultez Rôles utilisateur pour plus d'informations.

Paramètres

`(object)` Paramètres définissant le résultat souhaité.

La méthode prend en charge les paramètres suivants.

Paramètre	Type	Description
triggerids	ID/array	Retourne uniquement les déclencheurs correspondant aux ID indiqués.

Paramètre	Type	Description
groupids	ID/array	Retourne uniquement les déclencheurs appartenant aux hôtes ou aux modèles des groupes d'hôtes ou de modèles indiqués.
templateids	ID/array	Retourne uniquement les déclencheurs appartenant aux modèles indiqués.
hostids	ID/array	Retourne uniquement les déclencheurs appartenant aux hôtes indiqués.
itemids	ID/array	Retourne uniquement les déclencheurs contenant les éléments indiqués.
maintenanceids	ID/array	Retourne uniquement les déclencheurs affectés par les maintenances indiquées.
functions	string/array	Retourne uniquement les déclencheurs utilisant les fonctions indiquées. Consultez la page des fonctions prises en charge pour obtenir la liste des fonctions prises en charge.
group	string	Retourne uniquement les déclencheurs appartenant aux hôtes ou aux modèles du groupe d'hôtes ou de modèles portant le nom indiqué.
host	string	Retourne uniquement les déclencheurs appartenant à l'hôte portant le nom technique indiqué.
inherited	boolean	Si défini sur `true`, retourne uniquement les déclencheurs hérités d'un modèle.
templated	boolean	Si défini sur `true`, retourne uniquement les déclencheurs appartenant à des modèles.
dependent	boolean	Si défini sur `true`, retourne uniquement les déclencheurs ayant des dépendances. Si défini sur `false`, retourne uniquement les déclencheurs n'ayant pas de dépendances.
monitored	flag	Retourne uniquement les déclencheurs activés appartenant à des hôtes surveillés et contenant uniquement des éléments activés.
active	flag	Retourne uniquement les déclencheurs activés appartenant à des hôtes surveillés.
maintenance	boolean	Si défini sur `true`, retourne uniquement les déclencheurs activés appartenant à des hôtes en maintenance.
withUnacknowledgedEvents	flag	Retourne uniquement les déclencheurs ayant des événements non acquittés.
withAcknowledgedEvents	flag	Retourne uniquement les déclencheurs dont tous les événements sont acquittés.
withLastEventUnacknowledged	flag	Retourne uniquement les déclencheurs dont le dernier événement n'est pas acquitté.
skipDependent	flag	Ignore les déclencheurs en état de problème qui dépendent d'autres déclencheurs. Notez que les autres déclencheurs sont ignorés s'ils sont désactivés, s'ils possèdent des éléments désactivés ou si leurs hôtes d'éléments sont désactivés.
lastChangeSince	timestamp	Retourne uniquement les déclencheurs dont l'état a changé après l'heure indiquée.
lastChangeTill	timestamp	Retourne uniquement les déclencheurs dont l'état a changé avant l'heure indiquée.
only_true	flag	Retourne uniquement les déclencheurs ayant été récemment en état de problème (au cours de la période `ok_period`).
min_severity	integer	Retourne uniquement les déclencheurs dont la sévérité est supérieure ou égale à la sévérité indiquée.
evaltype	integer	Méthode d'évaluation des tags. Valeurs possibles : 0 - *(par défaut)* And/Or ; 2 - Or.

Paramètre	Type	Description
tags	array	Retourne uniquement les déclencheurs possédant les tags indiqués. Format : `[{"tag": "<tag>", "value": "<value>", "operator": "<operator>"}, ...]`. Un tableau vide retourne tous les déclencheurs. Valeurs d'opérateur possibles : 0 - *(par défaut)* Contient ; 1 - Est égal à ; 2 - Ne contient pas ; 3 - N'est pas égal à ; 4 - Existe ; 5 - N'existe pas.
expandComment	flag	Développe les macros dans la description du déclencheur.
expandDescription	flag	Développe les macros dans le nom du déclencheur.
expandExpression	flag	Développe les fonctions et les macros dans l'expression du déclencheur.
selectHostGroups	query	Retourne les groupes d'hôtes auxquels appartient le déclencheur dans la propriété `hostgroups`.
selectHosts	query	Retourne les hôtes auxquels appartient le déclencheur dans la propriété `hosts`.
selectItems	query	Retourne les éléments contenus dans le déclencheur dans la propriété `items`.
selectFunctions	query	Retourne les fonctions utilisées dans le déclencheur dans la propriété `functions`. Les objets de fonction représentent les fonctions utilisées dans l'expression du déclencheur et possèdent les propriétés suivantes : `functionid` - `(ID)` ID de la fonction ; `itemid` - `(ID)` ID de l'élément utilisé dans la fonction ; `function` - `(string)` nom de la fonction ; `parameter` - `(string)` paramètre transmis à la fonction. Le paramètre de requête est remplacé par le symbole $ dans la chaîne retournée.
selectDependencies	query	Retourne les déclencheurs dont dépend le déclencheur dans la propriété `dependencies`.
selectDiscoveryData	query	Retourne une propriété `discoveryData` contenant les données de l'objet de découverte du déclencheur. L'objet de découverte du déclencheur relie un déclencheur découvert au prototype de déclencheur à partir duquel il a été découvert. Il possède les propriétés suivantes : `parent_triggerid` - `(ID)` ID du prototype de déclencheur à partir duquel le déclencheur a été créé ; `status` - `(int)` état de découverte du déclencheur : 0 - *(par défaut)* le déclencheur est découvert, 1 - le déclencheur n'est plus découvert ; `ts_delete` - `(timestamp)` heure à laquelle un déclencheur qui n'est plus découvert sera supprimé ; `ts_disable` - `(timestamp)` heure à laquelle un déclencheur qui n'est plus découvert sera désactivé ; `disable_source` - `(int)` indicateur indiquant si le déclencheur a été désactivé par une règle LLD ou manuellement : 0 - *(par défaut)* désactivé automatiquement, 1 - désactivé par une règle LLD.
selectDiscoveryRule	query	Retourne la règle de découverte de bas niveau ayant créé le déclencheur dans la propriété `discoveryRule`.
selectLastEvent	query	Retourne le dernier événement significatif du déclencheur dans la propriété `lastEvent`.
selectTags	query	Retourne les tags du déclencheur dans la propriété `tags`.

Paramètre	Type	Description
inheritedTags	boolean	Retourne les déclencheurs possédant les `tags` indiqués également dans le modèle, l'hôte ou les modèles liés. Valeurs possibles : `true` - le modèle, l'hôte ou les modèles liés doivent également posséder les tags indiqués ; `false` - *(par défaut)* les tags du modèle, de l'hôte ou des modèles liés sont ignorés.
selectInheritedTags	query	Retourne une propriété `inheritedTags` contenant les tags hérités des modèles, des modèles liés et des hôtes, ainsi que des éléments référencés dans les expressions du déclencheur ou de récupération.
selectTemplateGroups	query	Retourne les groupes de modèles auxquels appartient le déclencheur dans la propriété `templategroups`.
filter	object	Retourne uniquement les résultats correspondant exactement au filtre indiqué. Accepte un objet dont les clés sont des noms de propriétés et dont les valeurs sont soit une valeur unique, soit un tableau de valeurs à comparer. Ne prend pas en charge les propriétés du type de données `text`. Prend en charge les propriétés supplémentaires suivantes : `host` - nom technique de l'hôte auquel appartient le déclencheur ; `hostid` - ID de l'hôte auquel appartient le déclencheur.
limitSelects	integer	Limite le nombre d'enregistrements retournés par les sous-sélections. S'applique aux sous-sélections suivantes : `selectHosts` - les résultats seront triés par `host`.
sortfield	string/array	Trie le résultat selon les propriétés indiquées. Valeurs possibles : `triggerid`, `description`, `status`, `priority`, `lastchange`, `hostname`.
countOutput	boolean	Ces paramètres sont décrits dans le commentaire de référence.
editable	boolean	
excludeSearch	boolean	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	
selectTriggerDiscovery	query	Retourne l'objet de découverte du déclencheur dans la propriété `triggerDiscovery`. Les objets de découverte du déclencheur relient le déclencheur au prototype de déclencheur à partir duquel il a été créé. Cette requête est **obsolète** ; veuillez utiliser `selectDiscoveryData` à la place.

Valeurs de retour

`(integer/array)` Renvoie soit :

- un tableau d'objets ;
- le nombre d'objets récupérés, si le paramètre `countOutput` a été utilisé.

Exemples

Récupération des données par ID de déclencheur

Récupérez toutes les données et les fonctions utilisées dans le déclencheur « 14062 ».

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "trigger.get",
    "params": {
        "triggerids": "14062",
        "output": "extend",
        "selectFunctions": "extend"
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "triggerid": "14062",
            "expression": "{13513}<10m",
            "description": "{HOST.NAME} has been restarted (uptime < 10m)",
            "url": "",
            "status": "0",
            "value": "0",
            "priority": "2",
            "lastchange": "0",
            "comments": "The host uptime is less than 10 minutes",
            "error": "",
            "templateid": "10016",
            "type": "0",
            "state": "0",
            "flags": "0",
            "recovery_mode": "0",
            "recovery_expression": "",
            "correlation_mode": "0",
            "correlation_tag": "",
            "manual_close": "0",
            "opdata": "",
            "event_name": "",
            "uuid": "",
            "url_name": "",
            "functions": [
                {
                    "functionid": "13513",
                    "itemid": "24350",
                    "triggerid": "14062",
                    "parameter": "$",
                    "function": "last"
                }
            ]
        }
    ],
    "id": 1
}
```

Récupération des déclencheurs à l'état de problème

Récupérez l'ID, le nom et la sévérité de tous les déclencheurs à l'état de problème, puis triez-les par sévérité dans l'ordre décroissant.

Requête:

```
{
    "jsonrpc": "2.0",
```

```
    "method": "trigger.get",
    "params": {
        "output": [
            "triggerid",
            "description",
            "priority"
        ],
        "filter": {
            "value": 1
        },
        "sortfield": "priority",
        "sortorder": "DESC"
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "triggerid": "13907",
            "description": "Zabbix self-monitoring processes < 100% busy",
            "priority": "4"
        },
        {
            "triggerid": "13824",
            "description": "Zabbix discoverer processes more than 75% busy",
            "priority": "3"
        }
    ],
    "id": 1
}
```

Récupération d'un déclencheur spécifique avec des tags

Récupérez un déclencheur spécifique avec des tags.

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "trigger.get",
    "params": {
        "output": [
            "triggerid",
            "description"
        ],
        "selectTags": "extend",
        "triggerids": [
            "17578"
        ]
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "triggerid": "17370",
            "description": "Service status",
            "tags": [
```

```
                {
                    "tag": "service",
                    "value": "{{ITEM.VALUE}.regsub(\"Service (.*) has stopped\", \"\\1\")}",
                    "automatic": "1"
                },
                {
                    "tag": "error",
                    "value": "",
                    "automatic": "1"
                }
            ]
        }
    ],
    "id": 1
}
```

Voir aussi

- Règle de découverte
- Élément
- Hôte
- Groupe d'hôtes
- Groupe de modèles

Source

CTrigger::get() dans *ui/include/classes/api/services/CTrigger.php*.

trigger.update

Description

`object trigger.update(object/array triggers)`

Cette méthode vous permet de mettre à jour des déclencheurs existants.

Note:
Cette méthode est disponible uniquement pour les types d'utilisateur *Admin* et *Super admin*. Les autorisations d'appel de cette méthode peuvent être révoquées dans les paramètres des rôles utilisateur. Consultez Rôles utilisateur pour plus d'informations.

Paramètres

`(object/array)` Propriétés du déclencheur à mettre à jour.

La propriété `triggerid` doit être définie pour chaque déclencheur, toutes les autres propriétés sont facultatives. Seules les propriétés transmises seront mises à jour, toutes les autres resteront inchangées.

En plus des propriétés standard du déclencheur, la méthode accepte les paramètres suivants.

Parameter	Type	Description
dependencies	array	Déclencheurs dont le déclencheur dépend. Les déclencheurs doivent avoir uniquement la propriété `triggerid` définie.
tags	array	Tags de déclencheur.

Valeurs retournées

`(object)` Retourne un objet contenant les identifiants des déclencheurs mis à jour sous la propriété `triggerids`.

Exemples

Activation d'un déclencheur

Activez un déclencheur, c'est-à-dire définissez son statut sur « 0 ».

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "trigger.update",
    "params": {
        "triggerid": "13938",
        "status": 0
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "triggerids": [
            "13938"
        ]
    },
    "id": 1
}
```

Remplacement des tags de déclencheur

Remplacez les tags du déclencheur.

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "trigger.update",
    "params": {
        "triggerid": "13938",
        "tags": [
            {
                "tag": "service",
                "value": "{{ITEM.VALUE}.regsub(\"Service (.*) has stopped\", \"\\1\")}"
            },
            {
                "tag": "error",
                "value": ""
            }
        ]
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "triggerids": [
            "13938"
        ]
    },
    "id": 1
}
```

Remplacement des dépendances

Remplace les dépendances du déclencheur.

Requête:

```
{
    "jsonrpc": "2.0",
```

```
    "method": "trigger.update",
    "params": {
        "triggerid": "22713",
        "dependencies": [
            {
                "triggerid": "22712"
            },
            {
                "triggerid": "22772"
            }
        ]
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "triggerids": [
            "22713"
        ]
    },
    "id": 1
}
```

Source

CTrigger::update() dans *ui/include/classes/api/services/CTrigger.php*.

Enregistrement automatique

Cette classe est conçue pour fonctionner avec l'enregistrement automatique.

Références d'objet :

- Autoregistration

Méthodes disponibles :

- autoregistration.get - récupérer l'enregistrement automatique
- autoregistration.update - mettre à jour l'enregistrement automatique

Objet Enregistrement automatique

Les objets suivants sont directement liés à l'API `autoregistration`.

Enregistrement automatique

L'objet d'enregistrement automatique possède les propriétés suivantes.

Propriété	Type	Description
tls_accept	integer	Type de connexions entrantes autorisées pour l'enregistrement automatique. Valeurs possibles : 1 - autoriser les connexions non chiffrées ; 2 - autoriser TLS avec PSK ; 3 - autoriser à la fois les connexions non chiffrées et TLS avec PSK.

Propriété	Type	Description
tls_psk_identity	string	Identité PSK ; doit être associée à une seule PSK (dans autoregistration, hosts et proxies). N'incluez pas d'informations sensibles dans l'identité PSK, car elle est envoyée sans chiffrement sur le réseau afin d'informer le destinataire de la PSK à utiliser. Comportement de la propriété : - *écriture seule*
tls_psk	string	Clé pré-partagée (PSK) ; doit comporter au moins 32 chiffres hexadécimaux. Comportement de la propriété : - *écriture seule*

autoregistration.get

Description

`object autoregistration.get(object parameters)`

La méthode permet de récupérer un objet d'autoregistration selon les paramètres fournis.

> **Note:**
> Cette méthode est disponible uniquement pour le type d'utilisateur *Super admin*. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres du rôle utilisateur. Consultez User roles pour plus d'informations.

Paramètres

`(object)` Paramètres définissant la sortie souhaitée.

La méthode ne prend en charge qu'un seul paramètre.

Paramètre	Type	Description
output	query	Ce paramètre est décrit dans le commentaire de référence.

Valeurs retournées

`(object)` Retourne l'objet enregistrement automatique

Exemples

Requête:

```
{
 "jsonrpc": "2.0",
 "method": "autoregistration.get",
 "params": {
 "output": "extend"
 },
 "auth": "038e1d7b1735c6a5436ee9eae095879e",
 "id": 1
}
```

Réponse :

```
{
 "jsonrpc": "2.0",
 "result": {
 "tls_accept": "3"
 },
 "id": 1
}
```

Source

CAutoregistration::get() dans *ui/include/classes/api/services/CAutoregistration.php*.

autoregistration.update

Description

`object autoregistration.update(object autoregistration)`

Cette méthode vous permet de mettre à jour une autoréservation existante.

> **Note:**
> Cette méthode est disponible uniquement pour le type d'utilisateur *Super admin*. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres des rôles utilisateur. Voir User roles pour plus d'informations.

Paramètres

`(object)` Propriétés d'enregistrement automatique à mettre à jour.

Valeurs de retour

`(boolean )` Renvoie la valeur booléenne true en résultat en cas de mise à jour réussie.

Exemples

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "autoregistration.update",
    "params": {
        "tls_accept": "3",
        "tls_psk_identity": "PSK 001",
        "tls_psk": "11111595725ac58dd977beef14b97461a7c1045b9a1c923453302c5473193478"
    },
    "auth": "038e1d7b1735c6a5436ee9eae095879e",
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": true,
    "id": 1
}
```

Source

CAutoregistration::update() dans *ui/include/classes/api/services/CAutoregistration.php*.

Expression régulière

Cette classe est conçue pour fonctionner avec des expressions régulières globales.

Références d'objet :

- Expression régulière
- Expressions

Méthodes disponibles :

- regexp.create - créer de nouvelles expressions régulières
- regexp.delete - supprimer des expressions régulières
- regexp.get - récupérer des expressions régulières
- regexp.update - mettre à jour des expressions régulières

Objet d'expression régulière

Les objets suivants sont directement liés à l'API `regexp`.

Expression régulière

L'objet d'expression régulière globale possède les propriétés suivantes.

Property	Type	Description
regexpid	ID	ID de l'expression régulière. Property behavior: - *read-only* - *required* for update operations.
name	string	Nom de l'expression régulière. Property behavior: - *required* for create operations.
description	text	Description de l'expression régulière.
test_string	string	Chaîne de test.

Expressions

L'objet expressions possède les propriétés suivantes.

Property	Type	Description
expression	string	Expression régulière. Comportement de la propriété: - *required*
expression_type	integer	Type d'expression régulière. Valeurs possibles : 0 - Contient la chaîne ; 1 - Contient n'importe quelle sous-chaîne de la liste ; 2 - Ne contient pas la chaîne ; 3 - Correspond à l'expression régulière ; 4 - Ne correspond pas à l'expression régulière. Comportement de la propriété: - *required*
exp_delimiter	string	Délimiteur d'expression. Valeur par défaut : `","`. Valeurs possibles : `","` ou `"."`, ou `"/"`. Comportement de la propriété: - *supported* si `expression_type` est défini sur *Contient n'importe quelle sous-chaîne de la liste*.
case_sensitive	integer	Sensibilité à la casse. Valeur par défaut : 0. Valeurs possibles : 0 - Insensible à la casse ; 1 - Sensible à la casse.

regexp.create

Description

`object regexp.create(object/array regularExpressions)`

Cette méthode vous permet de créer de nouvelles expressions régulières globales.

> **Note:**
> Cette méthode est disponible uniquement pour les types d'utilisateurs *Super admin*. Les autorisations d'appel de cette méthode peuvent être révoquées dans les paramètres des rôles utilisateur. Consultez User roles pour plus d'informations.

Paramètres

`(object/array)` Expressions régulières à créer.

En plus des propriétés standard, la méthode accepte les paramètres suivants.

Paramètre	Type	Description
expressions	array	Options des expressions. Comportement des paramètres: - *obligatoire*

Valeurs retournées

`(object)` Retourne un objet contenant les identifiants des expressions régulières créées sous la propriété `regexpids`.

Exemples

Création d'une nouvelle expression régulière globale.

Request :

```
{
    "jsonrpc": "2.0",
    "method": "regexp.create",
    "params": {
        "name": "Storage devices for SNMP discovery",
        "description": "Expression régulière globale utilisée pour découvrir et identifier les ressources
        "test_string": "/boot",
        "expressions": [
            {
                "expression": "^(Physical memory|Virtual memory|Memory buffers|Cached memory|Swap space)$"
                "expression_type": "4",
                "case_sensitive": "1"
            }
        ]
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "regexpids": [
            "9"
        ]
    },
    "id": 1
}
```

Source

CRegexp::create() dans *ui/include/classes/api/services/CRegexp.php*.

regexp.delete

Description

`object regexp.delete(array regexpids)`

Cette méthode permet de supprimer des expressions régulières globales.

> **Note:**
> Cette méthode est disponible uniquement pour les types d'utilisateurs *Super admin*. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres des rôles utilisateur. Voir User roles pour plus d'informations.

Paramètres

`(array)` Identifiants des expressions régulières à supprimer.

Valeurs de retour

`(object)` Renvoie un objet contenant les ID des expressions régulières supprimées dans la propriété `regexpids`.

Exemples

Suppression de plusieurs expressions régulières globales.

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "regexp.delete",
    "params": [
        "16",
        "17"
    ],
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "regexpids": [
            "16",
            "17"
        ]
    },
    "id": 1
}
```

Source

CRegexp::delete() dans *ui/include/classes/api/services/CRegexp.php*.

regexp.get

Description

`integer/array regexp.get(object parameters)`

Cette méthode permet de récupérer les expressions régulières globales selon les paramètres fournis.

> **Note:**
> Cette méthode est disponible uniquement pour les *Super Admin*. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres des rôles utilisateur. Consultez Rôles utilisateur pour plus d'informations.

Paramètres

`(object)` Paramètres définissant la sortie souhaitée.

La méthode prend en charge les paramètres suivants.

Paramètre	Type	Description
regexpids	ID/array	Retourner uniquement les expressions régulières avec les ID donnés.
selectExpressions	query	Retourner une propriété `expressions`.
sortfield	string/array	Trier le résultat selon les propriétés indiquées. Valeurs possibles : `regexpid`, `name`.
countOutput	boolean	Ces paramètres sont décrits dans le commentaire de référence.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Valeurs retournées

`(integer/array)` Retourne soit :

- un tableau d'objets ;
- le nombre d'objets récupérés, si le paramètre `countOutput` a été utilisé.

Exemples

Récupération des expressions régulières globales.

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "regexp.get",
    "params": {
        "output": ["regexpid","name","description"],
        "selectExpressions": ["expression","expression_type"],
        "regexpids": [1,2],
        "preservekeys": true
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "1": {
            "regexpid": "1",
            "name": "File systems for discovery",
            "description": "",
            "expressions": [
                {
                    "expression": "^(btrfs|ext2|ext3|ext4|reiser|xfs|ffs|ufs|jfs|jfs2|vxfs|hfs|apfs|refs|n
                    "expression_type": "3"
                }
            ]
        },
        "2": {
            "regexpid": "2",
            "name": "Network interfaces for discovery",
            "description": "",
            "expressions": [
```

```
                {
                    "expression": "^Software Loopback Interface",
                    "expression_type": "4"
                },
                {
                    "expression": "^(In)?[Ll]oop[Bb]ack[0-9._]*$",
                    "expression_type": "4"
                },
                {
                    "expression": "^NULL[0-9.]*$",
                    "expression_type": "4"
                },
                {
                    "expression": "^[Ll]o[0-9.]*$",
                    "expression_type": "4"
                },
                {
                    "expression": "^[Ss]ystem$",
                    "expression_type": "4"
                },
                {
                    "expression": "^Nu[0-9.]*$",
                    "expression_type": "4"
                }
            ]
        }
    },
    "id": 1
}
```


Source

CRegexp::get() dans *ui/include/classes/api/services/CRegexp.php*.

regexp.update

Description

`object regexp.update(object/array regularExpressions)`

Cette méthode vous permet de mettre à jour des expressions régulières globales existantes.

> **Note:**
> Cette méthode est disponible uniquement pour les types d'utilisateur *Super admin*. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres des rôles utilisateur. Consultez User roles pour plus d'informations.

Paramètres

`(object/array)` Propriétés de l'expression régulière à mettre à jour.

La propriété `regexpid` doit être définie pour chaque objet, toutes les autres propriétés sont facultatives. Seules les propriétés transmises seront mises à jour, toutes les autres resteront inchangées.

En plus des propriétés standard, la méthode accepte les paramètres suivants.

Paramètre	Type	Description
expressions	array	Options des expressions.

Valeurs retournées

`(object)` Retourne un objet contenant les identifiants des expressions régulières mis à jour sous la propriété `regexpids`.

Exemples

Mise à jour de l'expression régulière globale pour la découverte des systèmes de fichiers.

Request:

```
{
    "jsonrpc": "2.0",
    "method": "regexp.update",
    "params": {
        "regexpid": "1",
        "name": "File systems for discovery",
        "description": "Expression régulière globale utilisée pour découvrir et identifier les systèmes de
        "test_string": "",
        "expressions": [
            {
                "expression": "^(btrfs|ext2|ext3|ext4|reiser|xfs|ffs|ufs|jfs|jfs2|vxfs|hfs|apfs|refs|zfs)$
                "expression_type": "3",
                "case_sensitive": "0"
            },
            {
                "expression": "^(ntfs|fat32|fat16)$",
                "expression_type": "3",
                "case_sensitive": "0"
            }
        ]
    },
    "id": 1
}
```

Response:

```
{
    "jsonrpc": "2.0",
    "result": {
        "regexpids": [
            "1"
        ]
    },
    "id": 1
}
```

Source

CRegexp::update() dans *ui/include/classes/api/services/CRegexp.php*.

Graphique

Cette classe est conçue pour fonctionner avec les graphiques.

Références d'objet :

- Graph

Méthodes disponibles :

- graph.create - créer de nouveaux graphiques
- graph.delete - supprimer des graphiques
- graph.get - récupérer des graphiques
- graph.update - mettre à jour des graphiques

Objet graphique

Les objets suivants sont directement liés à l'API `graph`.

Graphique

L'objet graphique possède les propriétés suivantes.

Propriété	Type	Description
graphid	ID	ID du graphique. Comportement de la propriété: - *lecture seule* - *obligatoire* pour les opérations de mise à jour
height	integer	Hauteur du graphique en pixels. Comportement de la propriété: - *obligatoire* pour les opérations de création
name	string	Nom du graphique. Comportement de la propriété: - *obligatoire* pour les opérations de création
width	integer	Largeur du graphique en pixels. Comportement de la propriété: - *obligatoire* pour les opérations de création
flags	integer	Origine du graphique. Valeurs possibles : 0 - *(par défaut)* un graphique simple ; 4 - un graphique converti à partir d'un prototype. Comportement de la propriété: - *lecture seule*
graphtype	integer	Type de disposition du graphique. Valeurs possibles : 0 - *(par défaut)* normal ; 1 - empilé ; 2 - camembert ; 3 - éclaté.
percent_left	float	Percentile gauche. Par défaut : 0.
percent_right	float	Percentile droit. Par défaut : 0.
show_3d	integer	Indique s'il faut afficher les graphiques en camembert et éclatés en 3D. Valeurs possibles : 0 - *(par défaut)* affichage en 2D ; 1 - affichage en 3D.
show_legend	integer	Indique s'il faut afficher la légende sur le graphique. Valeurs possibles : 0 - masquer ; 1 - *(par défaut)* afficher.
show_work_period	integer	Indique s'il faut afficher le temps de travail sur le graphique. Valeurs possibles : 0 - masquer ; 1 - *(par défaut)* afficher.
show_triggers	integer	Indique s'il faut afficher la ligne du déclencheur sur le graphique. Valeurs possibles : 0 - masquer ; 1 - *(par défaut)* afficher.
templateid	ID	ID du graphique du modèle parent. Comportement de la propriété: - *lecture seule*

Propriété	Type	Description
yaxismax	float	Valeur maximale fixe pour l'axe Y. Par défaut : 100.
yaxismin	float	Valeur minimale fixe pour l'axe Y. Par défaut : 0.
ymax_itemid	ID	ID de l'élément utilisé comme valeur maximale pour l'axe Y. Si un utilisateur n'a pas accès à l'élément spécifié, le graphique est affiché comme si `ymax_type` était défini sur "calculated".
ymax_type	integer	Méthode de calcul de la valeur maximale pour l'axe Y. Valeurs possibles : 0 - *(par défaut)* calculated ; 1 - fixe ; 2 - élément.
ymin_itemid	ID	ID de l'élément utilisé comme valeur minimale pour l'axe Y. Si un utilisateur n'a pas accès à l'élément spécifié, le graphique est affiché comme si `ymin_type` était défini sur "calculated".
ymin_type	integer	Méthode de calcul de la valeur minimale pour l'axe Y. Valeurs possibles : 0 - *(par défaut)* calculated ; 1 - fixe ; 2 - élément.
uuid	string	Identifiant unique universel, utilisé pour lier les graphiques importés à ceux qui existent déjà. Généré automatiquement s'il n'est pas fourni. Comportement de la propriété: - *pris en charge* si le graphique appartient à un modèle

graph.create

Description

`object graph.create(object/array graphs)`

Cette méthode permet de créer de nouveaux graphiques.

Note:
Cette méthode est disponible uniquement pour les types d'utilisateur *Admin* et *Super admin*. Les autorisations d'appel de cette méthode peuvent être révoquées dans les paramètres des rôles utilisateur. Consultez Rôles utilisateur pour plus d'informations.

Paramètres

`(object/array)` Graphiques à créer.

En plus des propriétés de graphique standard, la méthode accepte les paramètres suivants.

Paramètre	Type	Description
gitems	array	Éléments de graphique à créer pour le graphique. Comportement du paramètre: - *obligatoire*

Valeurs retournées

`(object)` Retourne un objet contenant les identifiants des graphiques créés sous la propriété `graphids`. L'ordre des ID retournés correspond à l'ordre des graphiques transmis.

Exemples

Création d'un graphique

Créez un graphique avec deux éléments.

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "graph.create",
    "params": {
        "name": "MySQL bandwidth",
        "width": 900,
        "height": 200,
        "gitems": [
            {
                "itemid": "22828",
                "color": "00AA00"
            },
            {
                "itemid": "22829",
                "color": "3333FF"
            }
        ]
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "graphids": [
            "652"
        ]
    },
    "id": 1
}
```

Voir aussi

- Élément de graphique

Source

CGraph::create() dans *ui/include/classes/api/services/CGraph.php*.

graph.delete

Description

`object graph.delete(array graphIds)`

Cette méthode vous permet de supprimer des graphiques.

> **Note:**
> Cette méthode est disponible uniquement pour les types d'utilisateur *Admin* et *Super admin*. Les autorisations d'appel de cette méthode peuvent être révoquées dans les paramètres des rôles utilisateur. Consultez Rôles utilisateur pour plus d'informations.

Paramètres

`(array)` ID des graphiques à supprimer.

Valeurs retournées

`(object)` Retourne un objet contenant les identifiants des graphiques supprimés sous la propriété `graphids`.

Exemples

Suppression de plusieurs graphiques

Supprimer deux graphiques.

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "graph.delete",
    "params": [
        "652",
        "653"
    ],
    "auth": "3a57200802b24cda67c4e4010b50c065",
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "graphids": [
            "652",
            "653"
        ]
    },
    "id": 1
}
```

Source

CGraph::delete() dans *ui/include/classes/api/services/CGraph.php*.

graph.get

Description

`integer/array graph.get(object parameters)`

Cette méthode permet de récupérer les graphiques selon les paramètres fournis.

> **Note:**
> Cette méthode est disponible pour les utilisateurs de tout type. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres des rôles utilisateur. Voir Rôles utilisateur pour plus d'informations.

Paramètres

`(object)` Paramètres définissant la sortie souhaitée.

La méthode prend en charge les paramètres suivants.

Paramètre	Type	Description
graphids	ID/array	Retourner uniquement les graphiques avec les ID donnés.
groupids	ID/array	Retourner uniquement les graphiques qui appartiennent aux hôtes ou aux modèles des groupes d'hôtes ou groupes de modèles donnés.
templateids	ID/array	Retourner uniquement les graphiques qui appartiennent aux modèles donnés.
hostids	ID/array	Retourner uniquement les graphiques qui appartiennent aux hôtes donnés.
itemids	ID/array	Retourner uniquement les graphiques qui contiennent les éléments donnés.
templated	boolean	Si défini sur `true`, retourner uniquement les graphiques qui appartiennent à des modèles.

Paramètre	Type	Description
inherited	boolean	Si défini sur `true`, retourner uniquement les graphiques hérités d'un modèle.
expandName	flag	Développer les macros dans le nom du graphique.
selectDiscoveryData	query	Retourner une propriété `discoveryData` avec les données de l'objet de découverte du graphique. L'objet de découverte du graphique relie un graphique découvert à un prototype de graphique à partir duquel il a été découvert. Il possède les propriétés suivantes : `parent_graphid` - `(ID)` ID du prototype de graphique à partir duquel le graphique a été créé ; `status` - `(int)` état de découverte du graphique : 0 - *(par défaut)* le graphique est découvert, 1 - le graphique n'est plus découvert ; `ts_delete` - `(timestamp)` heure à laquelle un graphique qui n'est plus découvert sera supprimé.
selectDiscoveryRule	query	Retourner une propriété `discoveryRule` avec la règle de découverte de bas niveau qui a créé le graphique.
selectHostGroups	query	Retourner une propriété `hostgroups` avec les groupes d'hôtes auxquels le graphique appartient.
selectTemplateGroups	query	Retourner une propriété `templategroups` avec les groupes de modèles auxquels le graphique appartient.
selectTemplates	query	Retourner une propriété `templates` avec les modèles auxquels le graphique appartient.
selectHosts	query	Retourner une propriété `hosts` avec les hôtes auxquels le graphique appartient.
selectItems	query	Retourner une propriété `items` avec les éléments utilisés dans le graphique.
selectGraphItems	query	Retourner une propriété `gitems` avec les éléments utilisés dans le graphique.
filter	object	Retourner uniquement les résultats qui correspondent exactement au filtre donné. Accepte un objet, où les clés sont des noms de propriété et les valeurs sont soit une valeur unique, soit un tableau de valeurs à comparer. Ne prend pas en charge les propriétés de type de données `text` data type. Prend en charge des propriétés supplémentaires : `host` - nom technique de l'hôte auquel le graphique appartient ; `hostid` - ID de l'hôte auquel le graphique appartient.
sortfield	string/array	Trier le résultat selon les propriétés données. Valeurs possibles : `graphid`, `name`, `graphtype`.
countOutput	boolean	Ces paramètres sont décrits dans le commentaire de référence.
editable	boolean	
excludeSearch	boolean	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Paramètre	Type	Description
selectGraphDiscovery	query	Retourner une propriété `graphDiscovery` avec l'objet de découverte du graphique. Les objets de découverte du graphique relient le graphique à un prototype de graphique à partir duquel il a été créé. Cette requête est **obsolète**, veuillez utiliser `selectDiscoveryData` à la place.

Valeurs retournées

`(entier/tableau)` Renvoie soit :

- un tableau d'objets ;
- le nombre d'objets récupérés, si le paramètre `countOutput` a été utilisé.

Exemples

Récupération des graphiques des hôtes

Récupérez tous les graphiques de l'hôte "10107" et triez-les par nom.

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "graph.get",
    "params": {
        "output": "extend",
        "hostids": 10107,
        "sortfield": "name"
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "graphid": "612",
            "name": "CPU jumps",
            "width": "900",
            "height": "200",
            "yaxismin": "0",
            "yaxismax": "100",
            "templateid": "439",
            "show_work_period": "1",
            "show_triggers": "1",
            "graphtype": "0",
            "show_legend": "1",
            "show_3d": "0",
            "percent_left": "0",
            "percent_right": "0",
            "ymin_type": "0",
            "ymax_type": "0",
            "ymin_itemid": "0",
            "ymax_itemid": "0",
            "flags": "0"
        },
        {
            "graphid": "613",
            "name": "CPU load",
            "width": "900",
            "height": "200",
            "yaxismin": "0",
```

```
            "yaxismax": "100",
            "templateid": "433",
            "show_work_period": "1",
            "show_triggers": "1",
            "graphtype": "0",
            "show_legend": "1",
            "show_3d": "0",
            "percent_left": "0",
            "percent_right": "0",
            "ymin_type": "1",
            "ymax_type": "0",
            "ymin_itemid": "0",
            "ymax_itemid": "0",
            "flags": "0"
        },
        {
            "graphid": "614",
            "name": "CPU utilization",
            "width": "900",
            "height": "200",
            "yaxismin": "0",
            "yaxismax": "100",
            "templateid": "387",
            "show_work_period": "1",
            "show_triggers": "0",
            "graphtype": "1",
            "show_legend": "1",
            "show_3d": "0",
            "percent_left": "0",
            "percent_right": "0",
            "ymin_type": "1",
            "ymax_type": "1",
            "ymin_itemid": "0",
            "ymax_itemid": "0",
            "flags": "0"
        },
        {
            "graphid": "645",
            "name": "Disk space usage /",
            "width": "600",
            "height": "340",
            "yaxismin": "0",
            "yaxismax": "0",
            "templateid": "0",
            "show_work_period": "0",
            "show_triggers": "0",
            "graphtype": "2",
            "show_legend": "1",
            "show_3d": "1",
            "percent_left": "0",
            "percent_right": "0",
            "ymin_type": "0",
            "ymax_type": "0",
            "ymin_itemid": "0",
            "ymax_itemid": "0",
            "flags": "4"
        }
    ],
    "id": 1
}
```


Voir aussi

- Règle de découverte
- Élément de graphique
- Élément
- Hôte
- Groupe d'hôtes
- Modèle
- Groupe de modèles

Source

CGraph::get() dans *ui/include/classes/api/services/CGraph.php*.

graph.update

Description

`object graph.update(object/array graphs)`

Cette méthode permet de mettre à jour des graphiques existants.

> **Note:**
> Cette méthode est disponible uniquement pour les types d'utilisateur *Admin* et *Super admin*. Les autorisations d'appel de cette méthode peuvent être révoquées dans les paramètres des rôles utilisateur. Consultez Rôles utilisateur pour plus d'informations.

Paramètres

`(object/array)` Propriétés du graphique à mettre à jour.

La propriété `graphid` doit être définie pour chaque graphique, toutes les autres propriétés sont facultatives. Seules les propriétés transmises seront mises à jour, toutes les autres resteront inchangées.

En plus des propriétés de graphique standard, la méthode accepte les paramètres suivants.

Paramètre	Type	Description
gitems	array	Éléments de graphique pour remplacer les éléments de graphique existants. Si un élément de graphique a la propriété `gitemid` définie, il sera mis à jour ; sinon, un nouvel élément de graphique sera créé.

Valeurs retournées

`(object)` Retourne un objet contenant les identifiants des graphiques mis à jour sous la propriété `graphids`.

Exemples

Définition du maximum de l'échelle Y

Définissez le maximum de l'échelle Y sur une valeur fixe de 100.

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "graph.update",
    "params": {
        "graphid": "439",
        "ymax_type": 1,
        "yaxismax": 100
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "graphids": [
```

```
            "439"
        ]
    },
    "id": 1
}
```

Source

CGraph::update() dans *ui/include/classes/api/services/CGraph.php*.

Groupe d'hôtes

Cette classe est conçue pour fonctionner avec les groupes d'hôtes.

Références d'objet :

- Groupe d'hôtes

Méthodes disponibles :

- hostgroup.create - créer de nouveaux groupes d'hôtes
- hostgroup.delete - supprimer des groupes d'hôtes
- hostgroup.get - récupérer des groupes d'hôtes
- hostgroup.massadd - ajouter des objets associés à des groupes d'hôtes
- hostgroup.massremove - supprimer des objets associés de groupes d'hôtes
- hostgroup.propagate - propager les permissions et les filtres de tags aux sous-groupes des groupes d'hôtes
- hostgroup.update - mettre à jour des groupes d'hôtes

Objet de groupe d'hôtes

Les objets suivants sont directement liés à l'API `hostgroup`.

Groupe d'hôtes

L'objet groupe d'hôtes possède les propriétés suivantes.

Propriété	Type	Description
groupid	ID	ID du groupe d'hôtes. Comportement de la propriété: - *lecture seule* - *obligatoire* pour les opérations de mise à jour
name	string	Nom du groupe d'hôtes. Comportement de la propriété: - *obligatoire* pour les opérations de création
flags	integer	Origine du groupe d'hôtes. Valeurs possibles : 0 - un groupe d'hôtes simple ; 4 - un groupe d'hôtes découvert. Comportement de la propriété: - *lecture seule*
uuid	string	Identifiant unique universel, utilisé pour lier les groupes d'hôtes importés à ceux qui existent déjà. Généré automatiquement s'il n'est pas fourni.

hostgroup.create

Description

`object hostgroup.create(object/array hostGroups)`

Cette méthode permet de créer de nouveaux groupes d'hôtes.

> **Note:**
> Cette méthode est disponible uniquement pour le type d'utilisateur *Super admin*. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres des rôles utilisateur. Voir User roles pour plus d'informations.

Paramètres

`(object/array)` Groupes d'hôtes à créer.

La méthode accepte des groupes d'hôtes avec les propriétés standard des groupes d'hôtes.

Valeurs retournées

`(object)` Renvoie un objet contenant les identifiants des groupes d'hôtes créés sous la propriété `groupids`. L'ordre des identifiants retournés correspond à l'ordre des groupes d'hôtes transmis.

Exemples

Création d'un groupe d'hôtes

Créez un groupe d'hôtes nommé « Linux servers ».

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "hostgroup.create",
    "params": {
        "name": "Linux servers"
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "groupids": [
            "107819"
        ]
    },
    "id": 1
}
```

Source

CHostGroup::create() dans *ui/include/classes/api/services/CHostGroup.php*.

hostgroup.delete

Description

`object hostgroup.delete(array hostGroupIds)`

Cette méthode vous permet de supprimer des groupes d'hôtes.

Un groupe d'hôtes ne peut pas être supprimé si :

- il contient des hôtes qui appartiennent uniquement à ce groupe ;
- il est marqué comme interne ;
- il est utilisé par un prototype d'hôte ;
- il est utilisé dans un script global ;
- il est utilisé dans une condition de corrélation.

> **Note:**
> Cette méthode est disponible uniquement pour les types d'utilisateur *Admin* et *Super admin*. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres du rôle utilisateur. Consultez Rôles d'utilisateur pour plus d'informations.

Paramètres

`(array)` ID des groupes d'hôtes à supprimer.

Valeurs de retour

`(object)` Renvoie un objet contenant les ID des groupes d'hôtes supprimés dans la propriété `groupids`.

Exemples

Suppression de plusieurs groupes d'hôtes

Supprimez deux groupes d'hôtes.

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "hostgroup.delete",
    "params": [
        "107824",
        "107825"
    ],
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "groupids": [
            "107824",
            "107825"
        ]
    },
    "id": 1
}
```

Source

CHostGroup::delete() dans *ui/include/classes/api/services/CHostGroup.php*.

hostgroup.get

Description

`integer/array hostgroup.get(object parameters)`

La méthode permet de récupérer des groupes d'hôtes selon les paramètres fournis.

> **Note:**
> Cette méthode est disponible pour les utilisateurs de tout type. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres des rôles utilisateur. Consultez Rôles utilisateur pour plus d'informations.

Paramètres

`(object)` Paramètres définissant la sortie souhaitée.

La méthode prend en charge les paramètres suivants.

Paramètre	Type	Description
graphids	ID/array	Retourner uniquement les groupes d'hôtes qui contiennent des hôtes avec les graphiques donnés.
groupids	ID/array	Retourner uniquement les groupes d'hôtes avec les ID de groupes d'hôtes donnés.
hostids	ID/array	Retourner uniquement les groupes d'hôtes qui contiennent les hôtes donnés.

Paramètre	Type	Description
maintenanceids	ID/array	Retourner uniquement les groupes d'hôtes qui sont affectés par les maintenances données.
triggerids	ID/array	Retourner uniquement les groupes d'hôtes qui contiennent des hôtes avec les déclencheurs donnés.
with_graphs	boolean	Retourner uniquement les groupes d'hôtes qui contiennent des hôtes avec des graphiques.
with_graph_prototypes	boolean	Retourner uniquement les groupes d'hôtes qui contiennent des hôtes avec des prototypes de graphiques.
with_hosts	boolean	Retourner uniquement les groupes d'hôtes qui contiennent des hôtes.
with_httptests	boolean	Retourner uniquement les groupes d'hôtes qui contiennent des hôtes avec des scénarios web. Remplace le paramètre `with_monitored_httptests`.
with_items	boolean	Retourner uniquement les groupes d'hôtes qui contiennent des hôtes avec des éléments. Remplace les paramètres `with_monitored_items` et `with_simple_graph_items`.
with_item_prototypes	boolean	Retourner uniquement les groupes d'hôtes qui contiennent des hôtes avec des prototypes d'éléments. Remplace le paramètre `with_simple_graph_item_prototypes`.
with_simple_graph_item_prototypes	boolean	Retourner uniquement les groupes d'hôtes qui contiennent des hôtes avec des prototypes d'éléments, qui sont activés pour la création et ont un type d'information numérique.
with_monitored_httptests	boolean	Retourner uniquement les groupes d'hôtes qui contiennent des hôtes avec des scénarios web activés.
with_monitored_hosts	boolean	Retourner uniquement les groupes d'hôtes qui contiennent des hôtes surveillés.
with_monitored_items	boolean	Retourner uniquement les groupes d'hôtes qui contiennent des hôtes avec des éléments activés. Remplace le paramètre `with_simple_graph_items`.
with_monitored_triggers	boolean	Retourner uniquement les groupes d'hôtes qui contiennent des hôtes avec des déclencheurs activés. Tous les éléments utilisés dans le déclencheur doivent également être activés.
with_simple_graph_items	boolean	Retourner uniquement les groupes d'hôtes qui contiennent des hôtes avec des éléments numériques.
with_triggers	boolean	Retourner uniquement les groupes d'hôtes qui contiennent des hôtes avec des déclencheurs. Remplace le paramètre `with_monitored_triggers`.
selectDiscoveryRules	query	Retourner une propriété `discoveryRules` avec les règles LLD qui ont découvert le groupe d'hôtes.
selectDiscoveryData	query	Retourner une propriété `discoveryData` avec les objets de découverte de groupes d'hôtes. Chaque objet de découverte de groupe d'hôtes est un prototype de groupe d'hôtes lié au groupe d'hôtes découvert et possède les propriétés suivantes : `parent_group_prototypeid` - `(ID)` ID du prototype de groupe d'hôtes à partir duquel le groupe d'hôtes a été découvert ; `name` - `(string)` nom du prototype de groupe d'hôtes ; `ts_delete` - `(timestamp)` heure à laquelle le groupe d'hôtes qui n'est plus découvert sera supprimé ; `status` - `(int)` statut de découverte du groupe d'hôtes : 0 - *(par défaut)* le groupe d'hôtes est découvert, 1 - le groupe d'hôtes n'est plus découvert.
selectHostPrototypes	query	Retourner une propriété `hostPrototypes` avec les prototypes d'hôtes qui ont découvert ce groupe d'hôtes.

Paramètre	Type	Description
selectHosts	query	Retourner une propriété `hosts` avec les hôtes qui appartiennent au groupe d'hôtes. Prend en charge `count`.
limitSelects	integer	Limite le nombre d'enregistrements renvoyés par les sous-sélections. S'applique aux sous-sélections suivantes : `selectHosts` - les résultats seront triés par `host`.
sortfield	string/array	Trier le résultat selon les propriétés données. Valeurs possibles : `groupid`, `name`.
countOutput	boolean	Ces paramètres sont décrits dans le commentaire de référence.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	
selectGroupDiscoveries	query	Retourner une propriété `groupDiscoveries` avec les objets de découverte de groupes d'hôtes. Chaque objet de découverte de groupe d'hôtes est un prototype de groupe d'hôtes lié au groupe d'hôtes découvert. Cette requête est **obsolète**, veuillez utiliser `selectDiscoveryData` à la place.

Valeurs de retour

`(integer/array)` Renvoie soit :

- un tableau d'objets
- le nombre d'objets récupérés, si le paramètre `countOutput` a été utilisé

Exemples

Récupération des données par nom

Récupérez toutes les données concernant deux groupes d'hôtes nommés « Zabbix servers » et « Linux servers ».

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "hostgroup.get",
    "params": {
        "output": "extend",
        "filter": {
            "name": [
                "Zabbix servers",
                "Linux servers"
            ]
        }
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "groupid": "2",
            "name": "Linux servers",
            "flags": "0",
            "uuid": "dc579cd7a1a34222933f24f52a68bcd8"
        },
        {
            "groupid": "4",
            "name": "Zabbix servers",
            "flags": "0",
            "uuid": "6f6799aa69e844b4b3918f779f2abf08"
        }
    ],
    "id": 1
}
```


Voir aussi

- Hôte

Source

CHostGroup::get() dans *ui/include/classes/api/services/CHostGroup.php*.

hostgroup.massadd

Description

`object hostgroup.massadd(object parameters)`

Cette méthode permet d'ajouter simultanément plusieurs objets associés à tous les groupes d'hôtes spécifiés.

> **Note:**
> Cette méthode est disponible uniquement pour les types d'utilisateur *Admin* et *Super admin*. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres du rôle utilisateur. Consultez Rôles d'utilisateur pour plus d'informations.

Paramètres

`(object)` Paramètres contenant les ID des groupes d'hôtes à mettre à jour et les objets à ajouter à tous les groupes d'hôtes.

La méthode accepte les paramètres suivants.

Paramètre	Type	Description
groups	object/array	Groupes d'hôtes à mettre à jour. Les groupes d'hôtes doivent avoir uniquement la propriété `groupid` définie. Comportement des paramètres : - *obligatoire*
hosts	object/array	Hôtes à ajouter à tous les groupes d'hôtes. Les hôtes doivent avoir uniquement la propriété `hostid` définie.

Valeurs retournées

`(object)` Renvoie un objet contenant les identifiants des groupes d'hôtes mis à jour sous la propriété `groupids`.

Exemples

Ajout d'hôtes à des groupes d'hôtes

Ajoutez deux hôtes aux groupes d'hôtes ayant les ID 5 et 6.

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "hostgroup.massadd",
    "params": {
        "groups": [
            {
                "groupid": "5"
            },
            {
                "groupid": "6"
            }
        ],
        "hosts": [
            {
                "hostid": "30050"
            },
            {
                "hostid": "30001"
            }
        ]
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "groupids": [
            "5",
            "6"
        ]
    },
    "id": 1
}
```

Voir aussi

- Hôte

Source

CHostGroup::massAdd() dans *ui/include/classes/api/services/CHostGroup.php*.

hostgroup.massremove

Description

`object hostgroup.massremove(object parameters)`

Cette méthode vous permet de supprimer des objets associés de plusieurs groupes d'hôtes.

> **Note:**
> Cette méthode est disponible uniquement pour les types d'utilisateur *Admin* et *Super admin*. Les autorisations d'appel de cette méthode peuvent être révoquées dans les paramètres des rôles utilisateur. Consultez Rôles utilisateur pour plus d'informations.

Paramètres

`(object)` Paramètres contenant les ID des groupes d'hôtes à mettre à jour et les objets qui doivent être supprimés.

Paramètre	Type	Description
groupids	ID/array	ID des groupes d'hôtes à mettre à jour. Comportement du paramètre: - *obligatoire*
hostids	ID/array	ID des hôtes à supprimer de tous les groupes d'hôtes.

Valeurs retournées

`(object)` Renvoie un objet contenant les identifiants des groupes d'hôtes mis à jour sous la propriété `groupids`.

Exemples

Suppression d'hôtes de groupes d'hôtes

Supprimez deux hôtes des groupes d'hôtes indiqués.

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "hostgroup.massremove",
    "params": {
        "groupids": [
            "5",
            "6"
        ],
        "hostids": [
            "30050",
            "30001"
        ]
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "groupids": [
            "5",
            "6"
        ]
    },
    "id": 1
}
```

Source

CHostGroup::massRemove() dans *ui/include/classes/api/services/CHostGroup.php*.

hostgroup.propagate

Description

`object hostgroup.propagate(object parameters)`

Cette méthode vous permet d'appliquer des autorisations et des filtres de balises à tous les sous-groupes d'un groupe d'hôtes.

> **Note:**
> Cette méthode est disponible uniquement pour les types d'utilisateur *Super admin*. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres des rôles utilisateur. Consultez User roles pour plus d'informations.

Paramètres

`(object)` Paramètres définissant la sortie souhaitée.

La méthode prend en charge les paramètres suivants.

Paramètre	Type	Description
groups	object/array	Groupes d'hôtes à propager. Les groupes d'hôtes doivent avoir la propriété `groupid` définie. Comportement des paramètres : - *obligatoire*
permissions	boolean	Définissez sur "true" pour propager les autorisations. Comportement des paramètres : - *obligatoire* si `tag_filters` n'est pas défini
tag_filters	boolean	Définissez sur "true" pour propager les filtres de tags. Comportement des paramètres : - *obligatoire* si `permissions` n'est pas défini

Valeurs de retour

`(object)` Retourne un objet contenant des ID des hotes propagés groupe sous la propriété `groupids`.

Exemples

Propagation des permissions de groupe d'hôtes et des filtres de tags à ses sous-groupes.

Propager les permissions de groupe d'hôtes et les filtres de tags à ses sous-groupes.

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "hostgroup.propagate",
    "params": {
        "groups": [
            {
                "groupid": "6"
            }
        ],
        "permissions": true,
        "tag_filters": true
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "groupids": [
            "6",
        ]
    },
    "id": 1
}
```

Voir aussi

- hostgroup.update
- hostgroup.massadd
- Hôte

Source

CHostGroup::propagate() dans *ui/include/classes/api/services/CHostGroup.php*.

hostgroup.update

Description

`object hostgroup.update(object/array hostGroups)`

Cette méthode vous permet de mettre à jour des groupes d'hôtes existants.

> **Note:**
> Cette méthode est disponible uniquement pour les types d'utilisateur *Admin* et *Super admin*. Les autorisations d'appel de cette méthode peuvent être révoquées dans les paramètres des rôles utilisateur. Consultez Rôles utilisateur pour plus d'informations.

Paramètres

`(object/array)` Propriétés du groupe d'hôtes à mettre à jour.

La propriété `groupid` doit être définie pour chaque groupe d'hôtes, toutes les autres propriétés sont facultatives. Seules les propriétés indiquées seront mises à jour, toutes les autres resteront inchangées.

Valeurs retournées

`(object)` Renvoie un objet contenant les identifiants des groupes d'hôtes mis à jour sous la propriété `groupids`.

Exemples

Renommer un groupe d'hôtes

Renommez un groupe d'hôtes en « Linux hosts ».

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "hostgroup.update",
    "params": {
        "groupid": "7",
        "name": "Linux hosts"
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "groupids": [
            "7"
        ]
    },
    "id": 1
}
```

Renommage de plusieurs groupes d'hôtes

Renommez deux groupes d'hôtes pour inclure des informations de région dans leurs noms.

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "hostgroup.update",
    "params": [
        {
            "groupid": "8",
            "name": "Europe/Linux hosts"
        },
        {
            "groupid": "9",
```

```
            "name": "Europe/Windows hosts"
        }
    ],
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "groupids": [
            "8",
            "9"
        ]
    },
    "id": 1
}
```

Source

CHostGroup::update() dans *ui/include/classes/api/services/CHostGroup.php*.

Groupe d'utilisateurs

Cette classe est conçue pour fonctionner avec les groupes d'utilisateurs.

Références d'objet :

- Groupe d'utilisateurs
- Permission
- Permission basée sur les tags

Méthodes disponibles :

- usergroup.create - créer de nouveaux groupes d'utilisateurs
- usergroup.delete - supprimer des groupes d'utilisateurs
- usergroup.get - récupérer des groupes d'utilisateurs
- usergroup.update - mettre à jour des groupes d'utilisateurs

Objet groupe d'utilisateurs

Les objets suivants sont directement liés à l'API `usergroup`.

Groupe d'utilisateurs

L'objet groupe d'utilisateurs possède les propriétés suivantes.

Property	Type	Description
usrgrpid	ID	ID du groupe d'utilisateurs. Comportement de la propriété : - *en lecture seule* - *requis* pour les opérations de mise à jour
name	string	Nom du groupe d'utilisateurs. Comportement de la propriété : - *requis* pour les opérations de création
debug_mode	integer	Indique si le mode de débogage est activé ou désactivé. Valeurs possibles : 0 - *(par défaut)* désactivé ; 1 - activé.

Property	Type	Description
gui_access	integer	Méthode d'authentification de l'interface pour les utilisateurs du groupe. Valeurs possibles : 0 - *(par défaut)* utiliser la méthode d'authentification par défaut du système ; 1 - utiliser l'authentification interne ; 2 - utiliser l'authentification LDAP ; 3 - désactiver l'accès à l'interface.
mfa_status	integer	Indique si l'authentification multifacteur (MFA) est activée ou désactivée pour les utilisateurs du groupe. Valeurs possibles : 0 - désactivée (pour toutes les méthodes MFA configurées) ; 1 - activée (pour toutes les méthodes MFA configurées).
mfaid	ID	Méthode MFA utilisée pour les utilisateurs du groupe. Comportement de la propriété : - *prise en charge* pour les utilisateurs de type *Super administrateur* - *prise en charge* si `mfa_status` de l'objet Authentification est défini sur « Activée »
proxy_mode	integer	Mode de gestion de l'accès aux proxy qui ne sont pas inclus dans des groupes de proxy. Valeurs possibles : 0 - Liste de refus ; 1 - *(par défaut)* Liste d'autorisation. Comportement de la propriété : - *pris en charge* pour les utilisateurs de type *Super administrateur*.
proxy_group_mode	integer	Mode de gestion de l'accès aux groupes de proxy (et à leurs proxy). Valeurs possibles : 0 - Liste de refus ; 1 - *(par défaut)* Liste d'autorisation. Comportement de la propriété : - *pris en charge* pour les utilisateurs de type *Super administrateur*.
users_status	integer	Indique si le groupe d'utilisateurs est activé ou désactivé. Pour les utilisateurs déprovisionnés, le groupe d'utilisateurs ne peut pas être activé. Valeurs possibles : 0 - *(par défaut)* activé ; 1 - désactivé.
userdirectoryid	ID	ID de l'annuaire d'utilisateurs utilisé pour l'authentification. Comportement de la propriété : - *pris en charge* pour les utilisateurs de type *Super administrateur* - *pris en charge* si `gui_access` est défini sur « utiliser la méthode d'authentification par défaut du système » ou « utiliser l'authentification LDAP »

Autorisation

L'objet d'autorisation possède les propriétés suivantes.

Propriété	Type	Description
id	ID	ID du groupe d'hôtes ou du groupe de modèles auquel ajouter l'autorisation. Comportement de la propriété: - *obligatoire*
permission	integer	Niveau d'accès au groupe d'hôtes ou au groupe de modèles. Valeurs possibles : 0 - accès refusé ; 2 - accès en lecture seule ; 3 - accès en lecture-écriture. Comportement de la propriété: - *obligatoire* pour les opérations de création

Autorisation basée sur les tags

L'objet d'autorisation basée sur les tags possède les propriétés suivantes.

Propriété	Type	Description
groupid	ID	ID du groupe d'hôtes auquel ajouter l'autorisation. Comportement de la propriété : - *obligatoire*
tag	string	Nom du tag.
value	string	Valeur du tag.

APM-based permission

The APM-based permission object has the following properties.

Property	Type	Description
key	string	APM resource attribute key. Property behavior: - *required*
value	string	APM resource attribute value.
signal_type	integer	APM resource attribute category. Possible values: -1 - *(default)* All; 0 - Traces; 1 - Metrics; 2 - Logs.

usergroup.create

Description

`object usergroup.create(object/array userGroups)`

Cette méthode vous permet de créer de nouveaux groupes d'utilisateurs.

> **Note:**
> Cette méthode est disponible uniquement pour le type d'utilisateur *Super admin*. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres du rôle utilisateur. Consultez Rôles utilisateur pour plus d'informations.

Paramètres

`(object/array)` Groupes d'utilisateurs à créer.

En plus des propriétés standard des groupes d'utilisateurs, la méthode accepte les paramètres suivants.

Parameter	Type	Description
hostgroup_rights	object/array	Permissions des groupes d'hôtes à attribuer au groupe d'utilisateurs.
proxies	object/array	Proxies à associer au groupe d'utilisateurs. Les proxies doivent uniquement avoir la propriété `proxyid` définie.
proxy_groups	object/array	Groupes de proxies à associer au groupe d'utilisateurs. Les groupes de proxies doivent uniquement avoir la propriété `proxy_groupid` définie.
templategroup_rights	object/array	Permissions des groupes de modèles à attribuer au groupe d'utilisateurs.
tag_filters	array	Permissions basées sur les tags à attribuer au groupe d'utilisateurs.
users	object/array	Utilisateurs à ajouter au groupe d'utilisateurs. L'utilisateur doit uniquement avoir la propriété `userid` définie.

Valeurs de retour

`(object)` Renvoie un objet contenant les ID des groupes d'utilisateurs créés dans la propriété `usrgrpids`. L'ordre des ID renvoyés correspond à l'ordre des groupes d'utilisateurs transmis.

Exemples

Création d'un groupe d'utilisateurs

Créez un groupe d'utilisateurs *Operation managers* avec un accès refusé au groupe d'hôtes "2", et ajoutez-y un utilisateur.

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "usergroup.create",
    "params": {
        "name": "Operation managers",
        "hostgroup_rights": {
            "id": "2",
            "permission": 0
        },
        "users": [
            {
                "userid": "12"
            }
        ]
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "usrgrpids": [
            "20"
        ]
    },
    "id": 1
}
```

Voir également

- Autorisation

Source

CUserGroup::create() dans *ui/include/classes/api/services/CUserGroup.php*.

usergroup.delete

Description

`object usergroup.delete(array userGroupIds)`

Cette méthode permet de supprimer des groupes d'utilisateurs.

> **Attention:**
> Le groupe d'utilisateurs Deprovisioned (le groupe d'utilisateurs spécifié pour `disabled_usrgrpid` dans Authentication) ne peut pas être supprimé.

> **Note:**
> Cette méthode est disponible uniquement pour le type d'utilisateur *Super admin*. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres du rôle utilisateur. Voir Rôles utilisateur pour plus d'informations.

Paramètres

`(array)` ID des groupes d'utilisateurs à supprimer.

Valeurs de retour

`(object)` Renvoie un objet contenant les ID des groupes d'utilisateurs supprimés dans la propriété `usrgrpids`.

Exemples

Suppression de plusieurs groupes d'utilisateurs

Supprimez deux groupes d'utilisateurs.

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "usergroup.delete",
    "params": [
        "20",
        "21"
    ],
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "usrgrpids": [
            "20",
            "21"
        ]
    },
    "id": 1
}
```

Source

CUserGroup::delete() dans *ui/include/classes/api/services/CUserGroup.php*.

usergroup.get

Description

`integer/array usergroup.get(object parameters)`

La méthode permet de récupérer des groupes d'utilisateurs selon les paramètres fournis.

Note:
Cette méthode est disponible pour les utilisateurs de tout type. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres des rôles utilisateur. Consultez Rôles utilisateur pour plus d'informations.

Paramètres

`(object)` Paramètres définissant le résultat souhaité.

La méthode prend en charge les paramètres suivants.

Paramètre	Type	Description
mfaids	ID/array	Retourne uniquement les groupes d'utilisateurs associés aux méthodes MFA indiquées. Comportement du paramètre : - *pris en charge* pour les utilisateurs de type *Super administrateur*
mfa_status	integer	Retourne uniquement les groupes d'utilisateurs ayant le statut MFA indiqué. Consultez la page des groupes d'utilisateurs pour obtenir la liste des statuts pris en charge.
proxyids	ID/array	Retourne uniquement les groupes d'utilisateurs contenant les proxy indiqués. Comportement du paramètre : - *pris en charge* pour les utilisateurs de type *Super administrateur*.
proxy_groupids	ID/array	Retourne uniquement les groupes d'utilisateurs contenant les groupes de proxy indiqués. Comportement du paramètre : - *pris en charge* pour les utilisateurs de type *Super administrateur*.
selectProxies	ID/array	Retourne une propriété `proxies` contenant les proxy du groupe d'utilisateurs. Comportement du paramètre : - *pris en charge* pour les utilisateurs de type *Super administrateur*.
selectProxyGroups	ID/array	Retourne une propriété `proxy_groups` contenant les groupes de proxy du groupe d'utilisateurs. Comportement du paramètre : - *pris en charge* pour les utilisateurs de type *Super administrateur*.
status	integer	Retourne uniquement les groupes d'utilisateurs ayant le statut indiqué. Consultez la page des groupes d'utilisateurs pour obtenir la liste des statuts pris en charge.
userids	ID/array	Retourne uniquement les groupes d'utilisateurs contenant les utilisateurs indiqués.
usrgrpids	ID/array	Retourne uniquement les groupes d'utilisateurs possédant les ID indiqués.
selectTagFilters	query	Retourne les permissions des groupes d'utilisateurs basées sur les tags dans la propriété `tag_filters`.
selectUsers	query	Retourne les utilisateurs du groupe d'utilisateurs dans la propriété `users`. Consultez `user.get` pour connaître les restrictions selon le type d'utilisateur.
selectHostGroupRights	query	Retourne les permissions du groupe d'utilisateurs sur les groupes d'hôtes dans la propriété `hostgroup_rights`. Consultez la page des groupes d'utilisateurs pour obtenir la liste des niveaux d'accès aux groupes d'hôtes.

Paramètre	Type	Description
selectTemplateGroupRights	query	Retourne les permissions du groupe d'utilisateurs sur les groupes de modèles dans la propriété `templategroup_rights`. Consultez la page des groupes d'utilisateurs pour obtenir la liste des niveaux d'accès aux groupes de modèles.
limitSelects	integer	Limite le nombre d'enregistrements retournés par les sous-sélections.
output	query	Propriétés de l'objet groupe d'utilisateurs à retourner. Les utilisateurs de type *Administrateur* et *Utilisateur* peuvent récupérer uniquement les propriétés suivantes : `usrgrpid`, `name`, `gui_access`, `users_status`, `debug_mode`, `mfa_status`. Par défaut : `extend`.
sortfield	string/array	Trie le résultat selon les propriétés indiquées. Valeurs possibles : `usrgrpid`, `name`.
countOutput	boolean	Ces paramètres sont décrits dans le commentaire de référence.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Valeurs retournées

`(integer/array)` Retourne soit:

- un tableau d'objets ;
- le nombre d'objets récupérés, si le paramètre `countOutput` a été utilisé.

Exemples

Récupération des groupes d'utilisateurs activés

Récupérez tous les groupes d'utilisateurs activés.

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "usergroup.get",
    "params": {
        "output": "extend",
        "status": 0
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "usrgrpid": "7",
            "name": "Zabbix administrators",
            "gui_access": "0",
            "users_status": "0",
            "debug_mode": "1",
            "userdirectoryid": "0",
```

```
            "mfa_status": "0",
            "mfaid": "0"
        },
        {
            "usrgrpid": "8",
            "name": "Guests",
            "gui_access": "0",
            "users_status": "0",
            "debug_mode": "0",
            "userdirectoryid": "0",
            "mfa_status": "0",
            "mfaid": "0"
        },
        {
            "usrgrpid": "11",
            "name": "Enabled debug mode",
            "gui_access": "0",
            "users_status": "0",
            "debug_mode": "1",
            "userdirectoryid": "0",
            "mfa_status": "0",
            "mfaid": "0"
        },
        {
            "usrgrpid": "12",
            "name": "No access to the frontend",
            "gui_access": "2",
            "users_status": "0",
            "debug_mode": "0",
            "userdirectoryid": "0",
            "mfa_status": "0",
            "mfaid": "0"
        },
        {
            "usrgrpid": "14",
            "name": "Read only",
            "gui_access": "0",
            "users_status": "0",
            "debug_mode": "0",
            "userdirectoryid": "0",
            "mfa_status": "0",
            "mfaid": "0"
        },
        {
            "usrgrpid": "18",
            "name": "Deny",
            "gui_access": "0",
            "users_status": "0",
            "debug_mode": "0",
            "userdirectoryid": "0",
            "mfa_status": "0",
            "mfaid": "0"
        }
    ],
    "id": 1
}
```


Voir également

- Utilisateur

Source

CUserGroup::get() dans *ui/include/classes/api/services/CUserGroup.php*.

usergroup.update

Description

`object usergroup.update(object/array userGroups)`

Cette méthode vous permet de mettre à jour des groupes d'utilisateurs existants.

> **Note:**
> Cette méthode est disponible uniquement pour le type d'utilisateur *Super admin*. Les autorisations d'appel de cette méthode peuvent être révoquées dans les paramètres du rôle utilisateur. Consultez User roles pour plus d'informations.

Paramètres

`(object/array)` Propriétés des groupes d'utilisateurs à mettre à jour.

La propriété `usrgrpid` doit être définie pour chaque groupe d'utilisateurs ; toutes les autres propriétés sont facultatives. Seules les propriétés transmises seront mises à jour, toutes les autres resteront inchangées.

En plus des propriétés standard des groupes d'utilisateurs, la méthode accepte les paramètres suivants.

Parameter	Type	Description
hostgroup_rights	object/array	Permissions des groupes de hôtes qui remplaceront les permissions actuellement attribuées au groupe d'utilisateurs.
proxies	object/array	Proxies à associer au groupe d'utilisateurs. Les proxies doivent avoir uniquement la propriété `proxyid` définie.
proxy_groups	object/array	Groupes de proxies à associer au groupe d'utilisateurs. Les groupes de proxies doivent avoir uniquement la propriété `proxy_groupid` définie.
templategroup_rights	object/array	Permissions des groupes de modèles qui remplaceront les permissions actuellement attribuées au groupe d'utilisateurs.
tag_filters	array	Permissions basées sur les tags qui remplaceront les permissions actuellement attribuées au groupe d'utilisateurs.
users	object/array	Utilisateurs qui remplaceront les utilisateurs actuellement associés au groupe d'utilisateurs. L'utilisateur doit avoir uniquement la propriété `userid` définie.

Valeurs retournées

`(object)` Retourne un objet contenant les identifiants des groupes d'utilisateurs mis à jour sous la propriété `usrgrpids`.

Exemples

Activation d'un groupe d'utilisateurs et mise à jour des autorisations

Activez un groupe d'utilisateurs et accordez-lui un accès en lecture-écriture aux groupes d'hôtes « 2 » et « 4 ».

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "usergroup.update",
    "params": {
        "usrgrpid": "17",
        "users_status": "0",
        "hostgroup_rights": [
            {
                "id": "2",
                "permission": 3
            },
            {
                "id": "4",
                "permission": 3
            }
        ]
```

```
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "usrgrpids": [
            "17"
        ]
    },
    "id": 1
}
```

Voir également

- Autorisation

Source

CUserGroup::update() dans *ui/include/classes/api/services/CUserGroup.php*.

Groupe de modèles

Cette classe est conçue pour fonctionner avec les groupes de modèles.

Références d'objet :

- Groupe de modèles

Méthodes disponibles :

- templategroup.create - créer de nouveaux groupes de modèles
- templategroup.delete - supprimer des groupes de modèles
- templategroup.get - récupérer des groupes de modèles
- templategroup.massadd - ajouter des objets associés aux groupes de modèles
- templategroup.massremove - supprimer des objets associés des groupes de modèles
- templategroup.propagate - propager les permissions aux sous-groupes des groupes de modèles
- templategroup.update - mettre à jour des groupes de modèles

Objet groupe de modèles

Les objets suivants sont directement liés à l'API `templategroup`.

Groupe de modèles

L'objet groupe de modèles possède les propriétés suivantes.

Propriété	Type	Description
groupid	ID	ID du groupe de modèles. Comportement de la propriété: - *lecture seule* - *obligatoire* pour les opérations de mise à jour
name	string	Nom du groupe de modèles. Comportement de la propriété: - *obligatoire* pour les opérations de création
uuid	string	Identifiant unique universel, utilisé pour lier les groupes de modèles importés à ceux qui existent déjà. Généré automatiquement s'il n'est pas fourni.

templategroup.create

Description

`object templategroup.create(object/array templateGroups)`

Cette méthode permet de créer de nouveaux groupes de modèles.

> **Note:**
> Cette méthode est disponible uniquement pour le type d'utilisateur *Super admin*. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres des rôles utilisateur. Voir User roles pour plus d'informations.

Paramètres

`(object/array)` Groupes de modèles à créer. La méthode accepte des groupes de modèles avec les propriétés standard des groupes de modèles.

Valeurs de retour

`(object)` Renvoie un objet contenant les ID des groupes de modèles créés dans la propriété `groupids`. L'ordre des ID renvoyés correspond à l'ordre des groupes de modèles transmis.

Exemples

Création d'un groupe de modèles

Créez un groupe de modèles appelé "Templates/Databases".

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "templategroup.create",
    "params": {
        "name": "Templates/Databases"
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "groupids": [
            "107820"
        ]
    },
    "id": 1
}
```

Source

CTemplateGroup::create() dans *ui/include/classes/api/services/CTemplateGroup.php*.

templategroup.delete

Description

`object templategroup.delete(array templateGroupIds)`

Cette méthode permet de supprimer des groupes de modèles.

Un groupe de modèles ne peut pas être supprimé s'il contient des modèles qui appartiennent uniquement à ce groupe.

> **Note:**
> Cette méthode est disponible uniquement pour les types d'utilisateur *Admin* et *Super admin*. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres des rôles utilisateur. Consultez Rôles utilisateur pour plus d'informations.

Paramètres

`(tableau)` IDs du groupe de modèle a supprimer.

Valeurs de retour

`(object)` Retourne un objet contenant l'ID des groupes de modèles sous la propriété `groupids`.

Exemples

Supprimer plusieurs Groupes de modèles

Supprimer deux groupes de modèles.

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "templategroup.delete",
    "params": [
        "107814",
        "107815"
    ],
    "id": 1
}
```

Réponse:

```
{
    "jsonrpc": "2.0",
    "result": {
        "groupids": [
            "107814",
            "107815"
        ]
    },
    "id": 1
}
```

Source

CTemplateGroup::delete() in *ui/include/classes/api/services/CTemplateGroup.php*.

templategroup.get

Description

`integer/array templategroup.get(object parameters)`

La méthode permet de récupérer des groupes de modèles selon les paramètres fournis.

> **Note:**
> Cette méthode est disponible pour les utilisateurs de tout type. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres des rôles utilisateur. Consultez Rôles utilisateur pour plus d'informations.

Paramètres

`(object)` Paramètres définissant la sortie souhaitée.

La méthode prend en charge les paramètres suivants.

Paramètre	Type	Description
graphids	ID/array	Retourner uniquement les groupes de modèles qui contiennent des modèles avec les graphes indiqués.
groupids	ID/array	Retourner uniquement les groupes de modèles avec les ID de groupes de modèles indiqués.
templateids	ID/array	Retourner uniquement les groupes de modèles qui contiennent les modèles indiqués.

Paramètre	Type	Description
triggerids	ID/array	Retourner uniquement les groupes de modèles qui contiennent des modèles avec les déclencheurs indiqués.
with_graphs	boolean	Retourner uniquement les groupes de modèles qui contiennent des modèles avec des graphes.
with_graph_prototypes	boolean	Retourner uniquement les groupes de modèles qui contiennent des modèles avec des prototypes de graphes.
with_httptests	boolean	Retourner uniquement les groupes de modèles qui contiennent des modèles avec des scénarios web.
with_items	boolean	Retourner uniquement les groupes de modèles qui contiennent des modèles avec des éléments. Remplace les paramètres `with_simple_graph_items`.
with_item_prototypes	boolean	Retourner uniquement les groupes de modèles qui contiennent des modèles avec des prototypes d'éléments. Remplace le paramètre `with_simple_graph_item_prototypes`.
with_simple_graph_item_prototypes	boolean	Retourner uniquement les groupes de modèles qui contiennent des modèles avec des prototypes d'éléments, qui sont activés pour la création et ont un type d'information numérique.
with_simple_graph_items	boolean	Retourner uniquement les groupes de modèles qui contiennent des modèles avec des éléments numériques.
with_templates	boolean	Retourner uniquement les groupes de modèles qui contiennent des modèles.
with_triggers	boolean	Retourner uniquement les groupes de modèles qui contiennent des modèles avec des déclencheurs.
selectTemplates	query	Retourner une propriété `templates` avec les modèles qui appartiennent au groupe de modèles. Prend en charge `count`.
limitSelects	integer	Limite le nombre d'enregistrements renvoyés par les sous-sélections. S'applique aux sous-sélections suivantes : `selectTemplates` - les résultats seront triés par `template`.
sortfield	string/array	Trier le résultat selon les propriétés indiquées. Valeurs possibles : `groupid`, `name`.
countOutput	boolean	Ces paramètres sont décrits dans le commentaire de référence.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Valeurs de retour

`(integer/array)` Renvoie soit :

- un tableau d'objets ;
- le nombre d'objets récupérés, si le paramètre `countOutput` a été utilisé.

Exemples

Récupération des données par nom

Récupérez toutes les données concernant deux groupes de modèles nommés "Templates/Databases" et "Templates/Modules".

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "templategroup.get",
    "params": {
        "output": "extend",
        "filter": {
            "name": [
                "Templates/Databases",
                "Templates/Modules"
            ]
        }
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "groupid": "13",
            "name": "Templates/Databases",
            "uuid": "748ad4d098d447d492bb935c907f652f"
        },
        {
            "groupid": "8",
            "name": "Templates/Modules",
            "uuid": "57b7ae836ca64446ba2c296389c009b7"
        }
    ],
    "id": 1
}
```

Voir aussi

- Modèle

Source

CTemplateGroup::get() dans *ui/include/classes/api/services/CTemplateGroup.php*.

templategroup.massadd

Description

`object templategroup.massadd(object parameters)`

Cette méthode permet d'ajouter simultanément plusieurs objets associés à tous les groupes de modèles spécifiés.

> **Note:**
> Cette méthode est disponible uniquement pour les types d'utilisateur *Admin* et *Super admin*. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres des rôles utilisateur. Consultez Rôles utilisateur pour plus d'informations.

Paramètres

`(object)` Paramètres contenant les ID des groupes de modèles à mettre à jour et les objets à ajouter à tous les groupes de modèles.

La méthode accepte les paramètres suivants.

Paramètre	Type	Description
groups	object/array	Groupes de modèles à mettre à jour. Les groupes de modèles doivent avoir uniquement la propriété `groupid` définie. Comportement des paramètres : - *obligatoire*
templates	object/array	Modèles à ajouter à tous les groupes de modèles. Les modèles doivent avoir uniquement la propriété `templateid` définie. Comportement des paramètres : - *obligatoire*

Valeurs de retour

`(object)` Renvoie un objet contenant les ID des groupes de modèles mis à jour dans la propriété `groupids`.

Exemples

Ajout de modèles à des groupes de modèles

Ajoutez deux modèles aux groupes de modèles ayant les ID 12 et 13.

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "templategroup.massadd",
    "params": {
        "groups": [
            {
                "groupid": "12"
            },
            {
                "groupid": "13"
            }
        ],
        "templates": [
            {
                "templateid": "10486"
            },
            {
                "templateid": "10487"
            }
        ]
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "groupids": [
            "12",
            "13"
        ]
    },
    "id": 1
}
```

Voir aussi

- Modèle

Source

CTemplateGroup::massAdd() dans *ui/include/classes/api/services/CTemplateGroup.php*.

templategroup.massremove

Description

`object templategroup.massremove(object parameters)`

Cette méthode vous permet de supprimer des objets associés de plusieurs groupes de modèles.

> **Note:**
> Cette méthode est disponible uniquement pour les types d'utilisateur *Admin* et *Super admin*. Les autorisations d'appel de cette méthode peuvent être révoquées dans les paramètres des rôles utilisateur. Consultez Rôles utilisateur pour plus d'informations.

Paramètres

`(object)` Paramètres contenant les ID des groupes de modèles à mettre à jour et les objets qui doivent être supprimés.

Paramètre	Type	Description
groupids	ID/array	ID des groupes de modèles à mettre à jour. Comportement des paramètres: - *obligatoire*
templateids	ID/array	ID des modèles à supprimer de tous les groupes de modèles. Comportement des paramètres: - *obligatoire*

Valeurs de retour

`(object)` Renvoie un objet contenant les identifiants des groupes de modèles mis à jour dans la propriété `groupids`.

Exemples

Suppression de modèles des groupes de modèles

Supprimez deux modèles des groupes de modèles indiqués.

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "templategroup.massremove",
    "params": {
        "groupids": [
            "5",
            "6"
        ],
        "templateids": [
            "30050",
            "30001"
        ]
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "groupids": [
```

```
            "5",
            "6"
        ]
    },
    "id": 1
}
```

Source

CTemplateGroup::massRemove() dans *ui/include/classes/api/services/CTemplateGroup.php*.

templategroup.propagate

Description

`object templategroup.propagate(object parameters)`

Cette méthode vous permet d'appliquer des autorisations à tous les sous-groupes des groupes de modèles.

Note:
Cette méthode est disponible uniquement pour les types d'utilisateurs *Super admin*. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres des rôles utilisateur. Consultez User roles pour plus d'informations.

Paramètres

`(object)` Paramètres définissant la sortie souhaitée.

La méthode prend en charge les paramètres suivants.

Paramètre	Type	Description
groups	object/array	Groupes de modèles à propager. Les groupes de modèles doivent avoir uniquement la propriété `groupid` définie. Comportement des paramètres : - *obligatoire*
permissions	boolean	Définissez `true` s'il est nécessaire de propager les permissions. Comportement des paramètres : - *obligatoire*

Valeurs de retour

`(object)` Renvoie un objet contenant les ID des groupes de modèles propagés dans la propriété `groupids`.

Exemples

Propagation des permissions du groupe de modèles à ses sous-groupes.

Propager les permissions du groupe de modèles à ses sous-groupes.

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "templategroup.propagate",
    "params": {
        "groups": [
            {
                "groupid": "15"
            }
        ],
        "permissions": true
    },
```

```
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "groupids": [
            "15",
        ]
    },
    "id": 1
}
```

Voir aussi

- templategroup.update
- templategroup.massadd
- Template

Source

CTemplateGroup::propagate() dans *ui/include/classes/api/services/CTemplateGroup.php*.

templategroup.update

Description

`object templategroup.update(object/array templateGroups)`

Cette méthode vous permet de mettre à jour des groupes de modèles existants.

> **Note:**
> Cette méthode est disponible uniquement pour les types d'utilisateur *Admin* et *Super admin*. Les autorisations d'appel de cette méthode peuvent être révoquées dans les paramètres des rôles utilisateur. Consultez Rôles utilisateur pour plus d'informations.

Paramètres

`(object/array)` Propriétés du groupe de modèles à mettre à jour.

La propriété `groupid` doit être définie pour chaque groupe de modèles, toutes les autres propriétés sont facultatives. Seules les propriétés indiquées seront mises à jour, toutes les autres resteront inchangées.

Valeurs de retour

`(object)` Renvoie un objet contenant les ID des groupes de modèles mis à jour dans la propriété `groupids`.

Exemples

Renommer un groupe de modèles

Renommez un groupe de modèles en « Templates/Databases »

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "templategroup.update",
    "params": {
        "groupid": "7",
        "name": "Templates/Databases"
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "groupids": [
            "7"
        ]
    },
    "id": 1
}
```

Source

CTemplateGroup::update() dans *ui/include/classes/api/services/CTemplateGroup.php*.

Groupe de proxies

Cette classe est conçue pour fonctionner avec des groupes de proxies.

Références d'objet :

- Groupe de proxies

Méthodes disponibles :

- proxygroup.create - créer de nouveaux groupes de proxies
- proxygroup.delete - supprimer des groupes de proxies
- proxygroup.get - récupérer des groupes de proxies
- proxygroup.update - mettre à jour des groupes de proxies

Objet groupe de proxys

Les objets suivants sont directement liés à l'API `proxygroup`.

Groupe de proxies

L'objet groupe de proxies possède les propriétés suivantes.

Property	Type	Description
proxy_groupid	ID	ID du groupe de proxies. Comportement de la propriété : - *en lecture seule* - *obligatoire* pour les opérations de mise à jour
name	string	Nom du groupe de proxies. Comportement de la propriété : - *obligatoire* pour les opérations de création
description	text	Description du groupe de proxies.
failover_delay	string	Période pendant laquelle un proxy du groupe de proxies doit communiquer avec le serveur Zabbix pour être considéré comme étant en ligne. Les suffixes de temps sont pris en charge (par exemple, `30s`, `1m`). Les macros utilisateur sont prises en charge. Valeurs possibles : 10s-15m. Valeur par défaut : 1m.

Property	Type	Description
min_online	string	Nombre minimal de proxies en ligne requis pour maintenir le groupe de proxies en ligne. Les macros utilisateur sont prises en charge. Plage de valeurs possibles : 1-1000. Valeur par défaut : 1.
state	integer	État du groupe de proxies. Valeurs possibles : 0 - Inconnu ; 1 - Hors ligne ; 2 - Récupération ; 3 - En ligne ; 4 - Dégradation. Comportement de la propriété : *- en lecture seule*

proxygroup.create

Description

`object proxygroup.create(object/array proxyGroups)`

Cette méthode vous permet de créer de nouveaux groupes de proxy.

Note:
Cette méthode est disponible uniquement pour le type d'utilisateur *Super admin*. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres du rôle utilisateur. Consultez User roles pour plus d'informations.

Paramètres

`(object/array)` Groupes de proxy à créer.

La méthode accepte des groupes de proxy avec les propriétés standard des groupes de proxy.

Valeurs de retour

`(object)` Renvoie un objet contenant les ID des groupes de proxys créés dans la propriété `proxy_groupids`. L'ordre des ID renvoyés correspond à l'ordre des groupes de proxys transmis.

Exemples

Créer un groupe de proxys

Créez un groupe de proxys avec des paramètres personnalisés.

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "proxygroup.create",
    "params": {
        "name": "Proxy group",
        "failover_delay": "5m",
        "min_online": "10"
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
```

```
    "result": {
        "proxy_groupids": [
            "5"
        ]
    },
    "id": 1
}
```

Source

CProxyGroup::create() dans *ui/include/classes/api/services/CProxyGroup.php*.

proxygroup.delete

Description

`object proxygroup.delete(array proxyGroupIds)`

Cette méthode vous permet de supprimer des groupes de proxy.

> **Note:**
> Cette méthode est disponible uniquement pour le type d'utilisateur *Super admin*. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres du rôle utilisateur. Consultez Rôles utilisateur pour plus d'informations.

Paramètres

`(array)` ID des groupes de proxys à supprimer.

Valeurs de retour

`(object)` Renvoie un objet contenant les ID des groupes de proxys supprimés dans la propriété `proxy_groupids`.

Exemples

Supprimer plusieurs groupes de proxys

Supprimez deux groupes de proxys.

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "proxygroup.delete",
    "params": [
        "5",
        "10"
    ],
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "proxy_groupids": [
            "5",
            "10"
        ]
    },
    "id": 1
}
```

Source

CProxyGroup::delete() dans *ui/include/classes/api/services/CProxyGroup.php*.

proxygroup.get

Description

`integer/array proxygroup.get(object parameters)`

Cette méthode permet de récupérer des groupes de proxy selon les paramètres indiqués.

> **Note:**
> Cette méthode est disponible pour les utilisateurs de tout type. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres du rôle utilisateur. Consultez la page Rôles utilisateur pour plus d'informations. Seuls les groupes de proxy autorisés par la *liste d'accès aux proxy* des groupes d'utilisateurs de l'utilisateur sont renvoyés.

Paramètres

`(object)` Paramètres définissant la sortie souhaitée.

La méthode prend en charge les paramètres suivants.

Paramètre	Type	Description
proxy_groupids	ID/array	Retourne uniquement les groupes de proxies avec les ID donnés.
proxyids	ID/array	Retourne uniquement les groupes de proxies qui contiennent les proxies donnés.
selectProxies	query	Retourne une propriété `proxies` avec les proxies qui appartiennent au groupe de proxies. Prend en charge `count`.
sortfield	string/array	Trie le résultat selon les propriétés données. Valeurs possibles : `proxy_groupid`, `name`.
countOutput	boolean	Ces paramètres sont décrits dans le commentaire de référence.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Valeurs de retour

`(integer/array)` Renvoie soit :

- un tableau d'objets ;
- le nombre d'objets récupérés, si le paramètre `countOutput` a été utilisé.

Exemples

Récupérer tous les groupes de proxys

Récupérez tous les groupes de proxys configurés avec leurs proxys.

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "proxygroup.get",
    "params": {
        "output": "extend",
        "selectProxies": ["proxyid", "name"]
    },
    "id": 1
}
```

Réponse :


```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "proxy_groupid": "1",
            "name": "Proxy group 1",
            "failover_delay": "1m",
            "min_online": "3",
            "description": "",
            "state": "1",
            "proxies": [
                {
                    "proxyid": "1",
                    "name": "proxy 1"
                },
                {
                    "proxyid": "2",
                    "name": "proxy 2"
                }
            ]
        },
        {
            "proxy_groupid": "2",
            "name": "Proxy group 2",
            "failover_delay": "10m",
            "min_online": "3",
            "description": "",
            "state": "3",
            "proxies": [
                {
                    "proxyid": "3",
                    "name": "proxy 3"
                },
                {
                    "proxyid": "4",
                    "name": "proxy 4"
                },
                {
                    "proxyid": "5",
                    "name": "proxy 5"
                }
            ]
        }
    ],
    "id": 1
}
```


Voir aussi

- Proxy

Source

CProxyGroup::get() dans *ui/include/classes/api/services/CProxyGroup.php*.

proxygroup.update

Description

`object proxygroup.update(object/array proxyGroups)`

Cette méthode vous permet de mettre à jour des groupes de proxy existants.

Note:
Cette méthode est disponible uniquement pour le type d'utilisateur *Super admin*. Les autorisations d'appel de cette méthode peuvent être révoquées dans les paramètres du rôle utilisateur. Consultez Rôles utilisateur pour plus d'informations.

Paramètres

`(object/array)` Propriétés des groupes de proxy à mettre à jour.

La propriété `proxy_groupid` doit être définie pour chaque groupe de proxy, toutes les autres propriétés sont facultatives. Seules les propriétés transmises seront mises à jour, toutes les autres resteront inchangées.

La méthode accepte des groupes de proxy avec les propriétés standard des groupes de proxy.

Valeurs de retour

`(object)` Renvoie un objet contenant les ID des groupes de proxys mis à jour dans la propriété `proxy_groupids`.

Exemples

Modifier le nombre minimum de proxies en ligne

Modifiez le nombre minimum de proxies en ligne requis pour que le groupe soit considéré comme en ligne.

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "proxygroup.update",
    "params": {
        "proxy_groupid": "5",
        "min_online": "3"
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "proxy_groupids": [
            "5"
        ]
    },
    "id": 1
}
```

Source

CProxyGroup::update() dans *ui/include/classes/api/services/CProxyGroup.php*.

Historique

Cette classe est conçue pour fonctionner avec les données d'historique.

Références d'objet :

- Historique flottant
- Historique entier
- Historique chaîne
- Historique texte
- Historique journal

Méthodes disponibles :

- history.clear - effacer les données d'historique
- history.get - récupérer les données d'historique
- history.push - envoyer les données d'historique au serveur Zabbix

Objet history

Les objets suivants sont directement liés à l'API `history`.

> **Note:**
> Les objets history diffèrent selon le type d'information de l'élément. Ils sont créés par le serveur Zabbix et ne peuvent pas être modifiés via l'API.

Historique des flottants

L'objet d'historique des flottants possède les propriétés suivantes.

Propriété	Type	Description
clock	timestamp	Heure à laquelle cette valeur a été reçue.
itemid	ID	ID de l'élément associé.
ns	integer	Nanosecondes au moment où la valeur a été reçue.
value	float	Valeur reçue.

Historique des entiers

L'objet d'historique des entiers possède les propriétés suivantes.

Propriété	Type	Description
clock	timestamp	Heure à laquelle cette valeur a été reçue.
itemid	ID	ID de l'élément associé.
ns	integer	Nanosecondes au moment où la valeur a été reçue.
value	integer	Valeur reçue.

Historique des chaînes

L'objet d'historique des chaînes possède les propriétés suivantes.

Propriété	Type	Description
clock	timestamp	Heure à laquelle cette valeur a été reçue.
itemid	ID	ID de l'élément associé.
ns	integer	Nanosecondes au moment où la valeur a été reçue.
value	string	Valeur reçue.

Historique du texte

L'objet d'historique du texte possède les propriétés suivantes.

Propriété	Type	Description
id	ID	ID de l'entrée d'historique.
clock	timestamp	Heure à laquelle cette valeur a été reçue.
itemid	ID	ID de l'élément associé.
ns	integer	Nanosecondes au moment où la valeur a été reçue.
value	text	Valeur reçue.

Historique des journaux

L'objet d'historique des journaux possède les propriétés suivantes.

Propriété	Type	Description
id	ID	ID de l'entrée d'historique.
clock	timestamp	Heure à laquelle cette valeur a été reçue.
itemid	ID	ID de l'élément associé.
logeventid	integer	ID de l'entrée du journal des événements Windows.

Propriété	Type	Description
ns	integer	Nanosecondes au moment où la valeur a été reçue.
severity	integer	Niveau de l'entrée du journal des événements Windows.
source	string	Source de l'entrée du journal des événements Windows.
timestamp	timestamp	Heure de l'entrée du journal des événements Windows.
value	text	Valeur reçue.

Historique binaire

L'objet d'historique binaire possède les propriétés suivantes.

Propriété	Type	Description
clock	timestamp	Heure à laquelle cette valeur a été reçue.
itemid	ID	ID de l'élément associé.
ns	integer	Nanosecondes au moment où la valeur a été reçue.
value	text	Valeur reçue.

Historique JSON

L'objet d'historique JSON possède les propriétés suivantes.

Propriété	Type	Description
clock	timestamp	Heure à laquelle cette valeur a été reçue.
itemid	ID	ID de l'élément associé.
ns	integer	Nanosecondes au moment où la valeur a été reçue.
value	text	Valeur reçue.

history.clear

Description

`object history.clear(array itemids)`

Cette méthode vous permet de supprimer l'historique des éléments.

> **Note:**
> Cette méthode est disponible uniquement pour les types d'utilisateur *Admin* et *Super admin*. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres du rôle utilisateur. Consultez Rôles d'utilisateur pour plus d'informations.

Paramètres

`(array)` ID des éléments à effacer.

Valeurs retournées

`(object)` Retourne un objet contenant les identifiants des éléments effacés sous la propriété `itemids`.

Exemples

Effacer l'historique

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "history.clear",
    "params": [
        "10325",
        "13205"
    ],
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "10325",
            "13205"
        ]
    },
    "id": 1
}
```

Source

CHistory::clear() dans *ui/include/classes/api/services/CHistory.php*.

history.get

Description

`integer/array history.get(object parameters)`

La méthode permet de récupérer les données d'historique selon les paramètres fournis.

Attention:
Cette méthode peut renvoyer des données historiques d'une entité supprimée si ces données n'ont pas encore été supprimées par le housekeeper.

Note:
Cette méthode est disponible pour les utilisateurs de tout type. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres des rôles utilisateur. Consultez Rôles utilisateur pour plus d'informations.

Paramètres

`(object)` Paramètres définissant la sortie souhaitée.

La méthode prend en charge les paramètres suivants.

Parameter	Type	Description
history	integer	Types d'objets d'historique à renvoyer. Valeurs possibles : 0 - flottant numérique; 1 - caractère; 2 - journal; 3 - *(par défaut)* entier non signé numérique; 4 - texte; 5 - binaire; 6 - JSON.
hostids	ID/array	Ne renvoyer que l'historique des hôtes spécifiés.
itemids	ID/array	Ne renvoyer que l'historique des éléments spécifiés.
maxValueSize	integer	Nombre maximal d'octets à renvoyer dans la propriété `value`. Plage de valeurs possibles : 1-134217728 (1 octet à 128 MiB). Si défini sur `null`, aucune limite n'est appliquée. Par défaut : 65536 (64 KiB). Comportement du paramètre : - *pris en charge* si `history` est défini sur "binary" ou "JSON"
time_from	timestamp	Ne renvoyer que les valeurs reçues après ou à l'heure donnée.
time_till	timestamp	Ne renvoyer que les valeurs reçues avant ou à l'heure donnée.

Parameter	Type	Description
sortfield	string/array	Trier le résultat selon les propriétés données. Valeurs possibles : `itemid`, `clock`, `ns`. Le tri par `ns` n'est pas pris en charge si les données d'historique sont stockées dans ClickHouse.
search	object	Renvoyer les résultats correspondant au motif donné (insensible à la casse). Accepte un objet, où les clés sont des noms de propriétés et les valeurs sont des chaînes à rechercher. Si aucune option supplémentaire n'est fournie, cela effectuera une recherche `LIKE "%…%"`. La recherche par `value` n'est pas prise en charge si `history` est défini sur 6 (JSON).
countOutput	boolean	Ces paramètres sont décrits dans le commentaire de référence.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Valeurs retournées

`(integer/array)` Retourne soit :

- un tableau d'objets ;
- le nombre d'objets récupérés, si le paramètre `countOutput` a été utilisé.

Exemples

Récupération des données d'historique d'un élément

Renvoie les 10 dernières valeurs reçues d'un élément numérique (float).

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "history.get",
    "params": {
        "output": "extend",
        "history": 0,
        "itemids": "23296",
        "sortfield": "clock",
        "sortorder": "DESC",
        "limit": 10
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "itemid": "23296",
            "clock": "1351090996",
            "value": "0.085",
            "ns": "563157632"
```

```
        },
        {
            "itemid": "23296",
            "clock": "1351090936",
            "value": "0.16",
            "ns": "549216402"
        },
        {
            "itemid": "23296",
            "clock": "1351090876",
            "value": "0.18",
            "ns": "537418114"
        },
        {
            "itemid": "23296",
            "clock": "1351090816",
            "value": "0.21",
            "ns": "522659528"
        },
        {
            "itemid": "23296",
            "clock": "1351090756",
            "value": "0.215",
            "ns": "507809457"
        },
        {
            "itemid": "23296",
            "clock": "1351090696",
            "value": "0.255",
            "ns": "495509699"
        },
        {
            "itemid": "23296",
            "clock": "1351090636",
            "value": "0.36",
            "ns": "477708209"
        },
        {
            "itemid": "23296",
            "clock": "1351090576",
            "value": "0.375",
            "ns": "463251343"
        },
        {
            "itemid": "23296",
            "clock": "1351090516",
            "value": "0.315",
            "ns": "447947017"
        },
        {
            "itemid": "23296",
            "clock": "1351090456",
            "value": "0.275",
            "ns": "435307141"
        }
    ],
    "id": 1
}
```


Source

CHistory::get() dans *ui/include/classes/api/services/CHistory.php*.

history.push

Description

`object history.push(object/array itemHistoryData)`

Cette méthode permet d'envoyer les données d'historique des éléments au serveur Zabbix.

> **Note:**
> Cette méthode est disponible pour les utilisateurs de tout type. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres du rôle utilisateur. Consultez User roles pour plus d'informations.

Paramètres

`(object/array)` Données d'historique de l'élément à envoyer.

La méthode prend en charge les paramètres suivants.

Paramètre	Type	Description
itemid	ID	ID de l'élément associé. Comportement du paramètre: - *obligatoire* si `host` et `key` ne sont pas définis
host	string	Nom technique de l'hôte. Comportement du paramètre: - *obligatoire* si `itemid` n'est pas défini
key	string	Clé de l'élément. Comportement du paramètre: - *obligatoire* si `itemid` n'est pas défini
value	mixed	Valeur de l'élément. Comportement du paramètre: - *obligatoire*
clock	timestamp	Heure à laquelle la valeur a été reçue.
ns	integer	Nanosecondes au moment où la valeur a été reçue.

Valeurs de retour

`(object)` Renvoie le résultat de l'opération d'envoi des données.

Exemples

Envoyer les données d'historique des éléments

Envoyez les données d'historique des éléments au serveur Zabbix pour les éléments « 10600 », « 10601 » et « 999999 ».

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "history.push",
    "params": [
        {
            "itemid": 10600,
            "value": 0.5,
            "clock": 1690891294,
            "ns": 45440940
        },
        {
            "itemid": 10600,
            "value": 0.6,
            "clock": 1690891295,
            "ns": 312431
        },
```

```
        {
            "itemid": 10601,
            "value": "[Tue Aug 01 15:01:35 2023] [error] [client 1.2.3.4] File does not exist: /var/www/ht
        },
        {
            "itemid": 999999,
            "value": 123
        }
    ],
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "response": "success",
        "data": [
            {
                "itemid": "10600"
            },
            {
                "itemid": "10600"
            },
            {
                "itemid": "10601",
                "error": "L'élément est désactivé."
            },
            {
                "error": "Aucune autorisation pour l'objet référencé ou celui-ci n'existe pas."
            }
        ]
    },
    "id": 1
}
```

Voir aussi

- Éléments Trapper items
- Éléments HTTP agent
- Host
- Item

Source

CHistory::push() dans *ui/include/classes/api/services/CHistory.php*.

Hôte

Cette classe est conçue pour fonctionner avec les hôtes.

Références d'objet :

- Hôte
- Inventaire de l'hôte
- Tag d'hôte

Méthodes disponibles :

- host.create - créer de nouveaux hôtes
- host.delete - supprimer des hôtes
- host.get - récupérer des hôtes
- host.massadd - ajouter des objets associés aux hôtes
- host.massremove - supprimer des objets associés des hôtes
- host.update - mettre à jour des hôtes

Objet hôte

Les objets suivants sont directement liés à l'API des hôtes.

Hôte

L'objet hôte possède les propriétés suivantes.

Property	Type	Description
hostid	ID	ID de l'hôte. Comportement de la propriété : - *en lecture seule* - *obligatoire* pour les opérations de mise à jour
host	string	Nom technique de l'hôte. Comportement de la propriété : - *obligatoire* pour les opérations de création
description	text	Description de l'hôte.
flags	integer	Origine de l'hôte. Valeurs possibles : 0 - hôte standard ; 4 - hôte converti à partir d'un prototype. Comportement de la propriété : - *en lecture seule*
inventory_mode	integer	Mode de remplissage de l'inventaire de l'hôte. Valeurs possibles : -1 - *(par défaut)* désactivé ; 0 - manuel ; 1 - automatique.
ipmi_authtype	integer	Algorithme d'authentification IPMI. Valeurs possibles : -1 - *(par défaut)* par défaut ; 0 - aucun ; 1 - MD2 ; 2 - MD5 4 - direct ; 5 - OEM ; 6 - RMCP+.
ipmi_password	string	Mot de passe IPMI.
ipmi_privilege	integer	Niveau de privilège IPMI. Valeurs possibles : 1 - rappel ; 2 - *(par défaut)* utilisateur ; 3 - opérateur ; 4 - administrateur ; 5 - OEM.
ipmi_username	string	Nom d'utilisateur IPMI.
maintenance_from	timestamp	Heure de début de la maintenance effective. Comportement de la propriété : - *en lecture seule*

Property	Type	Description
maintenance_status	integer	État de maintenance effectif. Valeurs possibles : 0 - *(par défaut)* aucune maintenance ; 1 - maintenance en cours. Comportement de la propriété : - *en lecture seule*
maintenance_type	integer	Type de maintenance effectif. Valeurs possibles : 0 - *(par défaut)* maintenance avec collecte de données ; 1 - maintenance sans collecte de données. Comportement de la propriété : - *en lecture seule*
maintenanceid	ID	ID de la maintenance actuellement en cours sur l'hôte. Comportement de la propriété : - *en lecture seule*
name	string	Nom visible de l'hôte. Valeur par défaut : valeur de la propriété `host`.
monitored_by	integer	Source utilisée pour superviser l'hôte. Valeurs possibles : 0 - *(par défaut)* serveur Zabbix ; 1 - proxy ; 2 - groupe de proxys. Comportement de la propriété : - *obligatoire* si `select_server_for_monitoring` est défini sur « Disabled ».
proxyid	ID	ID du proxy utilisé pour superviser l'hôte. Comportement de la propriété : - *obligatoire* si `monitored_by` est défini sur « Proxy »
proxy_groupid	ID	ID du groupe de proxys utilisé pour superviser l'hôte. Comportement de la propriété : - *obligatoire* si `monitored_by` est défini sur « Proxy group »
status	integer	État et fonction de l'hôte. Valeurs possibles : 0 - *(par défaut)* activé ; 1 - désactivé.
tls_connect	integer	Connexions vers l'hôte. Valeurs possibles : 1 - *(par défaut)* aucun chiffrement ; 2 - PSK ; 4 - certificat.
tls_accept	integer	Connexions depuis l'hôte. Valeurs bitmap possibles : 1 - *(par défaut)* aucun chiffrement ; 2 - PSK ; 4 - certificat. Il s'agit d'un champ de masque de bits ; toute somme de valeurs bitmap possibles est acceptée (par exemple, 6 pour PSK et certificat).

Property	Type	Description
tls_issuer	string	Émetteur du certificat.
tls_subject	string	Sujet du certificat.
tls_psk_identity	string	Identité PSK ; doit être associée à une seule PSK (parmi l'enregistrement automatique, les hôtes et les proxys). N'incluez pas d'informations sensibles dans l'identité PSK, car celle-ci est envoyée sans chiffrement sur le réseau afin d'indiquer au destinataire quelle PSK utiliser. Comportement de la propriété : - *en écriture seule* - *obligatoire* si `tls_connect` est défini sur « PSK » ou si `tls_accept` contient le bit « PSK »
tls_psk	string	Clé prépartagée (PSK) ; doit comporter au moins 32 chiffres hexadécimaux. Comportement de la propriété : - *en écriture seule* - *obligatoire* si `tls_connect` est défini sur « PSK » ou si `tls_accept` contient le bit « PSK »
active_available	integer	État de disponibilité de l'interface active de l'hôte. Valeurs possibles : 0 - l'état de l'interface est inconnu ; 1 - l'interface est disponible ; 2 - l'interface n'est pas disponible. Comportement de la propriété : - *en lecture seule*
assigned_proxyid	ID	ID du proxy attribué par le serveur Zabbix si l'hôte est supervisé par un groupe de proxys. Comportement de la propriété : - *en lecture seule*

Inventaire de l'hôte

L'objet d'inventaire de l'hôte possède les propriétés suivantes.

Note:
Chaque propriété possède son propre numéro d'identification unique, utilisé pour associer les champs d'inventaire de l'hôte aux éléments.

ID	Propriété	Type	Description	Longueur maximale
4	alias	string	Alias.	128 caractères
11	asset_tag	string	Étiquette d'inventaire.	64 caractères
28	chassis	string	Châssis.	64 caractères
23	contact	string	Personne à contacter.	65535 caractères pour les bases de données SQL
32	contract_number	string	Numéro de contrat.	64 caractères
47	date_hw_decomm	string	Date de mise hors service du matériel.	64 caractères
46	date_hw_expiry	string	Date d'expiration de la maintenance du matériel.	64 caractères
45	date_hw_install	string	Date d'installation du matériel.	64 caractères
44	date_hw_purchase	string	Date d'achat du matériel.	64 caractères
34	deployment_status	string	Statut de déploiement.	64 caractères
14	hardware	string	Matériel.	255 caractères
15	hardware_full	string	Détails du matériel.	65535 caractères pour les bases de données SQL
39	host_netmask	string	Masque de sous-réseau de l'hôte.	39 caractères

ID	Propriété	Type	Description	Longueur maximale
38	host_networks	string	Réseaux de l’hôte.	65535 caractères pour les bases de données SQL
40	host_router	string	Routeur de l’hôte.	39 caractères
30	hw_arch	string	Architecture matérielle.	32 caractères
33	installer_name	string	Nom de l’installateur.	64 caractères
24	location	string	Emplacement.	65535 caractères pour les bases de données SQL
25	location_lat	string	Latitude de l’emplacement.	16 caractères
26	location_lon	string	Longitude de l’emplacement.	16 caractères
12	macaddress_a	string	Adresse MAC A.	64 caractères
13	macaddress_b	string	Adresse MAC B.	64 caractères
29	model	string	Modèle.	64 caractères
3	name	string	Nom.	128 caractères
27	notes	string	Notes.	65535 caractères pour les bases de données SQL
41	oob_ip	string	Adresse IP OOB.	39 caractères
42	oob_netmask	string	Masque de sous-réseau de l’hôte OOB.	39 caractères
43	oob_router	string	Routeur OOB.	39 caractères
5	os	string	Nom du système d’exploitation.	128 caractères
6	os_full	string	Nom détaillé du système d’exploitation.	255 caractères
7	os_short	string	Nom court du système d’exploitation.	128 caractères
61	poc_1_cell	string	Numéro de mobile du contact principal.	64 caractères
58	poc_1_email	string	E-mail principal.	128 caractères
57	poc_1_name	string	Nom du contact principal.	128 caractères
63	poc_1_notes	string	Notes du contact principal.	65535 caractères pour les bases de données SQL
59	poc_1_phone_a	string	Téléphone A du contact principal.	64 caractères
60	poc_1_phone_b	string	Téléphone B du contact principal.	64 caractères
62	poc_1_screen	string	Nom d’écran du contact principal.	64 caractères
68	poc_2_cell	string	Numéro de mobile du contact secondaire.	64 caractères
65	poc_2_email	string	E-mail secondaire.	128 caractères
64	poc_2_name	string	Nom du contact secondaire.	128 caractères
70	poc_2_notes	string	Notes du contact secondaire.	65535 caractères pour les bases de données SQL
66	poc_2_phone_a	string	Téléphone A du contact secondaire.	64 caractères
67	poc_2_phone_b	string	Téléphone B du contact secondaire.	64 caractères
69	poc_2_screen	string	Nom d’écran du contact secondaire.	64 caractères
8	serialno_a	string	Numéro de série A.	64 caractères
9	serialno_b	string	Numéro de série B.	64 caractères
48	site_address_a	string	Adresse du site A.	128 caractères
49	site_address_b	string	Adresse du site B.	128 caractères
50	site_address_c	string	Adresse du site C.	128 caractères
51	site_city	string	Ville du site.	128 caractères
53	site_country	string	Pays du site.	64 caractères
56	site_notes	string	Notes du site.	65535 caractères pour les bases de données SQL
55	site_rack	string	Emplacement de baie du site.	128 caractères
52	site_state	string	État/région du site.	64 caractères
54	site_zip	string	Code postal du site.	64 caractères
16	software	string	Logiciel.	255 caractères
18	software_app_a	string	Application logicielle A.	64 caractères
19	software_app_b	string	Application logicielle B.	64 caractères
20	software_app_c	string	Application logicielle C.	64 caractères
21	software_app_d	string	Application logicielle D.	64 caractères
22	software_app_e	string	Application logicielle E.	64 caractères
17	software_full	string	Détails du logiciel.	65535 caractères pour les bases de données SQL
10	tag	string	Étiquette.	64 caractères
1	type	string	Type.	64 caractères
2	type_full	string	Détails du type.	64 caractères

ID	Propriété	Type	Description	Longueur maximale
35	url_a	string	URL A.	2048 caractères
36	url_b	string	URL B.	2048 caractères
37	url_c	string	URL C.	2048 caractères
31	vendor	string	Fournisseur.	64 caractères

Balise d'hôte

L'objet de balise d'hôte possède les propriétés suivantes.

Propriété	Type	Description
tag	string	Nom de la balise d'hôte. Comportement de la propriété : - *obligatoire*
value	string	Valeur de la balise d'hôte. Comportement de la propriété : - *lecture seule*.
automatic	integer	Type de balise d'hôte. Valeurs possibles : 0 - *(par défaut)* manuel (balise créée par l'utilisateur) ; 1 - automatique (balise créée par la découverte de bas niveau) Comportement de la propriété : - *lecture seule*
object	integer	Type d'objet dont la balise a été héritée. Valeurs possibles : 0 - modèle. Comportement de la propriété : - *lecture seule*.
objectid	ID	ID de l'objet dont la balise a été héritée. Comportement de la propriété : - *lecture seule*.

host.create

Description

`object host.create(object/array hosts)`

Cette méthode vous permet de créer de nouveaux hôtes.

> **Note:**
> Cette méthode est disponible uniquement pour les types d'utilisateur *Admin* et *Super admin*. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres des rôles utilisateur. Consultez Rôles utilisateur pour plus d'informations.

Paramètres

`(object/array)` Hôtes à créer.

En plus des propriétés standard de l'hôte, la méthode accepte les paramètres suivants.

Paramètre	Type	Description
groups	object/array	Groupes d'hôtes auxquels ajouter l'hôte. Les groupes d'hôtes doivent avoir uniquement la propriété `groupid` définie. Comportement des paramètres : - *obligatoire*
interfaces	object/array	Interfaces à créer pour l'hôte.
tags	object/array	Tags d'hôte.
templates	object/array	Modèles à lier à l'hôte. Les modèles doivent avoir uniquement la propriété `templateid` définie.
macros	object/array	Macros utilisateur à créer pour l'hôte.
inventory	object	Propriétés de l'inventaire de l'hôte.

Valeurs retournées

`(object)` Retourne un objet contenant les identifiants des hôtes créés sous la propriété `hostids`. L'ordre des ID retournés correspond à l'ordre des hôtes transmis.

Exemples

Création d'un hôte

Créez un hôte nommé « Linux server » avec une interface IP et des tags, ajoutez-le à un groupe, liez-lui un modèle et définissez les adresses MAC dans l'inventaire de l'hôte.

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "host.create",
    "params": {
        "host": "Linux server",
        "interfaces": [
            {
                "type": 1,
                "main": 1,
                "useip": 1,
                "ip": "192.168.3.1",
                "dns": "",
                "port": "10050"
            }
        ],
        "groups": [
            {
                "groupid": "50"
            }
        ],
        "tags": [
            {
                "tag": "host-name",
                "value": "linux-server"
            }
        ],
        "templates": [
            {
                "templateid": "20045"
            }
        ],
        "macros": [
            {
                "macro": "{$USER_ID}",
```

```
                "value": "123321"
            },
            {
                "macro": "{$USER_LOCATION}",
                "value": "0:0:0",
                "description": "latitude, longitude and altitude coordinates"
            }
        ],
        "inventory_mode": 0,
        "inventory": {
            "macaddress_a": "01234",
            "macaddress_b": "56768"
        }
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "107819"
        ]
    },
    "id": 1
}
```

Création d'un hôte avec une interface SNMP

Créez un hôte nommé « SNMP host » avec une interface SNMPv3 avec des détails.

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "host.create",
    "params": {
        "host": "SNMP host",
        "interfaces": [
            {
                "type": 2,
                "main": 1,
                "useip": 1,
                "ip": "127.0.0.1",
                "dns": "",
                "port": "161",
                "details": {
                    "version": 3,
                    "bulk": 0,
                    "securityname": "mysecurityname",
                    "contextname": "",
                    "securitylevel": 1
                }
            }
        ],
        "groups": [
            {
                "groupid": "4"
            }
        ]
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "10658"
        ]
    },
    "id": 1
}
```

Création d'un hôte avec chiffrement PSK

Créez un hôte appelé "PSK host" avec le chiffrement PSK configuré. Notez que l'hôte doit être préconfiguré pour utiliser PSK.

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "host.create",
    "params": {
        "host": "PSK host",
        "interfaces": [
            {
                "type": 1,
                "ip": "192.168.3.1",
                "dns": "",
                "port": "10050",
                "useip": 1,
                "main": 1
            }
        ],
        "groups": [
            {
                "groupid": "2"
            }
        ],
        "tls_accept": 2,
        "tls_connect": 2,
        "tls_psk_identity": "PSK 001",
        "tls_psk": "1f87b595725ac58dd977beef14b97461a7c1045b9a1c963065002c5473194952"
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "10590"
        ]
    },
    "id": 1
}
```

Création d'un hôte surveillé par un proxy

Créez un hôte qui sera surveillé par le proxy avec l'ID « 1 ».

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "host.create",
    "params": {
```

```
        "host": "Host monitored by proxy",
        "groups": [
            {
                "groupid": "2"
            }
        ],
        "monitored_by": 1,
        "proxyid": 1
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "10591"
        ]
    },
    "id": 1
}
```

Création d'un hôte surveillé par un groupe de proxys

Créez un hôte qui sera surveillé par le groupe de proxys avec l'ID « 1 ».

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "host.create",
    "params": {
        "host": "Hôte surveillé par un groupe de proxys",
        "groups": [
            {
                "groupid": "2"
            }
        ],
        "monitored_by": 2,
        "proxy_groupid": 1
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "10592"
        ]
    },
    "id": 1
}
```

Voir aussi

- Groupe d'hôtes
- Modèle
- Macro utilisateur
- Interface d'hôte
- Inventaire de l'hôte
- Tag d'hôte
- Proxy

- Groupe de proxies

Source

CHost::create() dans *ui/include/classes/api/services/CHost.php*.

host.delete

Description

`object host.delete(array hosts)`

Cette méthode vous permet de supprimer des hôtes.

> **Note:**
> Cette méthode est disponible uniquement pour les types d'utilisateur *Admin* et *Super admin*. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres des rôles utilisateur. Consultez Rôles utilisateur pour plus d'informations.

Paramètres

`(array)` ID des hôtes à supprimer.

Valeurs de retour

`(object)` Renvoie un objet contenant les ID des hôtes supprimés dans la propriété `hostids`.

Exemples

Suppression de plusieurs hôtes

Supprimez deux hôtes.

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "host.delete",
    "params": [
        "13",
        "32"
    ],
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "13",
            "32"
        ]
    },
    "id": 1
}
```

Source

CHost::delete() dans *ui/include/classes/api/services/CHost.php*.

host.get

Description

`integer/array host.get(object parameters)`

Cette méthode permet de récupérer des hôtes selon les paramètres fournis.

Note:
Cette méthode est disponible pour les utilisateurs de tout type. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres du rôle utilisateur. Consultez Rôles utilisateur pour plus d'informations.

Paramètres

`(object)` Paramètres définissant la sortie souhaitée.

La méthode prend en charge les paramètres suivants.

Paramètre	Type	Description
groupids	ID/array	Renvoie uniquement les hôtes qui appartiennent aux groupes indiqués.
dserviceids	ID/array	Renvoie uniquement les hôtes liés aux services découverts indiqués.
graphids	ID/array	Renvoie uniquement les hôtes qui possèdent les graphiques indiqués.
hostids	ID/array	Renvoie uniquement les hôtes ayant les ID d'hôte indiqués.
httptestids	ID/array	Renvoie uniquement les hôtes qui possèdent les contrôles web indiqués.
interfaceids	ID/array	Renvoie uniquement les hôtes qui utilisent les interfaces indiquées.
itemids	ID/array	Renvoie uniquement les hôtes qui possèdent les éléments indiqués.
maintenanceids	ID/array	Renvoie uniquement les hôtes affectés par les maintenances indiquées.
monitored_hosts	flag	Renvoie uniquement les hôtes surveillés.
proxyids	ID/array	Renvoie uniquement les hôtes surveillés par les proxies indiqués.
proxy_groupids	ID/array	Renvoie uniquement les hôtes surveillés par les groupes de proxies indiqués.
templated_hosts	flag	Renvoie à la fois les hôtes et les modèles.
templateids	ID/array	Renvoie uniquement les hôtes liés aux modèles indiqués.
triggerids	ID/array	Renvoie uniquement les hôtes qui possèdent les déclencheurs indiqués.
with_items	flag	Renvoie uniquement les hôtes qui possèdent des éléments. Remplace les paramètres `with_monitored_items` et `with_simple_graph_items`.
with_item_prototypes	flag	Renvoie uniquement les hôtes qui possèdent des prototypes d'élément. Remplace le paramètre `with_simple_graph_item_prototypes`.
with_simple_graph_item_prototypes	flag	Renvoie uniquement les hôtes qui possèdent des prototypes d'élément activés pour la création et ayant un type d'information numérique.
with_graphs	flag	Renvoie uniquement les hôtes qui possèdent des graphiques.
with_graph_prototypes	flag	Renvoie uniquement les hôtes qui possèdent des prototypes de graphique.
with_httptests	flag	Renvoie uniquement les hôtes qui possèdent des contrôles web. Remplace le paramètre `with_monitored_httptests`.
with_monitored_httptests	flag	Renvoie uniquement les hôtes qui possèdent des contrôles web activés.
with_monitored_items	flag	Renvoie uniquement les hôtes qui possèdent des éléments activés. Remplace le paramètre `with_simple_graph_items`.
with_monitored_triggers	flag	Renvoie uniquement les hôtes qui possèdent des déclencheurs activés. Tous les éléments utilisés dans le déclencheur doivent également être activés.
with_simple_graph_items	flag	Renvoie uniquement les hôtes qui possèdent des éléments avec un type d'information numérique.
with_triggers	flag	Renvoie uniquement les hôtes qui possèdent des déclencheurs. Remplace le paramètre `with_monitored_triggers`.
withProblemsSuppressed	boolean	Si défini sur `true`, renvoie uniquement les hôtes avec des problèmes supprimés.

Paramètre	Type	Description
evaltype	integer	Méthode d'évaluation des tags. Valeurs possibles : 0 - *(par défaut)* Et/Ou ; 2 - Ou.
severities	integer/array	Renvoie les hôtes qui ont uniquement des problèmes avec les sévérités indiquées. S'applique uniquement si l'objet problème est un déclencheur.
tags	object/array	Renvoie uniquement les hôtes avec les tags indiqués. Format : `[{"tag": "<tag>", "value": "<value>", "operator": "<operator>"}, ...]`. Un tableau vide renvoie tous les hôtes. Valeurs possibles de operator : 0 - *(par défaut)* Contient ; 1 - Égal ; 2 - Ne contient pas ; 3 - Différent de ; 4 - Existe ; 5 - N'existe pas.
inheritedTags	boolean	Renvoie les hôtes qui possèdent les `tags` indiqués également dans tous leurs modèles liés. Valeurs possibles : `true` - les modèles liés doivent également posséder les `tags` indiqués ; `false` - *(par défaut)* les tags des modèles liés sont ignorés.
selectDiscoveryData	query	Renvoie une propriété `discoveryData` avec les données de l'objet de découverte d'hôte. L'objet de découverte d'hôte relie un hôte découvert à un prototype d'hôte à partir duquel il a été découvert. Il possède les propriétés suivantes : `host` - `(string)` ID de l'hôte ; `parent_hostid` - `(string)` ID du prototype d'hôte à partir duquel l'hôte a été créé ; `status` - `(int)` état de découverte de l'hôte : 0 - *(par défaut)* l'hôte est découvert, 1 - l'hôte n'est plus découvert ; `ts_delete` - `(timestamp)` heure à laquelle un hôte qui n'est plus découvert sera supprimé ; `ts_disable` - `(timestamp)` heure à laquelle un hôte qui n'est plus découvert sera désactivé ; `disable_source` - `(int)` indicateur précisant si l'hôte a été désactivé par une règle LLD ou manuellement : 0 - *(par défaut)* désactivé automatiquement, 1 - désactivé par une règle LLD.
selectDiscoveryRule	query	Renvoie une propriété `discoveryRule` avec la règle de découverte de bas niveau qui a créé l'hôte (à partir d'un prototype d'hôte dans la supervision VMware).
selectDiscoveryRules	query	Renvoie une propriété `discoveryRules` avec les règles LLD de l'hôte. Prend en charge `count`.
selectGraphs	query	Renvoie une propriété `graphs` avec les graphiques de l'hôte. Prend en charge `count`.
selectHostGroups	query	Renvoie une propriété `hostgroups` avec les données des groupes d'hôtes auxquels l'hôte appartient.
selectHttpTests	query	Renvoie une propriété `httpTests` avec les scénarios web de l'hôte. Prend en charge `count`.

Paramètre	Type	Description
selectInterfaces	query	Renvoie une propriété `interfaces` avec les interfaces de l'hôte. Prend en charge `count`.
selectInventory	query	Renvoie une propriété `inventory` avec les données d'inventaire de l'hôte.
selectItems	query	Renvoie une propriété `items` avec les éléments de l'hôte. Prend en charge `count`.
selectMacros	query	Renvoie une propriété `macros` avec les macros de l'hôte.
selectParentTemplates	query	Renvoie une propriété `parentTemplates` avec les modèles auxquels l'hôte est lié. En plus des champs de l'objet Template, elle contient `link_type` - `(integer)` la manière dont le modèle est lié à l'hôte. Valeurs possibles : 0 - *(par défaut)* lié manuellement ; 1 - lié automatiquement par LLD. Prend en charge `count`.
selectDashboards	query	Renvoie une propriété `dashboards`. Prend en charge `count`.
selectTags	query	Renvoie une propriété `tags` avec les tags de l'hôte.
selectInheritedTags	query	Renvoie une propriété `inheritedTags` avec les tags présents sur tous les modèles liés à l'hôte.
selectTriggers	query	Renvoie une propriété `triggers` avec les déclencheurs de l'hôte. Prend en charge `count`.
selectValueMaps	query	Renvoie une propriété `valuemaps` avec les correspondances de valeurs de l'hôte.
filter	object	Renvoie uniquement les résultats qui correspondent exactement au filtre indiqué. Accepte un objet où les clés sont des noms de propriété et les valeurs sont soit une valeur unique, soit un tableau de valeurs à faire correspondre. Ne prend pas en charge les propriétés de type de données `text` data type. Prend en charge des propriétés supplémentaires : propriétés de l'interface d'hôte.
limitSelects	integer	Limite le nombre d'enregistrements renvoyés par les sous-sélections. S'applique aux sous-sélections suivantes : `selectParentTemplates` - les résultats seront triés par `host` ; `selectInterfaces` ; `selectItems` - triés par `name` ; `selectDiscoveryRules` - triés par `name` ; `selectTriggers` - triés par `description` ; `selectGraphs` - triés par `name` ; `selectDashboards` - triés par `name`.

Paramètre	Type	Description
search	object	Renvoie les résultats qui correspondent au motif indiqué (insensible à la casse). Accepte un objet où les clés sont des noms de propriété et les valeurs sont des chaînes à rechercher. Si aucune option supplémentaire n'est fournie, cela effectuera une recherche `LIKE "%…%"`. Prend en charge uniquement les propriétés de type de données `string` et `text` data type. Prend en charge des propriétés supplémentaires : propriétés de l'interface d'hôte.
searchInventory	object	Renvoie les hôtes dont les données d'inventaire correspondent au motif indiqué (insensible à la casse). Accepte un objet où les clés sont des noms de propriété et les valeurs sont des chaînes à rechercher. Si aucune option supplémentaire n'est fournie, cela effectuera une recherche `LIKE "%…%"`. Prend en charge uniquement les propriétés de type de données `string` et `text` data type.
sortfield	string/array	Trie le résultat selon les propriétés indiquées. Valeurs possibles : `hostid`, `host`, `name`, `status`.
countOutput	boolean	Ces paramètres sont décrits dans le commentaire de référence.
editable	boolean	
excludeSearch	boolean	
limit	integer	
output	query	
preservekeys	boolean	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	
selectDiscoveries	query	Renvoie une propriété `discoveries` avec les règles de découverte de bas niveau de l'hôte. Prend en charge `count`. Cette requête est **obsolète**, veuillez utiliser `selectDiscoveryRules` à la place.
selectHostDiscovery	query	Renvoie une propriété `hostDiscovery` avec les données de l'objet de découverte d'hôte. Cette requête est **obsolète**, veuillez utiliser `selectDiscoveryData` à la place.

Valeurs retournées

`(integer/array)` Retourne soit :

- un tableau d'objets ;
- le nombre d'objets récupérés, si le paramètre `countOutput` a été utilisé.

Exemples

Récupération des données par nom

Récupérez toutes les données concernant deux hôtes nommés "Zabbix server" et "Linux server".

Requête:

```
{
    "jsonrpc": "2.0",
```

```
    "method": "host.get",
    "params": {
        "filter": {
            "host": [
                "Zabbix server",
                "Linux server"
            ]
        }
    },
    "id": 1
}
```

Réponse :


```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "hostid": "10160",
            "proxyid": "0",
            "host": "Zabbix server",
            "status": "0",
            "ipmi_authtype": "-1",
            "ipmi_privilege": "2",
            "ipmi_username": "",
            "ipmi_password": "",
            "maintenanceid": "0",
            "maintenance_status": "0",
            "maintenance_type": "0",
            "maintenance_from": "0",
            "name": "Zabbix server",
            "flags": "0",
            "description": "Le serveur de supervision Zabbix.",
            "tls_connect": "1",
            "tls_accept": "1",
            "tls_issuer": "",
            "tls_subject": "",
            "proxy_groupid": "0",
            "monitored_by": "0",
            "inventory_mode": "1",
            "active_available": "1",
            "assigned_proxyid": "0"
        },
        {
            "hostid": "10167",
            "proxyid": "0",
            "host": "Linux server",
            "status": "0",
            "ipmi_authtype": "-1",
            "ipmi_privilege": "2",
            "ipmi_username": "",
            "ipmi_password": "",
            "maintenanceid": "0",
            "maintenance_status": "0",
            "maintenance_type": "0",
            "maintenance_from": "0",
            "name": "Linux server",
            "flags": "0",
            "description": "",
            "tls_connect": "1",
            "tls_accept": "1",
            "tls_issuer": "",
            "tls_subject": "",
```

```
            "proxy_groupid": "0",
            "monitored_by": "0",
            "inventory_mode": "1",
            "active_available": "1",
            "assigned_proxyid": "0"
        }
    ],
    "id": 1
}
```

Récupération des groupes d'hôtes

Récupérez les groupes d'hôtes dont l'hôte "Zabbix server" est membre.

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "host.get",
    "params": {
        "output": ["hostid"],
        "selectHostGroups": "extend",
        "filter": {
            "host": [
                "Zabbix server"
            ]
        }
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "hostid": "10085",
            "hostgroups": [
                {
                    "groupid": "2",
                    "name": "Linux servers",
                    "flags": "0",
                    "uuid": "dc579cd7a1a34222933f24f52a68bcd8"
                },
                {
                    "groupid": "4",
                    "name": "Zabbix servers",
                    "flags": "0",
                    "uuid": "6f6799aa69e844b4b3918f779f2abf08"
                }
            ]
        }
    ],
    "id": 1
}
```

Récupération des modèles liés

Récupérez les ID et les noms des modèles liés à l'hôte « 10084 ».

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "host.get",
    "params": {
```

```
        "output": ["hostid"],
        "selectParentTemplates": [
            "templateid",
            "name"
        ],
        "hostids": "10084"
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "hostid": "10084",
            "parentTemplates": [
                {
                    "name": "Linux",
                    "templateid": "10001"
                },
                {
                    "name": "Zabbix Server",
                    "templateid": "10047"
                }
            ]
        }
    ],
    "id": 1
}
```

Récupération des hôtes par modèle

Récupérez les hôtes auxquels le modèle « 10001 » (*Linux by Zabbix agent*) est lié.

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "host.get",
    "params": {
        "output": ["hostid", "name"],
        "templateids": "10001"
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "templateid": "10001",
            "hosts": [
                {
                    "hostid": "10084",
                    "name": "Zabbix server"
                },
                {
                    "hostid": "10603",
                    "name": "Host 1"
                },
                {
                    "hostid": "10604",
```

```
                    "name": "Host 2"
                }
            ]
        }
    ],
    "id": 1
}
```

Recherche par données d'inventaire de l'hôte

Récupérez les hôtes qui contiennent « Linux » dans le champ « OS » de l'inventaire de l'hôte.

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "host.get",
    "params": {
        "output": [
            "host"
        ],
        "selectInventory": [
            "os"
        ],
        "searchInventory": {
            "os": "Linux"
        }
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "hostid": "10084",
            "host": "Zabbix server",
            "inventory": {
                "os": "Linux Ubuntu"
            }
        },
        {
            "hostid": "10107",
            "host": "Linux server",
            "inventory": {
                "os": "Linux Mint"
            }
        }
    ],
    "id": 1
}
```

Recherche par tags d'hôte

Récupérez les hôtes ayant le tag "host-name" égal à "linux-server".

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "host.get",
    "params": {
        "output": ["hostid"],
        "selectTags": "extend",
        "evaltype": 0,
```

```
        "tags": [
            {
                "tag": "host-name",
                "value": "linux-server",
                "operator": 1
            }
        ]
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "hostid": "10085",
            "tags": [
                {
                    "tag": "host-name",
                    "value": "linux-server",
                    "automatic": "0"
                },
                {
                    "tag": "os",
                    "value": "rhel-7",
                    "automatic": "0"
                }
            ]
        }
    ],
    "id": 1
}
```

Récupérez les hôtes ayant ces tags non seulement au niveau de l'hôte, mais aussi dans leurs modèles parents liés.

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "host.get",
    "params": {
        "output": ["name"],
        "tags": [
            {
                "tag": "os",
                "value": "rhel-7",
                "operator": 1
            }
        ],
        "inheritedTags": true
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "hostid": "10623",
            "name": "PC room 1"
        },
```

```
        {
            "hostid": "10601",
            "name": "Office"
        }
    ],
    "id": 1
}
```

Recherche d'un hôte avec des tags et des tags de modèle

Récupérez un hôte avec ses tags et tous les tags liés aux modèles parents.

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "host.get",
    "params": {
        "output": ["name"],
        "hostids": 10502,
        "selectTags": ["tag", "value"],
        "selectInheritedTags": ["tag", "value"]
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "hostid": "10502",
            "name": "Desktop",
            "tags": [
                {
                    "tag": "host-name",
                    "value": "linux-server"
                },
                {
                    "tag": "os",
                    "value": "rhel-7"
                }
            ],
            "inheritedTags": [
                {
                    "tag": "class",
                    "value": "os"
                },
                {
                    "tag": "target",
                    "value": "linux"
                },
                {
                    "tag": "os",
                    "value": "rhel-7"
                }
            ]
        }
    ],
    "id": 1
}
```

Recherche d'hôtes par gravité de problème

Récupérez les hôtes qui ont des problèmes de niveau « Désastre ».

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "host.get",
    "params": {
        "output": ["name"],
        "severities": 5
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "hostid": "10160",
            "name": "Zabbix server"
        }
    ],
    "id": 1
}
```

Récupérez les hôtes qui ont des problèmes de niveau « Moyen » et « Élevé ».

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "host.get",
    "params": {
        "output": ["name"],
        "severities": [3, 4]
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "hostid": "20170",
            "name": "Database"
        },
        {
            "hostid": "20183",
            "name": "workstation"
        }
    ],
    "id": 1
}
```

Récupération des éléments de l'hôte

Récupérez les éléments de l'hôte "Zabbix server". La requête est limitée aux propriétés d'élément `itemid`, `name` et `status`. En raison de la taille importante de la réponse, seul un sous-ensemble des éléments est affiché dans l'exemple.

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "host.get",
    "params": {
        "output": ["hostid"],
```

```
        "selectItems": ["itemid","name","status"],
        "filter": {
            "host": [
                "Zabbix server"
            ]
        }
    },
    "id": 1
}
```

Réponse :


```
{
  "jsonrpc": "2.0",
  "result": [
    {
      "hostid": "10084",
      "items": [
        {
          "itemid": "42227",
          "name": "Nom d'hôte du Zabbix agent en cours d'exécution",
          "status": "0"
        },
        {
          "itemid": "42237",
          "name": "Ping du Zabbix agent",
          "status": "0"
        },
        {
          "itemid": "42250",
          "name": "Version du Zabbix agent en cours d'exécution",
          "status": "0"
        },
        {
          "itemid": "42251",
          "name": "Nombre maximal de descripteurs de fichiers ouverts",
          "status": "0"
        },
        {
          "itemid": "42252",
          "name": "Nombre maximal de processus",
          "status": "0"
        },
        {
          "itemid": "69869",
          "name": "Interface enp0s3 : Paquets entrants rejetés",
          "status": "0"
        },
        {
          "itemid": "69870",
          "name": "Interface enp0s3 : Paquets entrants avec erreurs",
          "status": "0"
        },
        {
          "itemid": "69871",
          "name": "Interface enp0s3 : Bits reçus",
          "status": "0"
        },
        {
          "itemid": "69872",
          "name": "Interface enp0s3 : Paquets sortants rejetés",
          "status": "0"
        },
```

```
        {
          "itemid": "69873",
          "name": "Interface enp0s3 : Paquets sortants avec erreurs",
          "status": "0"
        },
        {
          "itemid": "69874",
          "name": "Interface enp0s3 : Bits envoyés",
          "status": "0"
        },
        {
          "itemid": "42253",
          "name": "Nombre de processus",
          "status": "0"
        },
        {
          "itemid": "42254",
          "name": "Nombre de processus en cours d'exécution",
          "status": "0"
        },
        {
          "itemid": "42255",
          "name": "Heure de démarrage du système",
          "status": "0"
        }
      ]
    }
  ],
  "id": 1
}
```


Voir aussi

- Groupe d'hôtes
- Modèle
- Macro utilisateur
- Interface d'hôte
- Proxy
- Groupe de proxies

Source

CHost::get() dans *ui/include/classes/api/services/CHost.php*.

host.massadd

Description

`object host.massadd(object parameters)`

Cette méthode permet d'ajouter simultanément plusieurs objets associés à tous les hôtes spécifiés.

> **Note:**
> Cette méthode est disponible uniquement pour les types d'utilisateur *Admin* et *Super admin*. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres des rôles utilisateur. Consultez Rôles utilisateur pour plus d'informations.

Paramètres

`(object)` Paramètres contenant les ID des hôtes à mettre à jour et les objets à ajouter à tous les hôtes.

La méthode accepte les paramètres suivants.

Paramètre	Type	Description
hosts	object/array	Hôtes à mettre à jour. Les hôtes doivent avoir uniquement la propriété `hostid` définie. Comportement des paramètres : - *obligatoire*
groups	object/array	Groupes d'hôtes à ajouter aux hôtes donnés. Les groupes d'hôtes doivent avoir uniquement la propriété `groupid` définie.
interfaces	object/array	Interfaces d'hôte à créer pour les hôtes donnés.
macros	object/array	Macros utilisateur à créer pour les hôtes donnés.
templates	object/array	Modèles à lier aux hôtes donnés. Les modèles doivent avoir uniquement la propriété `templateid` définie.

Valeurs retournées

`(object)` Retourne un objet contenant les identifiants des hôtes mis à jour sous la propriété `hostids`.

Exemples

Ajout de macros

Ajoutez deux nouvelles macros à deux hôtes.

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "host.massadd",
    "params": {
        "hosts": [
            {
                "hostid": "10160"
            },
            {
                "hostid": "10167"
            }
        ],
        "macros": [
            {
                "macro": "{$TEST1}",
                "value": "MACROTEST1"
            },
            {
                "macro": "{$TEST2}",
                "value": "MACROTEST2",
                "description": "Description de test"
            }
        ]
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "10160",
            "10167"
```

```
        ]
    },
    "id": 1
}
```

Voir aussi

- host.update
- Groupe d'hôtes
- Modèle
- Macro utilisateur
- Interface d'hôte

Source

CHost::massAdd() dans *ui/include/classes/api/services/CHost.php*.

host.massremove

Description

`object host.massremove(object parameters)`

Cette méthode vous permet de supprimer des objets associés de plusieurs hôtes.

Note:
Cette méthode est disponible uniquement pour les types d'utilisateur *Admin* et *Super admin*. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres du rôle utilisateur. Consultez Rôles d'utilisateur pour plus d'informations.

Paramètres

`(object)` Paramètres contenant les ID des hôtes à mettre à jour et les objets qui doivent être supprimés.

Paramètre	Type	Description
hostids	ID/array	ID des hôtes à mettre à jour. Comportement des paramètres : - *obligatoire*
groupids	ID/array	ID des groupes d'hôtes desquels supprimer les hôtes donnés.
interfaces	object/array	Interfaces d'hôte à supprimer des hôtes donnés. L'objet d'interface d'hôte doit avoir uniquement les propriétés `ip`, `dns` et `port` définies.
macros	string/array	Macros utilisateur à supprimer des hôtes donnés.
templateids	ID/array	ID des modèles à dissocier des hôtes donnés.
templateids_clear	ID/array	ID des modèles à dissocier et supprimer des hôtes donnés.

Valeurs retournées

`(object)` Retourne un objet contenant les identifiants des hôtes mis à jour sous la propriété `hostids`.

Exemples

Dissociation de modèles

Dissociez un modèle de deux hôtes et supprimez toutes les entités issues du modèle.

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "host.massremove",
    "params": {
        "hostids": ["69665", "69666"],
        "templateids_clear": "325"
```

```
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "69665",
            "69666"
        ]
    },
    "id": 1
}
```

Voir aussi

- host.update
- Macro utilisateur
- Interface d'hôte

Source

CHost::massRemove() dans *ui/include/classes/api/services/CHost.php*.

host.update

Description

`object host.update(object/array hosts)`

Cette méthode vous permet de mettre à jour des hôtes existants.

Note:
Cette méthode est disponible uniquement pour les types d'utilisateur *Admin* et *Super admin*. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres du rôle utilisateur. Consultez Rôles d'utilisateur pour plus d'informations.

Paramètres

`(object/array)` Propriétés de l'hôte à mettre à jour.

La propriété `hostid` doit être définie pour chaque hôte, toutes les autres propriétés sont facultatives. Seules les propriétés fournies seront mises à jour, toutes les autres resteront inchangées.

Notez toutefois que la mise à jour du nom technique de l'hôte mettra également à jour le nom visible de l'hôte (s'il n'est pas fourni séparément) avec la valeur du nom technique de l'hôte.

En plus des propriétés standard de l'hôte, la méthode accepte les paramètres suivants.

Parameter	Type	Description
groups	object/array	Groupes d'hôtes pour remplacer les groupes d'hôtes actuels auxquels l'hôte appartient. Tous les groupes d'hôtes qui ne sont pas listés dans la requête seront dissociés. Les groupes d'hôtes doivent avoir uniquement la propriété `groupid` définie.
interfaces	object/array	Interfaces de l'hôte pour remplacer les interfaces actuelles de l'hôte. Toutes les interfaces qui ne sont pas listées dans la requête seront supprimées.
tags	object/array	Tags de l'hôte pour remplacer les tags actuels de l'hôte. Tous les tags qui ne sont pas listés dans la requête seront supprimés.
inventory	object	Propriétés de l'inventaire de l'hôte.

Parameter	Type	Description
macros	object/array	Macros utilisateur pour remplacer les macros utilisateur actuelles. Toutes les macros qui ne sont pas listées dans la requête seront supprimées.
templates	object/array	Modèles pour remplacer les modèles actuellement liés. Tous les modèles qui ne sont pas listés dans la requête seront uniquement dissociés. Les modèles doivent avoir uniquement la propriété `templateid` définie.
templates_clear	object/array	Modèles à dissocier et supprimer de l'hôte. Les modèles doivent avoir uniquement la propriété `templateid` définie.

Valeurs retournées

`(object)` Retourne un objet contenant les identifiants des hôtes mis à jour sous la propriété `hostids`.

Exemples

Activation d'un hôte

Activez la surveillance de l'hôte, c'est-à-dire définissez son statut sur « 0 ».

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "host.update",
    "params": {
        "hostid": "10126",
        "status": 0
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "10126"
        ]
    },
    "id": 1
}
```

Activation de plusieurs hôtes

Activez la surveillance de deux hôtes, c'est-à-dire définissez leur statut sur « 0 ».

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "host.update",
    "params": [
        {
            "hostid": "10127",
            "status": 0
        },
        {
            "hostid": "10128",
            "status": 0
        }
```

```
    ],
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "10127",
            "10128"
        ]
    },
    "id": 1
}
```

Dissociation des modèles

Dissociez et effacez deux modèles de l'hôte.

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "host.update",
    "params": {
        "hostid": "10126",
        "templates_clear": [
            {
                "templateid": "10124"
            },
            {
                "templateid": "10125"
            }
        ]
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "10126"
        ]
    },
    "id": 1
}
```

Mise à jour des macros d'hôte

Remplacez toutes les macros d'hôte par deux nouvelles.

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "host.update",
    "params": {
        "hostid": "10126",
        "macros": [
            {
                "macro": "{$PASS}",
                "value": "password"
            },
```

```
            {
                "macro": "{$DISC}",
                "value": "sda",
                "description": "Description mise à jour"
            }
        ]
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "10126"
        ]
    },
    "id": 1
}
```

Mise à jour de l'inventaire de l'hôte

Modifier le mode d'inventaire et ajouter l'emplacement

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "host.update",
    "params": {
        "hostid": "10387",
        "inventory_mode": 0,
        "inventory": {
            "location": "Latvia, Riga"
        }
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "10387"
        ]
    },
    "id": 1
}
```

Mise à jour des tags d'hôte

Remplacez tous les tags d'hôte par un nouveau.

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "host.update",
    "params": {
        "hostid": "10387",
        "tags": {
            "tag": "os",
            "value": "rhel-7"
        }
```

```
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "10387"
        ]
    },
    "id": 1
}
```

Mise à jour des macros d'hôte découvertes

Convertissez la macro « automatique » créée par la règle de découverte en macro « manuelle » et modifiez sa valeur en « new-value ».

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "host.update",
    "params": {
        "hostid": "10387",
        "macros": {
            "hostmacroid": "5541",
            "value": "new-value",
            "automatic": "0"
        }
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "10387"
        ]
    },
    "id": 1
}
```

Mise à jour du chiffrement de l'hôte

Mettre à jour l'hôte "10590" pour utiliser le chiffrement PSK uniquement pour les connexions de l'hôte vers le serveur Zabbix, et modifier l'identité PSK ainsi que la clé PSK. Notez que l'hôte doit être préconfiguré pour utiliser PSK.

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "host.update",
    "params": {
        "hostid": "10590",
        "tls_connect": 1,
        "tls_accept": 2,
        "tls_psk_identity": "PSK 002",
        "tls_psk": "e560cb0d918d26d31b4f642181f5f570ad89a390931102e5391d08327ba434e9"
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "10590"
        ]
    },
    "id": 1
}
```

Voir aussi

- host.massadd
- host.massremove
- Groupe d'hôtes
- Modèle
- Macro utilisateur
- Interface d'hôte
- Inventaire de l'hôte
- Tag d'hôte
- Proxy
- Groupe de proxies

Source

CHost::update() dans *ui/include/classes/api/services/CHost.php*.

Hôte découvert

Cette classe est conçue pour fonctionner avec les hôtes découverts.

Références d'objet :

- Hôte découvert

Méthodes disponibles :

- dhost.get - récupérer les hôtes découverts

Objet hôte découvert

Les objets suivants sont directement liés à l'API `dhost`.

Hôte découvert

> **Note:**
> Les hôtes découverts sont créés par le serveur Zabbix et ne peuvent pas être modifiés via l'API.

L'objet hôte découvert contient des informations sur un hôte découvert par une règle de découverte réseau. Il possède les propriétés suivantes.

Propriété	Type	Description
dhostid	ID	ID de l'hôte découvert.
druleid	ID	ID de la règle de découverte qui a détecté l'hôte.
lastdown	timestamp	Heure à laquelle l'hôte découvert est tombé en panne pour la dernière fois.
lastup	timestamp	Heure à laquelle l'hôte découvert a été de nouveau disponible pour la dernière fois.

Propriété	Type	Description
status	integer	Indique si l'hôte découvert est disponible ou indisponible. Un hôte est disponible s'il possède au moins un service découvert actif. Valeurs possibles : 0 - hôte disponible ; 1 - hôte indisponible.

dhost.get

Description

`integer/array dhost.get(object parameters)`

La méthode permet de récupérer les hôtes découverts selon les paramètres fournis.

> **Note:**
> Cette méthode est disponible pour les utilisateurs de tout type. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres des rôles utilisateur. Consultez Rôles utilisateur pour plus d'informations.

Paramètres

`(object)` Paramètres définissant la sortie souhaitée.

La méthode prend en charge les paramètres suivants.

Paramètre	Type	Description
dhostids	ID/array	Retourner uniquement les hôtes découverts avec les ID donnés.
druleids	ID/array	Retourner uniquement les hôtes découverts qui ont été créés par les règles de découverte données.
dserviceids	ID/array	Retourner uniquement les hôtes découverts qui exécutent les services donnés.
selectDRules	query	Retourner une propriété `drules` avec un tableau des règles de découverte qui ont détecté l'hôte.
selectDServices	query	Retourner une propriété `dservices` avec les services découverts exécutés sur l'hôte. Prend en charge `count`.
limitSelects	integer	Limite le nombre d'enregistrements renvoyés par les sous-sélections. S'applique aux sous-sélections suivantes : `selectDServices` - les résultats seront triés par `dserviceid`.
sortfield	string/array	Trier le résultat selon les propriétés données. Valeurs possibles : `dhostid`, `druleid`.
countOutput	boolean	Ces paramètres sont décrits dans le commentaire de référence.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Valeurs retournées

`(integer/array)` Retourne soit :

- un tableau d'objets ;

- le nombre d'objets récupérés, si le paramètre `countOutput` a été utilisé.

Exemple

Récupérer les hôtes découverts par règle de découverte

Récupérez tous les hôtes et les services découverts qu'ils exécutent, qui ont été détectés par la règle de découverte « 4 ».

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "dhost.get",
    "params": {
        "output": "extend",
        "selectDServices": "extend",
        "druleids": "4"
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "dservices": [
                {
                    "dserviceid": "1",
                    "dhostid": "1",
                    "type": "4",
                    "key_": "",
                    "value": "",
                    "port": "80",
                    "status": "0",
                    "lastup": "1337697227",
                    "lastdown": "0",
                    "dcheckid": "5",
                    "ip": "192.168.1.1",
                    "dns": "station.company.lan"
                }
            ],
            "dhostid": "1",
            "druleid": "4",
            "status": "0",
            "lastup": "1337697227",
            "lastdown": "0"
        },
        {
            "dservices": [
                {
                    "dserviceid": "2",
                    "dhostid": "2",
                    "type": "4",
                    "key_": "",
                    "value": "",
                    "port": "80",
                    "status": "0",
                    "lastup": "1337697234",
                    "lastdown": "0",
                    "dcheckid": "5",
                    "ip": "192.168.1.4",
                    "dns": "john.company.lan"
                }
            ],
```

```
            "dhostid": "2",
            "druleid": "4",
            "status": "0",
            "lastup": "1337697234",
            "lastdown": "0"
        },
        {
            "dservices": [
                {
                    "dserviceid": "3",
                    "dhostid": "3",
                    "type": "4",
                    "key_": "",
                    "value": "",
                    "port": "80",
                    "status": "0",
                    "lastup": "1337697234",
                    "lastdown": "0",
                    "dcheckid": "5",
                    "ip": "192.168.1.26",
                    "dns": "printer.company.lan"
                }
            ],
            "dhostid": "3",
            "druleid": "4",
            "status": "0",
            "lastup": "1337697234",
            "lastdown": "0"
        },
        {
            "dservices": [
                {
                    "dserviceid": "4",
                    "dhostid": "4",
                    "type": "4",
                    "key_": "",
                    "value": "",
                    "port": "80",
                    "status": "0",
                    "lastup": "1337697234",
                    "lastdown": "0",
                    "dcheckid": "5",
                    "ip": "192.168.1.7",
                    "dns": "mail.company.lan"
                }
            ],
            "dhostid": "4",
            "druleid": "4",
            "status": "0",
            "lastup": "1337697234",
            "lastdown": "0"
        }
    ],
    "id": 1
}
```


Voir aussi

- Service découvert
- Règle de découverte

Source

CDHost::get() dans *ui/include/classes/api/services/CDHost.php*.

Image

Cette classe est conçue pour fonctionner avec des images.

Références d'objet :

- Image

Méthodes disponibles :

- image.create - créer de nouvelles images
- image.delete - supprimer des images
- image.get - récupérer des images
- image.update - mettre à jour des images

Objet image

Les objets suivants sont directement liés à l'API `image`.

Image

L'objet image possède les propriétés suivantes.

Propriété	Type	Description
imageid	ID	ID de l'image. Comportement de la propriété : - *lecture seule* - *obligatoire* pour les opérations de mise à jour
name	string	Nom de l'image. Comportement de la propriété : - *obligatoire* pour les opérations de création
imagetype	integer	Type d'image. Valeurs possibles : 1 - *(par défaut)* icône ; 2 - image d'arrière-plan. Comportement de la propriété : - *constant* - *obligatoire* pour les opérations de création
image	string	Image encodée en Base64. La taille maximale de l'image encodée est de 1 Mo. La taille maximale peut être ajustée en modifiant la valeur de la constante `ZBX_MAX_IMAGE_SIZE`. Formats d'image pris en charge : PNG, JPEG, GIF et WebP. Comportement de la propriété : - *obligatoire* pour les opérations de création

image.create

Description

`object image.create(object/array images)`

Cette méthode vous permet de créer de nouvelles images.

Note:
Cette méthode est disponible uniquement pour le type d'utilisateur *Super admin*. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres du rôle utilisateur. Consultez User roles pour plus d'informations.

Paramètres

(object/array) Images à créer.

La méthode accepte des images avec les propriétés d'image standard.

Valeurs retournées

(object) Retourne un objet contenant les identifiants des images créées sous la propriété imageids. L'ordre des ID retournés correspond à l'ordre des images transmises.

Exemples

Créer une image d'icône

Créez une icône de nuage.

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "image.create",
    "params": {
        "imagetype": 1,
        "name": "Cloud_(24)",
        "image": "iVBORw0KGgoAAAANSUhEUgAAABgAAAANCAYAAACzbK7QAAAABHNCSVQICAgIfAhkiAAAAAlwSFlzAAACmAAAApgB
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "imageids": [
            "188"
        ]
    },
    "id": 1
}
```

Créer une image d'arrière-plan

Créez une petite image d'arrière-plan à utiliser comme motif.

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "image.create",
    "params": {
        "imagetype": 2,
        "name": "background_image",
        "image": "iVBORw0KGgoAAAANSUhEUgAAABgAAAAYCAIAAABvFaqvAAABhWlDQ1BJQ0MgcHJvZmlsZQAAKJF9kb1Lw0AYxp+m
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "imageids": [
            "189"
        ]
    },
    "id": 1
}
```

Source

CImage::create() dans *ui/include/classes/api/services/CImage.php*.

image.delete

Description

`object image.delete(array imageIds)`

Cette méthode vous permet de supprimer des images.

> **Note:**
> Cette méthode est disponible uniquement pour le type d'utilisateur *Super admin*. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres du rôle utilisateur. Consultez User roles pour plus d'informations.

Paramètres

`(array)` ID des images à supprimer.

Valeurs de retour

`(object)` Renvoie un objet contenant les ID des images supprimées dans la propriété `imageids`.

Exemples

Supprimer plusieurs images

Supprimez deux images.

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "image.delete",
    "params": [
        "188",
        "192"
    ],
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "imageids": [
            "188",
            "192"
        ]
    },
    "id": 1
}
```

Source

CImage::delete() dans *ui/include/classes/api/services/CImage.php*.

image.get

Description

`integer/array image.get(object parameters)`

Cette méthode permet de récupérer des images selon les paramètres fournis.

> **Note:**
> Cette méthode est disponible pour les utilisateurs de tout type. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres des rôles utilisateur. Consultez Rôles utilisateur pour plus d'informations.

Paramètres

`(object)` Paramètres définissant la sortie souhaitée.

La méthode prend en charge les paramètres suivants.

Paramètre	Type	Description
imageids	ID/array	Retourner uniquement les images avec les ID donnés.
sysmapids	ID/array	Retourner les images utilisées sur les cartes données.
select_image	flag	Retourner une propriété `image` contenant l'image encodée en Base64.
sortfield	string/array	Trier le résultat selon les propriétés données. Valeurs possibles : `imageid`, `name`.
countOutput	boolean	Ces paramètres sont décrits dans le commentaire de référence.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Valeurs retournées

`(integer/array)` Retourne soit :

- un tableau d'objets ;
- le nombre d'objets récupérés, si le paramètre `countOutput` a été utilisé.

Exemples

Récupérer une image

Récupérez toutes les données de l'image avec l'ID « 2 ».

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "image.get",
    "params": {
        "output": "extend",
        "select_image": true,
        "imageids": "2"
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "imageid": "2",
            "imagetype": "1",
            "name": "Cloud_(24)",
            "image": "iVBORw0KGgoAAAANSUhEUgAAABgAAAANCAYAAACzbK7QAAAABHNCSVQICAgIfAhkiAAAAAlwSFlzAAACmAAA
        }
    ],
    "id": 1
}
```

Source

CImage::get() dans *ui/include/classes/api/services/CImage.php*.

image.update

Description

`object image.update(object/array images)`

Cette méthode vous permet de mettre à jour des images existantes.

> **Note:**
> Cette méthode est disponible uniquement pour le type d'utilisateur *Super admin*. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres du rôle utilisateur. Consultez User roles pour plus d'informations.

Paramètres

`(object/array)` Propriétés de l'image à mettre à jour.

La propriété `imageid` doit être définie pour chaque image, toutes les autres propriétés sont facultatives. Seules les propriétés transmises seront mises à jour, toutes les autres resteront inchangées.

La méthode accepte des images avec les propriétés d'image standard.

Valeurs retournées

`(object)` Retourne un objet contenant les identifiants des images mises à jours sous la propriété `imageids`.

Exemples

Renommer une image

Renommez l'image en « Cloud icon ».

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "image.update",
    "params": {
        "imageid": "2",
        "name": "Cloud icon"
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "imageids": [
            "2"
        ]
    },
    "id": 1
}
```

Source

CImage::update() dans *ui/include/classes/api/services/CImage.php*.

Informations sur le serveur

Cette classe est conçue pour récupérer des informations sur le serveur Zabbix.

Méthodes disponibles :

- serverinfo.get - récupérer l'heure de la dernière mise à jour du serveur Zabbix

serverinfo.get

Cette méthode vous permet de récupérer l'heure de la dernière mise à jour du serveur Zabbix indiqué.

La méthode est disponible pour les utilisateurs de type *User*, *Admin*, *Super admin* et les utilisateurs non authentifiés.

Signature de la méthode : `object serverinfo.get(object parameters)`

Source de la méthode : `CServerInfo::get()` dans `ui/include/classes/api/services/CServerInfo.php`

Paramètres

Parameter	Type	Description
serverid	ID	ID du serveur Zabbix. Comportement du paramètre : - Obligatoire
output	query	Ce paramètre est décrit dans le commentaire de référence.

Valeurs de retour

Property	Type	Description
serverid	ID	ID du serveur Zabbix.
lastaccess	timestamp	Heure de la dernière mise à jour provenant du serveur Zabbix. Pour un cluster à haute disponibilité (HA), cette valeur correspond à la valeur `lastaccess` du nœud actif.

Exemples

Récupérer les informations du serveur

Récupérer les informations sur le serveur Zabbix indiqué.

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "serverinfo.get",
    "params": {
        "serverid": "019fb1f9-22bd-74ea-bc03-51efb8b76aba",
        "output": "extend"
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "serverid": "019fb1f9-22bd-74ea-bc03-51efb8b76aba",
        "lastaccess": 1785337245
    },
    "id": 1
}
```

Interface du hôte

Cette classe permet d'accéder aux interfaces des hôtes.

Références d'objet :

- Interface du hôte
  - Détails

Méthode disponible :

- hostinterface.get - récupérer les interfaces des hôtes

Les interfaces des hôtes sont créées, mises à jour et supprimées à l'aide des méthodes des hôtes.

Objet interface d'hôte

Les objets suivants sont directement liés à l'API `hostinterface`.

Interface du hôte

L'objet d'interface du hôte possède les propriétés suivantes.

Property	Type	Description
interfaceid	ID	ID de l'interface. Comportement de la propriété : - *en lecture seule* - *obligatoire* pour les opérations de mise à jour
available	integer	Disponibilité de l'interface du hôte. Valeurs possibles : 0 - *(par défaut)* inconnue ; 1 - disponible ; 2 - indisponible. Comportement de la propriété : - *en lecture seule*
hostid	ID	ID du hôte auquel appartient l'interface. Comportement de la propriété : - *constante* - *obligatoire* pour les opérations de création
type	integer	Type d'interface. Valeurs possibles : 1 - Agent ; 2 - SNMP ; 3 - IPMI ; 4 - JMX. Comportement de la propriété : - *obligatoire* pour les opérations de création
ip	string	Adresse IP utilisée par l'interface. Peut être vide si la connexion est établie via DNS. Comportement de la propriété : - *obligatoire* pour les opérations de création
dns	string	Nom DNS utilisé par l'interface. Peut être vide si la connexion est établie via IP. Comportement de la propriété : - *obligatoire* pour les opérations de création
port	string	Numéro de port utilisé par l'interface. Peut contenir des macros utilisateur. Comportement de la propriété : - *obligatoire* pour les opérations de création

Property	Type	Description
useip	integer	Indique si la connexion doit être établie via IP. Valeurs possibles : 0 - se connecter en utilisant le nom DNS du hôte ; 1 - se connecter en utilisant l'adresse IP du hôte. Comportement de la propriété : - *obligatoire* pour les opérations de création
main	integer	Indique si l'interface est utilisée par défaut sur le hôte. Une seule interface d'un même type peut être définie comme interface par défaut sur un hôte. Valeurs possibles : 0 - pas par défaut ; 1 - par défaut. Comportement de la propriété : - *obligatoire* pour les opérations de création
details	object	Objet details supplémentaire pour l'interface. Comportement de la propriété : - *obligatoire* si `type` est défini sur « SNMP »
disable_until	timestamp	Prochaine heure d'interrogation d'une interface de hôte indisponible. Comportement de la propriété : - *en lecture seule*
error	string	Texte de l'erreur si l'interface du hôte est indisponible. Comportement de la propriété : - *en lecture seule*
errors_from	timestamp	Heure à laquelle l'interface du hôte est devenue indisponible. Comportement de la propriété : - *en lecture seule*

Détails

L'objet details possède les propriétés suivantes.

Propriété	Type	Description
version	integer	Version de l'interface SNMP. Valeurs possibles : 1 - SNMPv1 ; 2 - SNMPv2c ; 3 - SNMPv3. Comportement de la propriété : - *obligatoire*
bulk	integer	Indique s'il faut utiliser des requêtes SNMP bulk. Valeurs possibles : 0 - ne pas utiliser de requêtes bulk ; 1 - (par défaut) - utiliser des requêtes bulk.
community	string	Communauté SNMP. Utilisée uniquement par les interfaces SNMPv1 et SNMPv2. Comportement de la propriété : - *obligatoire* si `version` est définie sur "SNMPv1" ou "SNMPv2c"

Propriété	Type	Description
max_repetitions	integer	Valeur de répétition maximale pour les requêtes SNMP bulk natives (GetBulkRequest-PDUs). Utilisée uniquement pour les éléments `discovery[]` et `walk[]` en SNMPv2 et v3. Par défaut : 10.
securityname	string	Nom de sécurité SNMPv3. Utilisé uniquement par les interfaces SNMPv3.
securitylevel	integer	Niveau de sécurité SNMPv3. Utilisé uniquement par les interfaces SNMPv3. Valeurs possibles : 0 - (par défaut) - noAuthNoPriv ; 1 - authNoPriv ; 2 - authPriv.
authpassphrase	string	Phrase secrète d'authentification SNMPv3. Utilisée uniquement par les interfaces SNMPv3.
privpassphrase	string	Phrase secrète de confidentialité SNMPv3. Utilisée uniquement par les interfaces SNMPv3.
authprotocol	integer	Protocole d'authentification SNMPv3. Utilisé uniquement par les interfaces SNMPv3. Valeurs possibles : 0 - (par défaut) - MD5 ; 1 - SHA1 ; 2 - SHA224 ; 3 - SHA256 ; 4 - SHA384 ; 5 - SHA512.
privprotocol	integer	Protocole de confidentialité SNMPv3. Utilisé uniquement par les interfaces SNMPv3. Valeurs possibles : 0 - (par défaut) - DES ; 1 - AES128 ; 2 - AES192 ; 3 - AES256 ; 4 - AES192C ; 5 - AES256C.
contextname	string	Nom de contexte SNMPv3. Utilisé uniquement par les interfaces SNMPv3.

hostinterface.get

Description

`integer/array hostinterface.get(object parameters)`

Cette méthode vous permet de récupérer les interfaces des hôtes selon les paramètres indiqués.

> **Note:**
> Cette méthode est disponible pour les utilisateurs de tout type. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres des rôles utilisateur. Consultez la page Rôles utilisateur pour plus d'informations.

Paramètres

`(object)` Paramètres définissant la sortie souhaitée.

La méthode prend en charge les paramètres suivants.

Paramètre	Type	Description
hostids	ID/array	Retourner uniquement les interfaces d'hôte utilisées par les hôtes donnés.
interfaceids	ID/array	Retourner uniquement les interfaces d'hôte avec les ID donnés.
itemids	ID/array	Retourner uniquement les interfaces d'hôte utilisées par les éléments donnés.
triggerids	ID/array	Retourner uniquement les interfaces d'hôte utilisées par les éléments dans les déclencheurs donnés.
selectItems	query	Retourner une propriété `items` avec les éléments qui utilisent l'interface. Prend en charge `count`.
selectHosts	query	Retourner une propriété `hosts` avec un tableau d'hôtes qui utilisent l'interface.
limitSelects	integer	Limite le nombre d'enregistrements renvoyés par les sous-sélections. S'applique aux sous-sélections suivantes : `selectItems`.
sortfield	string/array	Trier le résultat selon les propriétés données. Valeurs possibles : `interfaceid`, `dns`, `ip`.
countOutput	boolean	Ces paramètres sont décrits dans le commentaire de référence.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Valeurs retournées

`(integer/array)` Retourne soit :

- un tableau d'objets ;
- le nombre d'objets récupérés, si le paramètre `countOutput` a été utilisé.

Exemples

Récupérer les interfaces de l'hôte

Récupérez toutes les données concernant les interfaces utilisées par l'hôte « 30057 ».

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "hostinterface.get",
    "params": {
        "output": "extend",
        "hostids": "30057"
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "interfaceid": "50039",
```

```
            "hostid": "30057",
            "main": "1",
            "type": "1",
            "useip": "0",
            "ip": "",
            "dns": "localhost",
            "port": "10050",
            "available": "0",
            "error": "",
            "errors_from": "0",
            "disable_until": "0",
            "details": []
        },
        {
            "interfaceid": "55082",
            "hostid": "30057",
            "main": "1",
            "type": "2",
            "useip": "1",
            "ip": "127.0.0.1",
            "dns": "",
            "port": "161",
            "available": "0",
            "error": "",
            "errors_from": "0",
            "disable_until": "0",
            "details": {
                "version": "2",
                "bulk": "0",
                "community": "{$SNMP_COMMUNITY}",
                "max_repetitions": "10"
            }
        }
    ],
    "id": 1
}
```


Voir également

- Hôte
- Elément

Source

CHostInterface::get() dans *ui/include/classes/api/services/CHostInterface.php*.

Journal APM

Cette classe est conçue pour fonctionner avec les journaux collectés pour la surveillance des performances des applications (APM).

Références d'objet :

- Journal APM

Méthodes disponibles :

- apmlog.get - récupérer les enregistrements du journal

Objet de journal APM

Les objets suivants sont directement liés à l'API apmlog.

Note:
Chaque propriété correspond à la colonne portant le même nom dans la table `otel_logs` de la source de données de télémétrie (par exemple, la propriété `severity_text` correspond à la colonne `SeverityText`). Pour obtenir la liste des colonnes, consultez Colonnes de données de télémétrie prises en charge.

Journal APM

L'objet de journal APM représente un enregistrement de journal (un enregistrement textuel d'un événement, tel qu'un paiement échoué ou un délai d'attente de base de données). Un enregistrement de journal contient des informations sur un événement survenu à un moment précis, notamment sa gravité, son message et ses attributs. Si un enregistrement de journal a été produit lors d'une opération faisant partie d'une trace, il contient également les ID de cette trace et de ce span.

L'objet de journal APM possède les propriétés suivantes :

Property	Type	Description
timestamp	integer	Heure Unix (en nanosecondes) à laquelle l'événement s'est produit. Comportement de la propriété : - *lecture seule*
traceid	string	ID, le cas échéant, de la trace pendant laquelle l'enregistrement de journal a été produit. Comportement de la propriété : - *lecture seule*
spanid	string	ID, le cas échéant, du span pendant lequel l'enregistrement de journal a été produit. Comportement de la propriété : - *lecture seule*
trace_flags	integer	Indicateurs de trace du contexte de trace de l'enregistrement de journal, sous forme de masque de bits. Si le bit de poids faible est défini (valeur 1), la trace est échantillonnée. Comportement de la propriété : - *lecture seule*
severity_text	string	Niveau de gravité de l'enregistrement de journal, tel qu'il est fourni par l'application (par exemple, `INFO`, `WARN` ou `ERROR`). Comportement de la propriété : - *lecture seule*
severity_number	integer	Niveau de gravité de l'enregistrement de journal, sous forme de nombre. Valeurs possibles : 0 - UNSPECIFIED ; 1-4 - TRACE ; 5-8 - DEBUG ; 9-12 - INFO ; 13-16 - WARN ; 17-20 - ERROR ; 21-24 - FATAL. Comportement de la propriété : - *lecture seule*
service_name	string	Nom du service ayant produit l'enregistrement de journal (par exemple, `zabbix-frontend`), extrait de l'attribut de ressource OpenTelemetry `service.name`. Comportement de la propriété : - *lecture seule*

Property	Type	Description
body	string	Message de l'enregistrement de journal. Comportement de la propriété : - *lecture seule*
resource_schema_url	string	URL, le cas échéant, du schéma OpenTelemetry des attributs de ressource. Comportement de la propriété : - *lecture seule*
resource_attributes	object	Attributs de ressource clé-valeur de l'enregistrement de journal (par exemple, `host.name: eabf8c2a5ddc`). Comportement de la propriété : - *lecture seule*
scope_schema_url	string	URL, le cas échéant, du schéma OpenTelemetry du périmètre d'instrumentation et de ses attributs. Comportement de la propriété : - *lecture seule*
scope_name	string	Nom du périmètre d'instrumentation ayant produit l'enregistrement de journal. Comportement de la propriété : - *lecture seule*
scope_version	string	Version, le cas échéant, du périmètre d'instrumentation. Comportement de la propriété : - *lecture seule*
scope_attributes	object	Attributs clé-valeur du périmètre d'instrumentation. Comportement de la propriété : - *lecture seule*
log_attributes	object	Attributs de journal clé-valeur de l'enregistrement de journal (par exemple, `log.file.path: /var/log/zabbix/zabbix_frontend.log`). Comportement de la propriété : - *lecture seule*
event_name	string	Nom, le cas échéant, du type d'événement que représente l'enregistrement de journal. Comportement de la propriété : - *lecture seule*

apmlog.get

Description

`integer/array apmlog.get(object parameters)`

Cette méthode vous permet de récupérer des enregistrements de journal selon les paramètres indiqués.

> **Note:**
> Cette méthode est disponible pour les utilisateurs de tout type. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres des rôles utilisateur. Consultez la page Rôles utilisateur pour plus d'informations.

Paramètres

`(object)` Paramètres définissant le résultat souhaité.

La méthode prend en charge les paramètres suivants.

Paramètre	Type	Description
time_from	timestamp	Retourne uniquement les enregistrements de journal des événements survenus à l'heure indiquée ou après. Comportement du paramètre : - *requis*
time_till	timestamp	Retourne uniquement les enregistrements de journal des événements survenus avant l'heure indiquée. Comportement du paramètre : - *requis*
traceids	string/array	Retourne uniquement les enregistrements de journal produits pendant les traces indiquées.
spanids	string/array	Retourne uniquement les enregistrements de journal produits pendant les spans indiqués.
with_flags_on	integer	Retourne uniquement les enregistrements de journal pour lesquels tous les indicateurs de trace indiqués sont définis. Pour spécifier plusieurs indicateurs, utilisez la somme de leurs valeurs.
with_flags_off	integer	Retourne uniquement les enregistrements de journal pour lesquels aucun des indicateurs de trace indiqués n'est défini. Pour spécifier plusieurs indicateurs, utilisez la somme de leurs valeurs.
resource_attributes	object/array	Retourne uniquement les enregistrements de journal possédant les attributs de ressource indiqués. Format : `[{"key": "<key>", "value": "<value>", "operator": "<operator>"}, ...]`. Un tableau vide retourne tous les enregistrements de journal. Valeurs possibles de l'opérateur : 0 - *(par défaut)* Contient ; 1 - Est égal à ; 2 - Ne contient pas ; 3 - N'est pas égal à ; 4 - Existe ; 5 - N'existe pas.
resource_attributes_evaltype	integer	Méthode d'évaluation des conditions dans `resource_attributes`. Valeurs possibles : 0 - *(par défaut)* Et/Ou ; 2 - Ou.
scope_attributes	object/array	Retourne uniquement les enregistrements de journal possédant les attributs de portée indiqués. Format : `[{"key": "<key>", "value": "<value>", "operator": "<operator>"}, ...]`. Un tableau vide retourne tous les enregistrements de journal. Valeurs possibles de l'opérateur : 0 - *(par défaut)* Contient ; 1 - Est égal à ; 2 - Ne contient pas ; 3 - N'est pas égal à ; 4 - Existe ; 5 - N'existe pas.
scope_attributes_evaltype	integer	Méthode d'évaluation des conditions dans `scope_attributes`. Valeurs possibles : 0 - *(par défaut)* Et/Ou ; 2 - Ou.

Paramètre	Type	Description
log_attributes	object/array	Retourne uniquement les enregistrements de journal possédant les attributs de journal indiqués. Format : `[{"key": "<key>", "value": "<value>", "operator": "<operator>"}, ...]`. Un tableau vide retourne tous les enregistrements de journal. Valeurs possibles de l'opérateur : 0 - *(par défaut)* Contient ; 1 - Est égal à ; 2 - Ne contient pas ; 3 - N'est pas égal à ; 4 - Existe ; 5 - N'existe pas.
log_attributes_evaltype	integer	Méthode d'évaluation des conditions dans `log_attributes`. Valeurs possibles : 0 - *(par défaut)* Et/Ou ; 2 - Ou.
filter	object	Retourne uniquement les résultats correspondant exactement au filtre indiqué. Accepte un objet dont les clés sont des noms de propriétés et dont les valeurs sont soit une valeur unique, soit un tableau de valeurs à comparer. Propriétés prises en charge : `traceid`, `spanid`, `trace_flags`, `severity_text`, `severity_number`, `service_name`, `resource_schema_url`, `scope_schema_url`, `scope_name`, `scope_version`, `event_name`.
search	object	Retourne les résultats correspondant au modèle indiqué (sans distinction entre majuscules et minuscules). Accepte un objet dont les clés sont des noms de propriétés et dont les valeurs sont les chaînes à rechercher. Si aucune option supplémentaire n'est indiquée, une recherche `LIKE "%…%"` est effectuée. Propriétés prises en charge : `severity_text`, `service_name`, `body`, `resource_schema_url`, `scope_schema_url`, `scope_name`, `scope_version`, `event_name`.
sortfield	string/array	Trie le résultat selon les propriétés indiquées. Valeurs possibles : `timestamp`, `traceid`, `spanid`, `trace_flags`, `severity_text`, `severity_number`, `service_name`, `resource_schema_url`, `scope_schema_url`, `scope_name`, `scope_version`, `event_name`.
offset	integer	Nombre d'enregistrements à ignorer avant de retourner le résultat.
countOutput	boolean	Ces paramètres sont décrits dans la documentation de référence.
excludeSearch	boolean	
limit	integer	
output	query	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Valeurs de retour

`(integer/array)` Renvoie soit :

- un tableau d'objets ;
- le nombre d'objets récupérés, si le paramètre `countOutput` a été utilisé.

Exemples

Récupération des journaux d'erreurs d'une trace

Récupérer les enregistrements de journal avec la gravité ERROR (numéros de gravité 17 à 20) associés à la trace "8a4f1c7e2b9d4e6fa0c3b5d7e

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "apmlog.get",
    "params": {
        "output": "extend",
        "time_from": 1790848800,
        "time_till": 1790852400,
        "traceids": "8a4f1c7e2b9d4e6fa0c3b5d7e9f1a2c4",
        "filter": {
            "severity_number": [17, 18, 19, 20]
        },
        "sortfield": "timestamp",
        "sortorder": "ASC"
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "timestamp": "1790849305091452376",
            "traceid": "8a4f1c7e2b9d4e6fa0c3b5d7e9f1a2c4",
            "spanid": "5d2e8f1a9c3b7046",
            "trace_flags": "1",
            "severity_text": "ERROR",
            "severity_number": "17",
            "service_name": "zabbix-frontend",
            "body": "mysqli_query(): (HY000/1205): Lock wait timeout exceeded; try restarting transaction"
            "resource_schema_url": "https://opentelemetry.io/schemas/1.26.0",
            "resource_attributes": {
                "service.name": "zabbix-frontend",
                "host.name": "6583ceeb5f94",
                "host.arch": "x86_64",
                "os.type": "linux"
            },
            "scope_schema_url": "",
            "scope_name": "zabbix-frontend",
            "scope_version": "",
            "scope_attributes": {},
            "log_attributes": {
                "log.file.path": "/var/log/zabbix/zabbix_frontend.log",
                "code.filepath": "/usr/share/zabbix/include/db.inc.php"
            },
            "event_name": ""
        }
    ],
    "id": 1
}
```

Recherche de messages dans les journaux

Récupérer l'heure, le service et le message des 10 enregistrements de journal les plus récents dont le message contient « timeout ».

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "apmlog.get",
    "params": {
        "output": ["timestamp", "service_name", "severity_text", "body"],
        "time_from": 1790848800,
        "time_till": 1790852400,
        "search": {
            "body": "timeout"
        },
        "sortfield": "timestamp",
        "sortorder": "DESC",
        "limit": 10
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "timestamp": "1790849305091452376",
            "service_name": "zabbix-frontend",
            "severity_text": "ERROR",
            "body": "mysqli_query(): (HY000/1205): Lock wait timeout exceeded; try restarting transaction"
        },
        {
            "timestamp": "1790848961730284115",
            "service_name": "zabbix-frontend",
            "severity_text": "ERROR",
            "body": "mysqli_query(): (HY000/1205): Lock wait timeout exceeded; try restarting transaction"
        }
    ],
    "id": 1
}
```

Voir aussi

- Trace APM
- Span APM
- Métrique APM

Source

CApmLog::get() dans *ui/include/classes/api/services/CApmLog.php*.

Journal d'audit

Cette classe est conçue pour fonctionner avec le journal d'audit.

Références d'objet :

- Journal d'audit

Méthodes disponibles :

- auditlog.get - récupérer les enregistrements du journal d'audit

Objet du journal d'audit

Les objets suivants sont directement liés à l'API `auditlog`.

Journal d'audit

L'objet journal d'audit contient des informations sur les actions des utilisateurs. Il possède les propriétés suivantes.

Propriété	Type	Description
auditid	ID	ID de l'entrée du journal d'audit. Généré à l'aide de l'algorithme CUID.
userid	ID	userid de l'auteur de l'entrée du journal d'audit.
username	string	nom d'utilisateur de l'auteur de l'entrée du journal d'audit.
clock	timestamp	horodatage de création de l'entrée du journal d'audit.
ip	string	adresse IP de l'auteur de l'entrée du journal d'audit.
action	integer	action de l'entrée du journal d'audit. Valeurs possibles : 0 - Ajouter ; 1 - Mettre à jour ; 2 - Supprimer ; 4 - Déconnexion ; 7 - Exécuter ; 8 - Connexion ; 9 - Échec de connexion ; 10 - Effacement de l'historique ; 11 - Actualisation de la configuration ; 12 - Push.

Propriété	Type	Description
resourcetype	integer	type de ressource de l'entrée du journal d'audit. Valeurs possibles : 0 - Utilisateur ; 3 - Type de média ; 4 - Hôte ; 5 - Action ; 6 - Graphique ; 11 - Groupe d'utilisateurs ; 13 - Déclencheur ; 14 - Groupe d'hôtes ; 15 - Élément ; 16 - Image ; 17 - Correspondance de valeurs ; 18 - Service ; 19 - Carte ; 22 - Scénario web ; 23 - Règle de découverte ; 25 - Script ; 26 - Proxy ; 27 - Maintenance ; 28 - Expression régulière ; 29 - Macro ; 30 - Modèle ; 31 - Prototype de déclencheur ; 32 - Correspondance d'icônes ; 33 - Tableau de bord ; 34 - Corrélation d'événements ; 35 - Prototype de graphique ; 36 - Prototype d'élément ; 37 - Prototype d'hôte ; 38 - Enregistrement automatique ; 39 - Module ; 40 - Paramètres ; 41 - Housekeeping ; 42 - Authentification ; 43 - Tableau de bord de modèle ; 44 - Rôle utilisateur ; 45 - Jeton d'API ; 46 - Rapport planifié ; 47 - Nœud de haute disponibilité ; 48 - SLA ; 49 - Répertoire d'utilisateurs ; 50 - Groupe de modèles ; 51 - Connecteur ; 52 - Règle LLD ; 53 - Historique ; 54 - Authentification multifacteur ; 55 - Groupe de proxy ; 56 - Prototype de règle LLD.
resourceid	ID	identifiant de ressource de l'entrée du journal d'audit.
resource_cuid	ID	identifiant unique de ressource de l'entrée du journal d'audit généré à l'aide de l'algorithme CUID.
resourcename	string	nom lisible par l'humain de la ressource de l'entrée du journal d'audit.
recordsetid	ID	ID de l'ensemble d'enregistrements de l'entrée du journal d'audit. Les enregistrements du journal d'audit créés lors de la même opération auront le même ID d'ensemble d'enregistrements. Généré à l'aide de l'algorithme CUID.

Propriété	Type	Description
details	text	détails de l'entrée du journal d'audit. Les détails sont stockés sous forme d'objet JSON, où chaque nom de propriété est un chemin vers la propriété ou l'objet imbriqué dans lequel la modification s'est produite, et où chaque valeur contient les données (au format tableau) concernant la modification de cette propriété ou de cet objet imbriqué. Formats de valeur possibles : ["add"] - L'objet imbriqué a été ajouté ; ["add", "<value>"] - La propriété de l'objet ajouté est égale à <value> ; ["update"] - L'objet imbriqué a été mis à jour ; ["update", "<new value>", "<old value>"] - La propriété de l'objet mis à jour a été modifiée de <old value> à <new value> ; ["delete"] - L'objet imbriqué a été supprimé.

auditlog.get

Description

`integer/array auditlog.get(object parameters)`

La méthode permet de récupérer les enregistrements du journal d'audit selon les paramètres fournis.

> **Note:**
> Cette méthode est disponible uniquement pour le type d'utilisateur *Super admin*. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres du rôle utilisateur. Consultez Rôles utilisateur pour plus d'informations.

Paramètres

`(object)` Paramètres définissant la sortie souhaitée.

La méthode prend en charge les paramètres suivants.

Paramètre	Type	Description
auditids	ID/array	Retourne uniquement le journal d'audit avec les ID indiqués.
userids	ID/array	Retourne uniquement le journal d'audit créé par les utilisateurs indiqués.
time_from	timestamp	Retourne uniquement les entrées du journal d'audit qui ont été créées après ou à l'heure indiquée.
time_till	timestamp	Retourne uniquement les entrées du journal d'audit qui ont été créées avant ou à l'heure indiquée.
sortfield	string/array	Trie le résultat selon les propriétés indiquées. Valeurs possibles : `auditid`, `userid`, `clock`.
countOutput	boolean	Ces paramètres sont décrits dans le commentaire de référence.
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Valeurs de retour

`(integer/array)` Renvoie soit :

- un tableau d'objets ;

- le nombre d'objets récupérés, si le paramètre `countOutput` a été utilisé.

Exemples

Récupérer le journal d'audit

Récupérez les deux enregistrements les plus récents du journal d'audit.

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "auditlog.get",
    "params": {
        "output": "extend",
        "sortfield": "clock",
        "sortorder": "DESC",
        "limit": 2
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "auditid": "cksstgfam0001yhdcc41y20q2",
            "userid": "1",
            "username": "Admin",
            "clock": "1629975715",
            "ip": "127.0.0.1",
            "action": "1",
            "resourcetype": "0",
            "resourceid": "0",
            "resource_cuid": "0",
            "resourcename": "Jim",
            "recordsetid": "cksstgfal0000yhdcso67ondl",
            "details": "{\"user.name\":[\"update\",\"Jim\",\"\"],\"user.medias[37]\":[\"add\"],\"user.medi
        },
        {
            "auditid": "ckssofl0p0001yhdcqxclsg8r",
            "userid": "1",
            "username": "Admin",
            "clock": "1629967278",
            "ip": "127.0.0.1",
            "action": "0",
            "resourcetype": "0",
            "resourceid": "20",
            "resource_cuid": "0",
            "resourcename": "John",
            "recordsetid": "ckssofl0p0000yhdcpxyo1jgo",
            "details": "{\"user.username\":[\"add\",\"John\"], \"user.userid:\":[\"add\",\"20\"],\"user.us
        }
    ],
    "id": 1
}
```

Voir également

- Objet Audit log

Source

CAuditLog::get() dans *ui/include/classes/api/services/CAuditLog.php*.

Macro utilisateur

Cette classe est conçue pour fonctionner avec les macros utilisateur au niveau de l'hôte et les macros utilisateur globales.

Références d'objet :

- Macro globale
- Macro d'hôte
  - Configuration de macro

Méthodes disponibles :

- usermacro.create - créer de nouvelles macros d'hôte
- usermacro.createglobal - créer de nouvelles macros globales
- usermacro.delete - supprimer des macros d'hôte
- usermacro.deleteglobal - supprimer des macros globales
- usermacro.get - récupérer des macros d'hôte et globales
- usermacro.update - mettre à jour des macros d'hôte
- usermacro.updateglobal - mettre à jour des macros globales

Objet macro utilisateur

Les objets suivants sont directement liés à l'API `usermacro`.

Macro globale

L'objet macro globale possède les propriétés suivantes.

Propriété	Type	Description
globalmacroid	ID	ID de la macro globale. Comportement de la propriété : - *lecture seule* - *obligatoire* pour les opérations de mise à jour
description	string	Description de la macro.
macro	string	Chaîne de macro. Comportement de la propriété : - *obligatoire* pour les opérations de création
type	integer	Type de macro. Valeurs possibles : 0 - *(par défaut)* Macro texte ; 1 - Macro secrète ; 2 - Secret Vault.
value	string	Valeur de la macro. Comportement de la propriété : - *écriture seule* si `type` est défini sur "Macro secrète" - *obligatoire* pour les opérations de création

Macro d'hôte

L'objet macro d'hôte définit une macro disponible sur un hôte, un prototype d'hôte ou un modèle. Il possède les propriétés suivantes.

Propriété	Type	Description
hostmacroid	ID	ID de la macro. Comportement de la propriété : - *lecture seule* - *obligatoire* pour les opérations de mise à jour

Propriété	Type	Description
automatic	integer	Définit si la macro est contrôlée par une règle de découverte. Valeurs possibles : 0 - *(par défaut)* La macro est gérée par l'utilisateur ; 1 - La macro est gérée par la règle de découverte. L'utilisateur n'est pas autorisé à créer une macro automatique. Pour mettre à jour une macro automatique, elle doit être convertie en macro manuelle.
config	object/array	Configuration de la macro responsable de la façon dont la macro est affichée dans l'assistant d'hôte.
description	string	Description de la macro.
hostid	ID	ID de l'hôte, du prototype d'hôte ou du modèle auquel la macro appartient. Comportement de la propriété : - *constant* - *obligatoire* pour les opérations de création
macro	string	Chaîne de macro. Comportement de la propriété : - *obligatoire* pour les opérations de création
type	integer	Type de macro. Valeurs possibles : 0 - *(par défaut)* Macro texte ; 1 - Macro secrète ; 2 - Secret du coffre-fort.
value	string	Valeur de la macro. Comportement de la propriété : - *écriture seule* si `type` est défini sur "Macro secrète" - *obligatoire* pour les opérations de création

Configuration des macros

L'objet de configuration des macros définit la manière dont une macro est affichée dans l'assistant d'hôte.

Propriété	Type	Description
type	integer	Type du champ de saisie de macro. Valeurs possibles : 0 - La macro n'est pas utilisée dans l'assistant d'hôte ; 1 - Zone de texte ; 2 - Liste ; 3 - Case à cocher. Comportement de la propriété : - *obligatoire*
label	string	Libellé du champ de saisie de macro. Comportement de la propriété : - *obligatoire* si `type` est défini sur "Textbox", "List" ou "Checkbox"
description	text	Texte d'aide affiché à côté du champ de saisie de macro. Prend en charge le formatage Markdown.
priority	integer	Position du champ de saisie de macro dans la liste des macros. Si `priority` n'est pas défini, la macro est ajoutée à la fin de la liste des macros non groupées.

Propriété	Type	Description
required	integer	Indique que la macro est obligatoire. Valeurs possibles : 0 - Non obligatoire ; 1 - Obligatoire. Comportement de la propriété : - *pris en charge* si `type` est défini sur ”Textbox” ou ”List”
regex	string	Expression régulière permettant de valider la saisie utilisateur dans un champ de zone de texte. Comportement de la propriété : - *pris en charge* si `type` est défini sur ”Textbox”
section_name	string	Libellé de la section repliable dans laquelle la macro est groupée. Si `section_name` n’est pas défini, la macro n’est pas groupée. Les macros non groupées sont affichées en premier ; les macros groupées sont affichées en dessous, triées selon leur `priority` dans chaque section repliable.
options	text	Chaîne JSON définissant les éléments de liste ou les valeurs de case à cocher. Pour les listes : un tableau d’objets avec les propriétés `value` et `text`. Exemple : `[{"value": "http", "text": "HTTP"}, {"value": "https", "text": "HTTPS"}]` Pour les cases à cocher : un objet avec les propriétés `checked` et `unchecked`. Exemple : `{"checked": true, "unchecked": false}` Comportement de la propriété : - *obligatoire* si `type` est défini sur ”List” ou ”Checkbox”

usermacro.create

Description

`object usermacro.create(object/array hostMacros)`

Cette méthode permet de créer de nouvelles macros d’hôte.

> **Note:**
> Cette méthode est disponible uniquement pour les types d’utilisateur *Admin* et *Super admin*. Les autorisations d’appel de la méthode peuvent être révoquées dans les paramètres des rôles utilisateur. Consultez Rôles utilisateur pour plus d’informations.

Paramètres

`(object/array)` Macros d’hôte à créer.

La méthode accepte des macros d’hôte avec les propriétés standard des macros d’hôte.

Valeurs retournées

`(object)` Renvoie un objet contenant les ID des macros hôtes créées sous la propriété `hostmacroids`. L’ordre des ID renvoyés correspond à l’ordre des macros hôtes transmises.

Exemples

Création d’une macro d’hôte

Créez une macro d’hôte `{$SNMP_COMMUNITY}` avec la valeur ”public” sur l’hôte ”10198”.

Request:

```
{
    "jsonrpc": "2.0",
    "method": "usermacro.create",
    "params": {
        "hostid": "10198",
        "macro": "{$SNMP_COMMUNITY}",
        "value": "public"
    },
    "id": 1
}
```

Réponse:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostmacroids": [
            "11"
        ]
    },
    "id": 1
}
```

Source

CUserMacro::create() dans *ui/include/classes/api/services/CUserMacro.php*.

usermacro.createglobal

Description

`object usermacro.createglobal(object/array globalMacros)`

Cette méthode permet de créer de nouvelles macros globales.

> **Note:**
> Cette méthode est disponible uniquement pour le type d'utilisateur *Super admin*. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres des rôles utilisateur. Voir User roles pour plus d'informations.

Paramètres

`(object/array)` Macros globales à créer.

La méthode accepte des macros globales avec les propriétés standard des macros globales.

Valeurs retournées

`(object)` Retourne un objet contenant les identifiants des macros globales créées sous la propriété `globalmacroids`. L'ordre des ID retournés correspond à l'ordre des macros globales transmises.

Exemples

Création d'une macro globale

Créez une macro globale `{$SNMP_COMMUNITY}` avec la valeur "public".

Request:

```
{
    "jsonrpc": "2.0",
    "method": "usermacro.createglobal",
    "params":  {
        "macro": "{$SNMP_COMMUNITY}",
        "value": "public"
    },
    "id": 1
}
```

Response:

```
{
    "jsonrpc": "2.0",
    "result": {
        "globalmacroids": [
            "6"
        ]
    },
    "id": 1
}
```

Source

CUserMacro::createGlobal() dans *ui/include/classes/api/services/CUserMacro.php*.

usermacro.delete

Description

`object usermacro.delete(array hostMacroIds)`

Cette méthode vous permet de supprimer des macros d'hôte.

> **Note:**
> Cette méthode est disponible uniquement pour les types d'utilisateur *Admin* et *Super admin*. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres des rôles utilisateur. Consultez Rôles utilisateur pour plus d'informations.

Paramètres

`(array)` ID des macros d'hôte à supprimer.

Valeurs de retour

`(object)` Renvoie un objet contenant les ID des macros d'hôte supprimées dans la propriété `hostmacroids`.

Exemples

Suppression de plusieurs macros d'hôte

Supprimez deux macros d'hôte.

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "usermacro.delete",
    "params": [
        "32",
        "11"
    ],
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostmacroids": [
            "32",
            "11"
        ]
    },
    "id": 1
}
```

Source

CUserMacro::delete() dans *ui/include/classes/api/services/CUserMacro.php*.

usermacro.deleteglobal

Description

`object usermacro.deleteglobal(array globalMacroIds)`

Cette méthode vous permet de supprimer des macros globales.

> **Note:**
> Cette méthode est disponible uniquement pour le type d'utilisateur *Super admin*. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres du rôle utilisateur. Consultez User roles pour plus d'informations.

Paramètres

`(array)` ID des macros globales à supprimer.

Valeurs de retour

`(object)` Renvoie un objet contenant les ID des macros globales supprimées dans la propriété `globalmacroids`.

Exemples

Suppression de plusieurs macros globales

Supprimez deux macros globales.

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "usermacro.deleteglobal",
    "params": [
        "32",
        "11"
    ],
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "globalmacroids": [
            "32",
            "11"
        ]
    },
    "id": 1
}
```

Source

CUserMacro::deleteGlobal() dans *ui/include/classes/api/services/CUserMacro.php*.

usermacro.get

Description

`integer/array usermacro.get(object parameters)`

La méthode permet de récupérer les macros d'hôte et globales selon les paramètres fournis.

> **Note:**
> Cette méthode est disponible pour les utilisateurs de tout type. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres des rôles utilisateur. Voir Rôles utilisateur pour plus d'informations.

Paramètres

`(object)` Paramètres définissant la sortie souhaitée.

La méthode prend en charge les paramètres suivants.

Paramètre	Type	Description
globalmacro	flag	Renvoyer les macros globales au lieu des macros d'hôte.
globalmacroids	ID/array	Renvoyer uniquement les macros globales avec les ID donnés.
groupids	ID/array	Renvoyer uniquement les macros d'hôte qui appartiennent à des hôtes ou à des modèles des groupes d'hôtes ou groupes de modèles donnés.
hostids	ID/array	Renvoyer uniquement les macros qui appartiennent aux hôtes ou modèles donnés.
hostmacroids	ID/array	Renvoyer uniquement les macros d'hôte avec les ID donnés.
inherited	boolean	Si défini sur `true`, renvoyer uniquement les macros utilisateur des prototypes d'hôte héritées d'un modèle.
selectHostGroups	query	Renvoyer les groupes d'hôtes auxquels appartient la macro d'hôte dans la propriété `hostgroups`.
selectHosts	query	Utilisé uniquement lors de la récupération des macros d'hôte. Renvoyer les hôtes auxquels appartient la macro d'hôte dans la propriété `hosts`.
selectTemplateGroups	query	Utilisé uniquement lors de la récupération des macros d'hôte. Renvoyer les groupes de modèles auxquels appartient la macro de modèle dans la propriété `templategroups`.
selectTemplates	query	Utilisé uniquement lors de la récupération des macros de modèle. Renvoyer les modèles auxquels appartient la macro d'hôte dans la propriété `templates`.
sortfield	string/array	Utilisé uniquement lors de la récupération des macros d'hôte. Trier le résultat selon les propriétés données. Valeurs possibles : `macro`.
countOutput	boolean	Ces paramètres sont décrits dans le commentaire de référence.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Valeurs retournées

`(integer/array)` Renvoie soit :

- un tableau d'objets ;
- le nombre d'objets récupérés, si le paramètre `countOutput` a été utilisé.

Exemples

Récupération des macros d'hôte pour un hôte

Récupérez toutes les macros d'hôte définies pour l'hôte « 10198 ».

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "usermacro.get",
    "params": {
        "output": "extend",
```

```
        "hostids": "10198"
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "hostmacroid": "9",
            "hostid": "10198",
            "macro": "{$INTERFACE}",
            "value": "eth0",
            "description": "",
            "type": "0",
            "automatic": "0"
        },
        {
            "hostmacroid": "11",
            "hostid": "10198",
            "macro": "{$SNMP_COMMUNITY}",
            "value": "public",
            "description": "",
            "type": "0",
            "automatic": "0"
        }
    ],
    "id": 1
}
```

Récupération des macros d'hôte pour un modèle

Récupérez les macros d'hôte définies pour le modèle "10265" qui contiennent "STATUS" dans le nom de la macro.

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "usermacro.get",
    "params": {
        "output": "extend",
        "hostids": "10265",
        "search": {
            "macro": "STATUS"
        }
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "hostmacroid": "6709",
            "hostid": "10265",
            "macro": "{$APACHE.STATUS.HOST}",
            "value": "",
            "description": "Le nom d'hôte ou l'adresse IP de l'hôte de la page de statut Apache.",
            "type": "0",
            "automatic": "0",
            "config": {
                "type": "1",
```

```
                "priority": "1",
                "section_name": "",
                "label": "Hôte de statut Apache",
                "description": "Nom d'hôte ou adresse IP de l'hôte de la page de statut Apache.",
                "required": "1",
                "regex": "",
                "options": []
            }
        },
        {
            "hostmacroid": "814",
            "hostid": "10265",
            "macro": "{$APACHE.STATUS.PATH}",
            "value": "server-status?auto",
            "description": "Le chemin URL de la page de statut Apache.",
            "type": "0",
            "automatic": "0",
            "config": {
                "type": "1",
                "priority": "3",
                "section_name": "",
                "label": "Chemin de la page de statut Apache",
                "description": "Chemin URL de la page de statut Apache.",
                "required": "1",
                "regex": "",
                "options": []
            }
        },
        {
            "hostmacroid": "815",
            "hostid": "10265",
            "macro": "{$APACHE.STATUS.PORT}",
            "value": "80",
            "description": "Le port de la page de statut Apache.",
            "type": "0",
            "automatic": "0",
            "config": {
                "type": "1",
                "priority": "2",
                "section_name": "",
                "label": "Port de la page de statut Apache",
                "description": "Dans la plage de 1 à 65535 inclus.",
                "required": "1",
                "regex": "^-?([0-9]+|(([0-9]+)\\.([0-9]+)))$",
                "options": []
            }
        },
        {
            "hostmacroid": "816",
            "hostid": "10265",
            "macro": "{$APACHE.STATUS.SCHEME}",
            "value": "http",
            "description": "Le schéma de requête, qui peut être HTTP ou HTTPS.",
            "type": "0",
            "automatic": "0",
            "config": {
                "type": "2",
                "priority": "4",
                "section_name": "",
                "label": "Schéma de requête",
                "description": "Schéma de requête, qui peut être HTTP ou HTTPS.",
                "required": "0",
```

```
                "regex": "",
                "options": [
                    {
                        "value": "http",
                        "text": "HTTP"
                    },
                    {
                        "value": "https",
                        "text": "HTTPS"
                    }
                ]
            }
        }
    ],
    "id": 1
}
```

Récupération des macros globales

Récupérez toutes les macros globales.

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "usermacro.get",
    "params": {
        "output": "extend",
        "globalmacro": true
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "globalmacroid": "6",
            "macro": "{$SNMP_COMMUNITY}",
            "value": "public",
            "description": "",
            "type": "0"
        }
    ],
    "id": 1
}
```

Source

CUserMacro::get() dans *ui/include/classes/api/services/CUserMacro.php*.

usermacro.update

Description

`object usermacro.update(object/array hostMacros)`

Cette méthode vous permet de mettre à jour des macros d'hôte existantes.

> **Note:**
> Cette méthode est disponible uniquement pour les types d'utilisateur *Admin* et *Super admin*. Les autorisations d'appel de cette méthode peuvent être révoquées dans les paramètres du rôle utilisateur. Consultez Rôles utilisateur pour plus d'informations.

Paramètres

`(object/array)` Propriétés de macro d'hôte à mettre à jour.

La propriété `hostmacroid` doit être définie pour chaque macro d'hôte, toutes les autres propriétés sont facultatives. Seules les propriétés transmises seront mises à jour, toutes les autres resteront inchangées.

Valeurs retournées

`(object)` Retourne un objet contenant les identifiants des macros d'hôtes mises à jour sous la propriété `hostmacroids`.

Exemples

Modification de la valeur d'une macro d'hôte

Modifiez la valeur d'une macro d'hôte en "public".

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "usermacro.update",
    "params": {
        "hostmacroid": "1",
        "value": "public"
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostmacroids": [
            "1"
        ]
    },
    "id": 1
}
```

Modifier la valeur d'une macro créée par une règle de découverte

Convertissez la macro « automatique » créée par la règle de découverte en macro « manuelle » et modifiez sa valeur en « new-value ».

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "usermacro.update",
    "params": {
        "hostmacroid": "1",
        "value": "new-value",
        "automatic": "0"
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostmacroids": [
            "1"
        ]
    },
    "id": 1
}
```

Source

CUserMacro::update() dans *ui/include/classes/api/services/CUserMacro.php*.

usermacro.updateglobal

Description

`object usermacro.updateglobal(object/array globalMacros)`

Cette méthode permet de mettre à jour des macros globales existantes.

> **Note:**
> Cette méthode est disponible uniquement pour le type d'utilisateur *Super admin*. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres du rôle utilisateur. Voir User roles pour plus d'informations.

Paramètres

`(object/array)` Propriétés de macro globale à mettre à jour.

La propriété `globalmacroid` doit être définie pour chaque macro globale, toutes les autres propriétés sont facultatives. Seules les propriétés transmises seront mises à jour, toutes les autres resteront inchangées.

Valeurs retournées

`(object)` Retourne un objet contenant les identifiants des macros globales mises à jour sous la propriété `globalmacroids`.

Exemples

Modification de la valeur d'une macro globale

Modifiez la valeur d'une macro globale en "public".

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "usermacro.updateglobal",
    "params": {
        "globalmacroid": "1",
        "value": "public"
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "globalmacroids": [
            "1"
        ]
    },
    "id": 1
}
```

Source

CUserMacro::updateGlobal() dans *ui/include/classes/api/services/CUserMacro.php*.

Maintenance

Cette classe est conçue pour fonctionner avec les maintenances.

Références d'objet :

- Maintenance
- Période de temps
- Tag de problème

Méthodes disponibles :

- maintenance.create - créer de nouvelles maintenances
- maintenance.delete - supprimer des maintenances
- maintenance.get - récupérer des maintenances
- maintenance.update - mettre à jour des maintenances

Objet de maintenance

Les objets suivants sont directement liés à l'API `maintenance`.

Maintenance

L'objet maintenance possède les propriétés suivantes.

Property	Type	Description
maintenanceid	ID	ID de la maintenance. Comportement de la propriété: - *lecture seule* - *obligatoire* pour les opérations de mise à jour
name	string	Nom de la maintenance. Comportement de la propriété: - *obligatoire* pour les opérations de création
active_since	timestamp	Heure à laquelle la maintenance devient active (inclus). La valeur fournie sera arrondie à la minute inférieure. Comportement de la propriété: - *obligatoire* pour les opérations de création
active_till	timestamp	Heure à laquelle la maintenance cesse d'être active (exclus). La valeur fournie sera arrondie à la minute inférieure. Comportement de la propriété: - *obligatoire* pour les opérations de création
description	string	Description de la maintenance.
maintenance_type	integer	Type de maintenance. Valeurs possibles : 0 - *(par défaut)* avec collecte de données ; 1 - sans collecte de données.
tags_evaltype	integer	Méthode d'évaluation des tags de problème. Valeurs possibles : 0 - *(par défaut)* Et/Ou ; 2 - Ou.

Période de temps

L'objet de période de temps est utilisé pour définir les périodes pendant lesquelles la maintenance doit entrer en vigueur. Il possède les propriétés suivantes.

Propriété	Type	Description
period	integer	Durée de la période de maintenance en secondes. La valeur fournie sera arrondie à la minute inférieure. Plage de valeurs possibles : 300 - 86399940. Par défaut : 3600.

Propriété	Type	Description
timeperiod_type	integer	Type de période de temps. Valeurs possibles : 0 - *(par défaut)* une seule fois ; 2 - quotidien ; 3 - hebdomadaire ; 4 - mensuel.
start_date	timestamp	Date à laquelle la période de maintenance doit entrer en vigueur. La valeur fournie sera arrondie à la minute inférieure. Par défaut : date actuelle. Comportement de la propriété : - *pris en charge* si `timeperiod_type` est défini sur ”one time only”
start_time	integer	Heure de la journée à laquelle la maintenance commence, en secondes. La valeur fournie sera arrondie à la minute inférieure. Par défaut : 0. Comportement de la propriété : - *pris en charge* si `timeperiod_type` est défini sur ”daily”, ”weekly” ou ”monthly”
every	integer	Pour les périodes quotidiennes et hebdomadaires, `every` définit les intervalles de jours ou de semaines auxquels la maintenance doit entrer en vigueur. Valeur par défaut si `timeperiod_type` est défini sur ”daily” ou ”weekly” : 1. Pour les périodes mensuelles lorsque `day` est défini, la propriété `every` définit le jour du mois auquel la maintenance doit entrer en vigueur. Valeur par défaut si `timeperiod_type` est défini sur ”monthly” et que `day` est défini : 1. Pour les périodes mensuelles lorsque `dayofweek` est défini, la propriété `every` définit la semaine du mois à laquelle la maintenance doit entrer en vigueur. Valeurs possibles si `timeperiod_type` est défini sur ”monthly” et que `dayofweek` est défini : 1 - *(par défaut)* première semaine ; 2 - deuxième semaine ; 3 - troisième semaine ; 4 - quatrième semaine ; 5 - dernière semaine. Comportement de la propriété : - *pris en charge* si `timeperiod_type` est défini sur ”daily”, ”weekly” ou ”monthly”

Propriété	Type	Description
dayofweek	integer	Jours de la semaine auxquels la maintenance doit entrer en vigueur. Valeurs bitmap possibles : 1 - lundi ; 2 - mardi ; 4 - mercredi ; 8 - jeudi ; 16 - vendredi ; 32 - samedi ; 64 - dimanche. Il s'agit d'un champ de masque de bits ; toute somme des valeurs bitmap possibles est acceptée (par exemple, 21 pour lundi, mercredi et vendredi). Comportement de la propriété : - *requis* si `timeperiod_type` est défini sur "weekly" ou si `timeperiod_type` est défini sur "monthly" et que `day` n'est pas défini
day	integer	Jour du mois auquel la maintenance doit entrer en vigueur. Comportement de la propriété : - *requis* si `timeperiod_type` est défini sur "monthly" et que `dayofweek` n'est pas défini
month	integer	Mois auxquels la maintenance doit entrer en vigueur. Valeurs bitmap possibles : 1 - janvier ; 2 - février ; 4 - mars ; 8 - avril ; 16 - mai ; 32 - juin ; 64 - juillet ; 128 - août ; 256 - septembre ; 512 - octobre ; 1024 - novembre ; 2048 - décembre. Il s'agit d'un champ de masque de bits ; toute somme des valeurs bitmap possibles est acceptée (par exemple, 585 pour janvier, avril, juillet et octobre). Comportement de la propriété : - *requis* si `timeperiod_type` est défini sur "monthly"

Nom de l'événement

L'objet de nom d'événement est utilisé pour définir les problèmes qui doivent être supprimés lorsque la maintenance entre en vigueur. Les noms d'événements ne peuvent être spécifiés que si `maintenance_type` de l'objet Maintenance est défini sur « avec collecte de données ». Il possède les propriétés suivantes.

Property	Type	Description
operator	integer	Opérateur de la condition. Valeurs possibles : 2 - *(par défaut)* Contient ; 3 - Ne contient pas.

Property	Type	Description
value	string	Valeur du nom de l'événement. Comportement de la propriété : - *obligatoire*

Étiquette de problème

L'objet d'étiquette de problème permet de définir quels problèmes doivent être supprimés lorsque la maintenance entre en vigueur. Les étiquettes ne peuvent être spécifiées que si `maintenance_type` de l'objet Maintenance est défini sur « avec collecte de données ». Il possède les propriétés suivantes.

Property	Type	Description
tag	string	Nom de l'étiquette de problème. Comportement de la propriété : - *obligatoire*
operator	integer	Opérateur de la condition. Valeurs possibles : 0 - Est égal à ; 1 - N'est pas égal à ; 2 - *(par défaut)* Contient ; 3 - Ne contient pas.
value	string	Valeur de l'étiquette de problème.

maintenance.create

Description

`object maintenance.create(object/array maintenances)`

Cette méthode vous permet de créer de nouvelles maintenances.

Note:
Cette méthode est disponible uniquement pour les types d'utilisateur *Admin* et *Super admin*. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres du rôle utilisateur. Voir Rôles d'utilisateur pour plus d'informations.

Paramètres

`(object/array)` Maintenances à créer.

En plus des propriétés standard des maintenances, la méthode accepte les paramètres suivants.

Parameter	Type	Description
groups	object/array	Groupes d'hôtes à placer en maintenance. Les groupes d'hôtes doivent avoir uniquement la propriété `groupid` définie. Comportement du paramètre : - *obligatoire* si `hosts` n'est pas défini
hosts	object/array	Hôtes à placer en maintenance. Les hôtes doivent avoir uniquement la propriété `hostid` définie. Comportement du paramètre : - *obligatoire* si `groups` n'est pas défini

Parameter	Type	Description
triggers	object/array	Déclencheurs à placer en maintenance. Les déclencheurs doivent avoir uniquement la propriété `triggerid` définie. Comportement du paramètre : - *obligatoire* si `groups` et `hosts` ne sont pas définis - *pris en charge* si maintenance_type=0 (avec collecte de données)
event_names	object/array	Noms d'événements permettant de définir les problèmes à placer en maintenance. Comportement du paramètre : - *pris en charge* si maintenance_type=0 (avec collecte de données)
tags	object/array	Étiquettes de problèmes permettant de définir les problèmes à placer en maintenance. Comportement du paramètre : - *pris en charge* si `maintenance_type` de l'objet Maintenance est défini sur « avec collecte de données »
timeperiods	object/array	Périodes de maintenance. Comportement du paramètre : - *obligatoire*

Valeurs retournées

`(object)` Retourne un objet contenant les identifiants des maintenance créées sous la propriété `maintenanceids`. L'ordre des ID retournés correspond à l'ordre des maintenances transmises.

Exemples

Création d'une maintenance

Créez une maintenance avec collecte des données pour le groupe de hôtes ayant l'ID « 20 », et avec les tags de problème **service:mysqld** et **error**.
Elle doit être active du 17.03.2026 au 17.03.2027, prendre effet chaque dimanche à 18:00 et durer une heure.

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "maintenance.create",
    "params": {
        "name": "Sunday maintenance",
        "active_since": 1773720240,
        "active_till": 1805256240,
        "tags_evaltype": 0,
        "groups": [
            {"groupid": "20"}
        ],
        "timeperiods": [
            {
                "period": 3600,
                "timeperiod_type": 3,
                "start_time": 64800,
                "every": 1,
                "dayofweek": 64
            }
        ],
        "tags": [
            {
                "tag": "service",
                "operator": "0",
                "value": "mysqld"
```

```
            },
            {
                "tag": "error",
                "operator": "2",
                "value": ""
            }
        ]
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "maintenanceids": [
            "3"
        ]
    },
    "id": 1
}
```

Création d'une maintenance ad hoc

Créez une maintenance ad hoc d'une heure avec collecte des données pour le groupe d'hôtes dont l'ID est « 4 », le déclencheur dont l'ID est « 12345 » et le nom d'événement **Printer: ink low**.

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "maintenance.create",
    "params": {
        "name": "Ad-hoc maintenance",
        "active_since": 1792057740,
        "active_till": 1792061340,
        "tags_evaltype": 0,
        "groups": [
            {"groupid": "4"}
        ],
        "triggers": [
            {"triggerid": "12345"}
        ],
        "timeperiods": [
            {
                "period": 3600,
                "timeperiod_type": 0,
                "start_date": 1792057740
            }
        ],
        "event_names": [
            {
                "operator": "2",
                "value": "Printer: ink low"
            }
        ]
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
```

```
        "maintenanceids": [
            "4"
        ]
    },
    "id": 1
}
```

Voir aussi

- Hôte
- Groupe d'hôtes
- Déclencheur
- Nom de l'événement
- Étiquette du problème
- Période

Source

CMaintenance::create() dans *ui/include/classes/api/services/CMaintenance.php*.

maintenance.delete

Description

`object maintenance.delete(array maintenanceIds)`

Cette méthode vous permet de supprimer des périodes de maintenance.

> **Note:**
> Cette méthode est disponible uniquement pour les types d'utilisateur *Admin* et *Super admin*. Les autorisations d'appel de cette méthode peuvent être révoquées dans les paramètres des rôles utilisateur. Consultez Rôles utilisateur pour plus d'informations.

Paramètres

`(array)` ID des périodes de maintenance à supprimer.

Valeurs retournées

`(object)` Retourne un objet contenant les identifiants des maintenances supprimées sous la propriété `maintenanceids`.

Exemples

Suppression de plusieurs périodes de maintenance

Supprimez deux périodes de maintenance.

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "maintenance.delete",
    "params": [
        "3",
        "1"
    ],
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "maintenanceids": [
            "3",
            "1"
        ]
    },
```

```
    "id": 1
}
```

Source

CMaintenance::delete() dans *ui/include/classes/api/services/CMaintenance.php*.

maintenance.get

Description

`integer/array maintenance.get(object parameters)`

La méthode permet de récupérer les maintenances selon les paramètres fournis.

> **Note:**
> Cette méthode est disponible pour les utilisateurs de tout type. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres du rôle utilisateur. Consultez User roles pour plus d'informations.

Paramètres

`(object)` Paramètres définissant le résultat souhaité.

La méthode prend en charge les paramètres suivants.

Paramètre	Type	Description
groupids	ID/array	Retourne uniquement les maintenances affectées aux groupes d'hôtes indiqués.
hostids	ID/array	Retourne uniquement les maintenances affectées aux hôtes indiqués.
triggerids	ID/array	Retourne uniquement les maintenances affectées aux déclencheurs indiqués.
maintenanceids	ID/array	Retourne uniquement les maintenances possédant les ID indiqués.
selectHostGroups	query	Retourne une propriété `hostgroups` contenant les groupes d'hôtes affectés à la maintenance.
selectHosts	query	Retourne une propriété `hosts` contenant les hôtes affectés à la maintenance.
selectTriggers	query	Retourne une propriété `triggers` contenant les déclencheurs affectés à la maintenance.
selectEventNames	query	Retourne une propriété `event_names` contenant les noms d'événements affectés à la maintenance.
selectTags	query	Retourne une propriété `tags` contenant les balises de problème de la maintenance.
selectTimeperiods	query	Retourne une propriété `timeperiods` contenant les périodes de la maintenance.
sortfield	string/array	Trie le résultat selon les propriétés indiquées. Valeurs possibles : `maintenanceid`, `name`, `maintenance_type`, `active_since`, `active_till`.
countOutput	boolean	Ces paramètres sont décrits dans le commentaire de référence.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Valeurs de retour

`(integer/array)` Renvoie soit :

- un tableau d'objets ;
- le nombre d'objets récupérés, si le paramètre `countOutput` a été utilisé.

Exemples

Récupération des maintenances

Récupérez toutes les maintenances configurées, les données relatives à la portée de maintenance attribuée (groupe d'hôtes, hôte, déclencheur, nom de l'événement, balises), ainsi que les périodes définies.

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "maintenance.get",
    "params": {
        "output": "extend",
        "selectHostGroups": "extend",
        "selectHosts": "extend",
        "selectTriggers": "extend",
        "selectEventNames": "extend",
        "selectTags": "extend",
        "selectTimeperiods": "extend"
    },
    "id": 1
}
```

Réponse :

```
{
  "jsonrpc": "2.0",
  "result": [
    {
      "maintenanceid": "1",
      "name": "Maintenance régulière de la base de données",
      "maintenance_type": "0",
      "description": "",
      "active_since": "1791936000",
      "active_till": "2107555200",
      "tags_evaltype": "0",
      "hostgroups": [
        {
          "groupid": "20",
          "name": "Bases de données",
          "flags": "0",
          "uuid": "748ad4d098d447d492bb935c907f652f"
        }
      ],
      "triggers": [],
      "event_names": [],
      "hosts": [],
      "tags": [],
      "timeperiods": [
        {
          "timeperiod_type": "4",
          "every": "1",
          "month": "4095",
          "dayofweek": "0",
          "day": "1",
          "start_time": "64800",
          "period": "3600",
          "start_date": "0"
        }
      ]
    },
    {
```

```
      "maintenanceid": "2",
      "name": "Ad hoc : Imprimante : encre ÉPUISÉE",
      "maintenance_type": "0",
      "description": "",
      "active_since": "1792057740",
      "active_till": "1792061340",
      "tags_evaltype": "0",
      "hostgroups": [
        {
          "groupid": "4",
          "name": "Serveurs Zabbix",
          "flags": "0",
          "uuid": "6f6799aa69e844b4b3918f779f2abf08"
        }
      ],
      "triggers": [],
      "event_names": [
        {
          "operator": "2",
          "value": "Imprimante : encre ÉPUISÉE"
        }
      ],
      "hosts": [],
      "tags": [],
      "timeperiods": [
        {
          "timeperiod_type": "0",
          "every": "1",
          "month": "0",
          "dayofweek": "0",
          "day": "0",
          "start_time": "0",
          "period": "3600",
          "start_date": "1792057740"
        }
      ]
    }
  ],
  "id": 1
}
```


Voir aussi

- Hôte
- Groupe d'hôtes
- Déclencheur
- Nom de l'événement
- Étiquette du problème
- Période

Source

CMaintenance::get() dans *ui/include/classes/api/services/CMaintenance.php*.

maintenance.update

Description

`object maintenance.update(object/array maintenances)`

Cette méthode vous permet de mettre à jour des maintenances existantes.

Note:
Cette méthode est disponible uniquement pour les types d'utilisateur *Admin* et *Super admin*. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres du rôle utilisateur. Consultez Rôles utilisateur pour plus d'informations.

Paramètres

`(object/array)` Propriétés de maintenance à mettre à jour.

La propriété `maintenanceid` doit être définie pour chaque maintenance ; toutes les autres propriétés sont facultatives. Seules les propriétés transmises seront mises à jour, toutes les autres resteront inchangées.

En plus des propriétés de maintenance standard, la méthode accepte les paramètres suivants.

Parameter	Type	Description
groups	object/array	Groupes d'hôtes qui remplaceront les groupes actuels. Les groupes d'hôtes doivent uniquement avoir la propriété `groupid` définie.
hosts	object/array	Comportement du paramètre : - *requis* si `hosts` n'est pas défini Hôtes qui remplaceront les hôtes actuels. Les hôtes doivent uniquement avoir la propriété `hostid` définie.
triggers	object/array	Comportement du paramètre : - *requis* si `groups` n'est pas défini Déclencheurs qui remplaceront les déclencheurs actuels. Les déclencheurs doivent uniquement avoir la propriété `triggerid` définie.
event_names	object/array	Comportement du paramètre : - *requis* si `groups` et `hosts` ne sont pas définis - *pris en charge* si maintenance_type=0 (avec collecte des données) Noms d'événements qui remplaceront les noms d'événements actuels.
tags	object/array	Comportement du paramètre : - *pris en charge* si maintenance_type=0 (avec collecte des données) Étiquettes de problème qui remplaceront les étiquettes actuelles.
timeperiods	object/array	Comportement du paramètre : - *pris en charge* si `maintenance_type` de l'objet Maintenance est défini sur « avec collecte des données » Périodes de maintenance qui remplaceront les périodes actuelles.

Valeurs retournées

`(object)` Retourne un objet contenant les identifiants des maintenances mises à jour sous la propriété `maintenanceids`.

Exemples

Attribution d'hôtes différents

Remplacez les hôtes actuellement attribués à la maintenance par deux autres.

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "maintenance.update",
    "params": {
        "maintenanceid": "3",
        "hosts": [
```

```
            {"hostid": "10085"},
            {"hostid": "10084"}
        ]
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "maintenanceids": [
            "3"
        ]
    },
    "id": 1
}
```

Voir aussi

- Période de temps

Source

CMaintenance::update() dans *ui/include/classes/api/services/CMaintenance.php*.

MFA

Cette classe est conçue pour fonctionner avec les méthodes MFA (authentification multifacteur).

Références d'objet :

- MFA

Méthodes disponibles :

- mfa.create - créer de nouvelles méthodes MFA
- mfa.delete - supprimer des méthodes MFA
- mfa.get - récupérer des méthodes MFA
- mfa.update - mettre à jour des méthodes MFA

Objet MFA

Les objets suivants sont directement liés à l'API `mfa`.

MFA

L'objet MFA (authentification multifacteur) possède les propriétés suivantes.

Propriété	Type	Description
mfaid	ID	ID de la méthode MFA. Comportement de la propriété : - *lecture seule* - *obligatoire* pour les opérations de mise à jour
type	integer	Type de la méthode MFA. Valeurs possibles : 1 - TOTP (mots de passe à usage unique basés sur le temps) ; 2 - Duo Universal Prompt.
name	string	Nom unique de la méthode MFA. Comportement de la propriété : - *obligatoire* pour les opérations de création

Propriété	Type	Description
hash_function	integer	Type de la fonction de hachage pour générer les codes TOTP. Valeurs possibles : 1 - SHA-1 ; 2 - SHA-256 ; 3 - SHA-512. Comportement de la propriété : - *obligatoire* si `type` est défini sur "TOTP"
code_length	integer	Longueur du code de vérification. Valeurs possibles : 6 - sur 6 chiffres ; 8 - sur 8 chiffres. Comportement de la propriété : - *obligatoire* si `type` est défini sur "TOTP"
api_hostname	string	Nom d'hôte API fourni par le service d'authentification Duo. Comportement de la propriété : - *obligatoire* si `type` est défini sur "Duo Universal Prompt"
clientid	string	ID client fourni par le service d'authentification Duo. Comportement de la propriété : - *obligatoire* si `type` est défini sur "Duo Universal Prompt"
client_secret	string	Secret client fourni par le service d'authentification Duo. Comportement de la propriété : - *écriture seule* - *obligatoire* si `type` est défini sur "Duo Universal Prompt"

mfa.create

Description

`object mfa.create(object/array MFA methods)`

Cette méthode permet de créer de nouvelles méthodes MFA.

> **Note:**
> Cette méthode est disponible uniquement pour le type d'utilisateur *Super admin*. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres des rôles utilisateur. Consultez Rôles utilisateur pour plus d'informations.

Paramètres

`(object/array)` Méthodes MFA à créer.

La méthode accepte des méthodes MFA avec les propriétés standard des méthodes MFA.

Valeurs de retour

`(object)` Renvoie un objet contenant les ID des méthodes MFA créées dans la propriété `mfaids`. L'ordre des ID renvoyés correspond à l'ordre des éléments transmis.

Exemples

Création de méthodes MFA

Créez une méthode MFA « Zabbix TOTP » utilisant des mots de passe à usage unique basés sur le temps (TOTP), avec la fonction de hachage utilisée pour générer les codes TOTP définie sur SHA-1 et la longueur du code de vérification définie sur 6 chiffres.

Requête:

```
{
    "jsonrpc": "2.0",
```

```
    "method": "mfa.create",
    "params": {
        "type": 1,
        "name": "Zabbix TOTP",
        "hash_function": 1,
        "code_length": 6
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "mfaids": [
            "1"
        ]
    },
    "id": 1
}
```

Source

CMfa::create() dans *ui/include/classes/api/services/CMfa.php*.

mfa.delete

Description

`object mfa.delete(array mfaids)`

Cette méthode vous permet de supprimer des méthodes MFA.

> **Note:**
> Cette méthode est disponible uniquement pour le type d'utilisateur *Super admin*. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres des rôles utilisateur. Consultez User roles pour plus d'informations.

Paramètres

`(array)` ID des méthodes MFA à supprimer.

Valeurs de retour

`(object)` Renvoie un objet contenant les ID des méthodes MFA supprimées dans la propriété `mfaids`.

Exemples

Suppression de méthodes MFA

Supprimez une méthode MFA.

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "mfa.delete",
    "params": [
        "2"
    ],
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "mfaids": [
```

```
            "2"
        ]
    },
    "id": 1
}
```

Source

CMfa::delete() dans *ui/include/classes/api/services/CMfa.php*.

mfa.get

Description

`integer/array mfa.get(object parameters)`

La méthode permet de récupérer les méthodes MFA selon les paramètres fournis.

> **Note:**
> Cette méthode est disponible uniquement pour le type d'utilisateur *Super admin*. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres du rôle utilisateur. Consultez Rôles utilisateur pour plus d'informations.

Paramètres

`(object)` Paramètres définissant la sortie souhaitée.

La méthode prend en charge les paramètres suivants.

Paramètre	Type	Description
mfaids	ID/array	Retourner uniquement les méthodes MFA avec les ID donnés.
selectUsrgrps	query	Retourner une propriété `usrgrps` avec les groupes d'utilisateurs associés aux méthodes MFA. Prend en charge `count`.
filter	object	Retourner uniquement les résultats qui correspondent exactement au filtre donné. Accepte un objet, où les clés sont des noms de propriété et les valeurs sont soit une valeur unique, soit un tableau de valeurs à comparer. Prend en charge les propriétés suivantes : `mfaid` - ID de la méthode MFA ; `type` - type de la méthode MFA.
sortfield	string/array	Trier le résultat selon les propriétés données. Valeurs possibles : `name`.
search	object	Retourner les résultats qui correspondent au motif donné (insensible à la casse). Accepte un objet, où les clés sont des noms de propriété et les valeurs sont des chaînes à rechercher. Si aucune option supplémentaire n'est fournie, cela effectuera une recherche `LIKE "%…%"`. Propriétés prises en charge : `name`.
countOutput	boolean	Ces paramètres sont décrits dans les commentaires de référence.
excludeSearch	boolean	
limit	integer	
output	query	
preservekeys	boolean	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Valeurs de retour

`(integer/array)` Renvoie soit :

- un tableau d'objets ;
- le nombre d'objets récupérés, si le paramètre `countOutput` a été utilisé.

Exemples

Recherche de méthodes MFA par nom

Récupérez toutes les méthodes MFA dont le nom contient « Zabbix ».

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "mfa.get",
    "params": {
        "output": "extend",
        "search": {
            "name": "Zabbix"
        }
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "mfaid": "1",
            "type": "1",
            "name": "Zabbix TOTP 1",
            "hash_function": "1",
            "code_length": "6",
            "api_hostname": "",
            "clientid": ""
        },
        {
            "mfaid": "2",
            "type": "1",
            "name": "Zabbix TOTP 2",
            "hash_function": "3",
            "code_length": "8",
            "api_hostname": "",
            "clientid": ""
        }
    ],
    "id": 1
}
```

Source

CMfa::get() dans *ui/include/classes/api/services/CMfa.php*.

mfa.update

Description

`object mfa.update(object/array MFA methods)`

Cette méthode vous permet de mettre à jour des méthodes MFA existantes.

Note:
Cette méthode est disponible uniquement pour le type d'utilisateur *Super admin*. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres des rôles utilisateur. Consultez Rôles utilisateur pour plus d'informations.

Paramètres

`(object/array)` Propriétés des méthodes MFA à mettre à jour.

La propriété `mfaid` doit être définie pour chaque élément, toutes les autres propriétés sont facultatives. Seules les propriétés transmises seront mises à jour, toutes les autres resteront inchangées.

La méthode accepte les méthodes MFA avec les propriétés standard des méthodes MFA.

Valeurs de retour

`(object)` Renvoie un objet contenant les identifiants des méthodes MFA mises à jour dans la propriété `mfaids`.

Exemples

Mise à jour des propriétés de la méthode

Mettez à jour la fonction de hachage utilisée pour générer les codes TOTP ainsi que la longueur du code de vérification pour la méthode MFA « Zabbix TOTP » utilisant des mots de passe à usage unique basés sur le temps (TOTP).

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "mfa.update",
    "params": {
        "mfaid": "1",
        "hash_function": 3,
        "code_length": 8
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "mfaids": [
            "1"
        ]
    },
    "id": 1
}
```

Source

CMfa::update() dans *ui/include/classes/api/services/CMfa.php*.

Module

Cette classe est conçue pour fonctionner avec les modules du frontend.

Références d'objet :

- Module

Méthodes disponibles :

- module.create - installer de nouveaux modules
- module.delete - désinstaller des modules
- module.get - récupérer des modules
- module.update - mettre à jour des modules

Objet module

Les objets suivants sont directement liés à l'API `module`.

Module

L'objet module possède les propriétés suivantes.

Property	Type	Description
moduleid	ID	ID du module tel qu'il est stocké dans la base de données. Comportement de la propriété: - *lecture seule* - *obligatoire* pour les opérations de mise à jour
id	string	ID unique du module, défini par un développeur dans le fichier `manifest.json` du module. Valeurs possibles pour les modules intégrés : voir la description de la propriété "type" dans Dashboard widget.
relative_path	string	Comportement de la propriété: - *obligatoire* pour les opérations de création Chemin vers le répertoire du module, relatif au répertoire de l'interface Zabbix. Valeurs possibles : `widgets/*` - pour les modules de widgets intégrés ; `modules/*` - pour les modules tiers.
status	integer	Comportement de la propriété: - *obligatoire* pour les opérations de création Indique si le module est activé ou désactivé. Valeurs possibles : 0 - *(par défaut)* Désactivé ; 1 - Activé.
config	object	Configuration du module.

module.create

Description

`object module.create(object/array modules)`

Cette méthode vous permet d'installer de nouveaux modules d'interface.

Note:
Cette méthode est uniquement disponible pour le type d'utilisateur *Super administrateur*. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres des rôles utilisateur. Consultez la page Rôles utilisateur pour plus d'informations.
 Si `$ZBX_FEATURE_FLAGS['modules_config_enabled']` est défini sur `false` dans le fichier de configuration de l'interface, l'utilisateur ne sera pas autorisé à appeler cette méthode.

Attention:
Les fichiers des modules doivent être décompressés manuellement dans les sous-répertoires appropriés, correspondant à la propriété `relative_path` des modules.

Paramètres

`(object/array)` Modules à créer.

La méthode accepte des modules avec les propriétés standard des modules.

Valeurs de retour

(object) Renvoie un objet contenant les ID des modules installés dans la propriété moduleids. L'ordre des ID renvoyés correspond à l'ordre des modules transmis.

Exemples

Installation d'un module

Installez un module avec le statut « Activé ».

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "module.create",
    "params": {
        "id": "example_module",
        "relative_path": "modules/example_module",
        "status": 1
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "moduleids": [
            "25"
        ]
    },
    "id": 1
}
```

Voir aussi

- Module
- Modules frontend

Source

CModule::create() dans *ui/include/classes/api/services/CModule.php*.

module.delete

Description

`object module.delete(array moduleids)`

Cette méthode permet de désinstaller des modules.

> **Note:**
> Cette méthode est uniquement disponible pour le type d'utilisateur *Super administrateur*. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres des rôles utilisateur. Consultez Rôles utilisateur pour plus d'informations.
 Si `$ZBX_FEATURE_FLAGS['modules_config_enabled']` est défini sur `false` dans le fichier de configuration de l'interface, l'utilisateur ne sera pas autorisé à appeler cette méthode.

> **Attention:**
> Les fichiers des modules doivent être supprimés manuellement.

Paramètres

(array) ID des modules à désinstaller.

Valeurs de retour

(object) Renvoie un objet contenant les identifiants des modules désinstallés dans la propriété moduleids.

Exemples

Désinstallation de plusieurs modules

Désinstallez les modules « 27 » et « 28 ».

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "module.delete",
    "params": [
        "27",
        "28"
    ],
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "moduleids": [
            "27",
            "28"
        ]
    },
    "id": 1
}
```

Source

CModule::delete() dans *ui/include/classes/api/services/CModule.php*.

module.get

Description

`integer/array module.get(object parameters)`

Cette méthode permet de récupérer des modules selon les paramètres indiqués.

Note:
Cette méthode est uniquement disponible pour le type d'utilisateur *Super administrateur*. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres des rôles utilisateur. Consultez Rôles utilisateur pour plus d'informations.
 Si `$ZBX_FEATURE_FLAGS['modules_config_enabled']` est défini sur `false` dans le fichier de configuration de l'interface, l'utilisateur ne sera pas autorisé à appeler cette méthode.

Paramètres

`(object)` Paramètres définissant la sortie souhaitée.

La méthode prend en charge les paramètres suivants.

Paramètre	Type	Description
moduleids	ID/array	Retourner uniquement les modules avec les ID donnés.
sortfield	string/array	Trier le résultat selon les propriétés données. Valeurs possibles : `moduleid`, `relative_path`.
countOutput	boolean	Ces paramètres sont décrits dans le commentaire de référence.
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	

Paramètre	Type	Description
sortorder	string/array	
startSearch	boolean	

Valeurs de retour

`(integer/array)` Renvoie soit :

- un tableau d'objets ;
- le nombre d'objets récupérés, si le paramètre `countOutput` a été utilisé.

Exemples

Récupération d'un module par ID

Récupérez toutes les données des modules « 1 », « 2 » et « 25 ».

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "module.get",
    "params": {
        "output": "extend",
        "moduleids": [
            "1",
            "2",
            "25"
        ]
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "moduleid": "1",
            "id": "actionlog",
            "relative_path": "widgets/actionlog",
            "status": "1",
            "config": []
        },
        {
            "moduleid": "2",
            "id": "clock",
            "relative_path": "widgets/clock",
            "status": "1",
            "config": []
        },
        {
            "moduleid": "25",
            "id": "example",
            "relative_path": "modules/example_module",
            "status": "1",
            "config": []
        }
    ],
    "id": 1
}
```

Voir aussi

- Module
- Widget de tableau de bord

- Modules frontend

Source

CModule::get() dans *ui/include/classes/api/services/CModule.php*.

module.update

Description

`object module.update(object/array modules)`

Cette méthode vous permet de mettre à jour des modules existants.

> **Note:**
> Cette méthode est uniquement disponible pour le type d'utilisateur *Super administrateur*. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres des rôles utilisateur. Consultez la page Rôles utilisateur pour plus d'informations.
 Si `$ZBX_FEATURE_FLAGS['modules_config_enabled']` est défini sur `false` dans le fichier de configuration de l'interface, l'utilisateur ne sera pas autorisé à appeler cette méthode.

Paramètres

`(object/array)` Propriétés du module à mettre à jour.

La propriété `moduleid` doit être définie pour chaque module, toutes les autres propriétés sont facultatives. Seules les propriétés spécifiées seront mises à jour.

La méthode accepte les modules avec les propriétés de module standard.

Valeurs de retour

`(object)` Renvoie un objet contenant les identifiants des modules mis à jour dans la propriété `moduleids`.

Exemples

Désactivation d'un module

Désactivez le module « 25 ».

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "module.update",
    "params": {
        "moduleid": "25",
        "status": 0
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "moduleids": [
            "25"
        ]
    },
    "id": 1
}
```

Voir aussi

- Module
- Modules frontend

Source

CModule::update() dans *ui/include/classes/api/services/CModule.php*.

Modèle

Cette classe est conçue pour fonctionner avec des modèles.

Références d'objet :

- Modèle
- Balise de modèle

Méthodes disponibles :

- template.create - créer de nouveaux modèles
- template.delete - supprimer des modèles
- template.get - récupérer des modèles
- template.massadd - ajouter des objets associés aux modèles
- template.massremove - supprimer des objets associés des modèles
- template.update - mettre à jour des modèles

Objet modèle

Les objets suivants sont directement liés à l'API `template`.

Modèle

L'objet modèle possède les propriétés suivantes.

Propriété	Type	Description
templateid	ID	ID du modèle. Comportement de la propriété : - *lecture seule* - *obligatoire* pour les opérations de mise à jour
description	text	Description du modèle.
host	string	Nom technique du modèle. Comportement de la propriété : - *obligatoire* pour les opérations de création
name	string	Nom visible du modèle. Par défaut : valeur de la propriété `host`.
readme	text	Instructions de configuration spécifiques au modèle à afficher dans l'assistant hôte. Prend en charge le formatage Markdown.
uuid	string	Identifiant universel unique, utilisé pour lier les modèles importés à ceux qui existent déjà. Généré automatiquement s'il n'est pas fourni.
vendor_name	string	Nom du fournisseur du modèle. Pour les opérations de création, `vendor_name` et `vendor_version` doivent soit être définis tous les deux, soit être laissés vides. Pour les opérations de mise à jour, `vendor_version` peut être laissé vide s'il possède une valeur dans la base de données.
vendor_version	string	Version du fournisseur du modèle. Pour les opérations de création, `vendor_name` et `vendor_version` doivent soit être définis tous les deux, soit être laissés vides. Pour les opérations de mise à jour, `vendor_name` peut être laissé vide s'il possède une valeur dans la base de données.
wizard_ready	integer	Indique si le modèle est disponible pour la sélection dans l'assistant hôte. Valeurs possibles : 0 - *(par défaut)* Non disponible. 1 - Disponible.

Balise de modèle

L'objet de balise de modèle possède les propriétés suivantes.

Propriété	Type	Description
tag	string	Nom de la balise du modèle. Comportement de la propriété: - *obligatoire*
value	string	Valeur de la balise du modèle. Comportement de la propriété: - *lecture seule*.
object	integer	Type d'objet dont la balise a été héritée. Valeurs possibles : 0 - modèle. Comportement de la propriété: - *lecture seule*.
objectid	ID	ID de l'objet dont la balise a été héritée. Comportement de la propriété: - *lecture seule*.

template.create

Description

`object template.create(object/array templates)`

Cette méthode permet de créer de nouveaux modèles.

> **Note:**
> Cette méthode est disponible uniquement pour les types d'utilisateur *Admin* et *Super admin*. Les autorisations d'appel de cette méthode peuvent être révoquées dans les paramètres des rôles utilisateur. Consultez Rôles utilisateur pour plus d'informations.

Paramètres

`(object/array)` Modèles à créer.

En plus des propriétés standard du modèle, la méthode accepte les paramètres suivants.

Paramètre	Type	Description
groups	object/array	Groupes de modèles auxquels ajouter le modèle. Les groupes de modèles doivent avoir uniquement la propriété `groupid` définie. Comportement des paramètres : - *obligatoire*
tags	object/array	Balises de modèle.
templates	object/array	Modèles à lier au modèle. Les modèles doivent avoir uniquement la propriété `templateid` définie.
macros	object/array	Macros utilisateur à créer pour le modèle.

Valeurs retournées

`(object)` Retourne un objet contenant les identifiants des modèles créés sous la propriété `templateids`. L'ordre des ID retournés correspond à l'ordre des modèles transmis.

Exemples

Création d'un modèle

Créez un modèle avec des tags et liez deux modèles à ce modèle.

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "template.create",
    "params": {
        "host": "Linux template",
        "groups": {
            "groupid": 1
        },
        "templates": [
            {
                "templateid": "11115"
            },
            {
                "templateid": "11116"
            }
        ],
        "tags": [
            {
                "tag": "host-name",
                "value": "{HOST.NAME}"
            }
        ]
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "templateids": [
            "11117"
        ]
    },
    "id": 1
}
```

Source

CTemplate::create() dans *ui/include/classes/api/services/CTemplate.php*.

template.delete

Description

`object template.delete(array templateIds)`

Cette méthode vous permet de supprimer des modèles.

La suppression d'un modèle supprime également toutes ses entités (éléments, déclencheurs, graphiques, etc.) de tous les hôtes ou modèles auxquels il est lié. Pour conserver ces entités, commencez par dissocier le modèle des hôtes ou modèles concernés à l'aide des méthodes `host.update`, `host.massremove`, `template.update` ou `template.massremove`.

> **Note:**
> Cette méthode est disponible uniquement pour les types d'utilisateur *Admin* et *Super admin*. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres du rôle utilisateur. Consultez Rôles utilisateur pour plus d'informations.

Paramètres

`(array)` ID des modèles à supprimer.

Valeurs de retour

`(object)` Renvoie un objet contenant les ID des modèles supprimés dans la propriété `templateids`.

Exemples

Suppression de plusieurs modèles

Supprimez deux modèles.

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "template.delete",
    "params": [
        "13",
        "32"
    ],
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "templateids": [
            "13",
            "32"
        ]
    },
    "id": 1
}
```

Source

CTemplate::delete() dans *ui/include/classes/api/services/CTemplate.php*.

template.get

Description

`integer/array template.get(object parameters)`

Cette méthode permet de récupérer des modèles selon les paramètres fournis.

> **Note:**
> Cette méthode est disponible pour les utilisateurs de tout type. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres du rôle utilisateur. Consultez Rôles d'utilisateur pour plus d'informations.

Paramètres

`(object)` Paramètres définissant la sortie souhaitée.

La méthode prend en charge les paramètres suivants.

Paramètre	Type	Description
templateids	ID/array	Renvoie uniquement les modèles avec les ID de modèle donnés.
groupids	ID/array	Renvoie uniquement les modèles qui appartiennent aux groupes de modèles donnés.
parentTemplateids	ID/array	Renvoie uniquement les modèles auxquels le modèle donné est lié.
hostids	ID/array	Renvoie uniquement les modèles qui sont liés aux hôtes/modèles donnés.
graphids	ID/array	Renvoie uniquement les modèles qui contiennent les graphes donnés.
itemids	ID/array	Renvoie uniquement les modèles qui contiennent les éléments donnés.

Paramètre	Type	Description
triggerids	ID/array	Renvoie uniquement les modèles qui contiennent les déclencheurs donnés.
with_items	flag	Renvoie uniquement les modèles qui ont des éléments.
with_triggers	flag	Renvoie uniquement les modèles qui ont des déclencheurs.
with_graphs	flag	Renvoie uniquement les modèles qui ont des graphes.
with_httptests	flag	Renvoie uniquement les modèles qui ont des scénarios web.
evaltype	integer	Méthode d'évaluation des tags. Valeurs possibles : 0 - *(par défaut)* And/Or; 2 - Or.
tags	object/array	Renvoie uniquement les modèles avec les tags donnés. Format : `[{"tag": "<tag>", "value": "<value>", "operator": "<operator>"}, ...]`. Un tableau vide renvoie tous les modèles. Valeurs possibles de operator : 0 - *(par défaut)* Contient ; 1 - Égal à ; 2 - Ne contient pas ; 3 - N'est pas égal à ; 4 - Existe ; 5 - N'existe pas.
inheritedTags	boolean	Renvoie les modèles qui ont les `tags` donnés également dans les modèles liés. Valeurs possibles : `true` - les modèles liés doivent également avoir les tags donnés ; `false` - *(par défaut)* les tags des modèles liés sont ignorés.
selectInheritedTags	query	Renvoie une propriété `inheritedTags` avec les tags présents sur les modèles liés.
selectTags	query	Renvoie les tags du modèle dans la propriété `tags`.
selectDiscoveryRules	query	Renvoie une propriété `discoveryRules` avec les règles LLD du modèle. Prend en charge `count`.
selectHosts	query	Renvoie les hôtes qui sont liés au modèle dans la propriété `hosts`. Prend en charge `count`.
selectTemplateGroups	query	Renvoie les groupes de modèles auxquels le modèle appartient dans la propriété `templategroups`.
selectTemplates	query	Renvoie les modèles auxquels le modèle donné est lié dans la propriété `templates`. Prend en charge `count`.
selectParentTemplates	query	Renvoie les modèles qui sont liés au modèle donné dans la propriété `parentTemplates`. Prend en charge `count`.
selectHttpTests	query	Renvoie les scénarios web du modèle dans la propriété `httpTests`. Prend en charge `count`.
selectItems	query	Renvoie les éléments du modèle dans la propriété `items`. Prend en charge `count`.
selectTriggers	query	Renvoie les déclencheurs du modèle dans la propriété `triggers`. Prend en charge `count`.
selectGraphs	query	Renvoie les graphes du modèle dans la propriété `graphs`. Prend en charge `count`.

Paramètre	Type	Description
selectMacros	query	Renvoie les macros du modèle dans la propriété `macros`.
selectDashboards	query	Renvoie les tableaux de bord du modèle dans la propriété `dashboards`. Prend en charge `count`.
selectValueMaps	query	Renvoie une propriété `valuemaps` avec les correspondances de valeurs du modèle.
limitSelects	integer	Limite le nombre d'enregistrements renvoyés par les sous-sélections. S'applique aux sous-sélections suivantes : `selectTemplates` - les résultats seront triés par `name` ; `selectHosts` - triés par `host` ; `selectParentTemplates` - triés par `host` ; `selectItems` - triés par `name` ; `selectDiscoveryRules` - triés par `name` ; `selectTriggers` - triés par `description` ; `selectGraphs` - triés par `name` ; `selectDashboards` - triés par `name`.
sortfield	string/array	Trie le résultat selon les propriétés données. Valeurs possibles : `hostid`, `host`, `name`, `status`.
countOutput	boolean	Ces paramètres sont décrits dans le commentaire de référence.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	
selectDiscoveries	query	Renvoie les découvertes de bas niveau du modèle dans la propriété `discoveries`. Prend en charge `count`. Cette requête est **obsolète** ; veuillez utiliser `selectDiscoveryRules` à la place.

Valeurs de retour

`(integer/array)` Renvoie soit :

- un tableau d'objets ;
- le nombre d'objets récupérés, si le paramètre `countOutput` a été utilisé.

Exemples

Récupération des modèles par nom

Récupérez toutes les données concernant deux modèles nommés "Linux by Zabbix agent" et "Windows by Zabbix agent".

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "template.get",
    "params": {
        "output": "extend",
        "filter": {
            "host": [
                "Linux by Zabbix agent",
```

```
                "Windows by Zabbix agent"
            ]
        }
    },
    "id": 1
}
```

Réponse :


```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "proxyid": "0",
            "host": "Linux by Zabbix agent",
            "status": "3",
            "ipmi_authtype": "-1",
            "ipmi_privilege": "2",
            "ipmi_username": "",
            "ipmi_password": "",
            "maintenanceid": "0",
            "maintenance_status": "0",
            "maintenance_type": "0",
            "maintenance_from": "0",
            "name": "Linux by Zabbix agent",
            "flags": "0",
            "templateid": "10001",
            "description": "Il s'agit d'un modèle Linux officiel. Il nécessite Zabbix agent 8.0 ou une ver
            "tls_connect": "1",
            "tls_accept": "1",
            "tls_issuer": "",
            "tls_subject": "",
            "tls_psk_identity": "",
            "tls_psk": "",
            "custom_interfaces": "0",
            "uuid": "f8f7908280354f2abeed07dc788c3747",
            "vendor_name": "Zabbix",
            "vendor_version": "8.0-2",
            "proxy_groupid": "0",
            "monitored_by": "0",
            "wizard_ready": "1",
            "readme": "## Vue d'ensemble\r\n\r\nIl s'agit d'un modèle Linux officiel. Il nécessite Zabbix
        },
        {
            "proxyid": "0",
            "host": "Windows by Zabbix agent",
            "status": "3",
            "ipmi_authtype": "-1",
            "ipmi_privilege": "2",
            "ipmi_username": "",
            "ipmi_password": "",
            "maintenanceid": "0",
            "maintenance_status": "0",
            "maintenance_type": "0",
            "maintenance_from": "0",
            "name": "Windows by Zabbix agent",
            "flags": "0",
            "templateid": "10081",
            "description": "Il s'agit d'un modèle Windows officiel. Il nécessite Zabbix agent 8.0 ou une v
            "tls_connect": "1",
            "tls_accept": "1",
            "tls_issuer": "",
            "tls_subject": "",
```

```
            "tls_psk_identity": "",
            "tls_psk": "",
            "custom_interfaces": "0",
            "uuid": "13b06904a6bf41cbb795e3193d896340",
            "vendor_name": "Zabbix",
            "vendor_version": "8.0-2",
            "proxy_groupid": "0",
            "monitored_by": "0",
            "wizard_ready": "1",
            "readme": "## Vue d'ensemble\r\n\r\nIl s'agit d'un modèle Windows officiel. Il nécessite Zabbi
        }
    ],
    "id": 1
}
```

Récupération des groupes de modèles

Récupérez les groupes de modèles dont le modèle « Linux by Zabbix agent » est membre.

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "template.get",
    "params": {
        "output": ["hostid"],
        "selectTemplateGroups": "extend",
        "filter": {
            "host": [
                "Linux by Zabbix agent"
            ]
        }
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "templateid": "10001",
            "templategroups": [
                {
                    "groupid": "10",
                    "name": "Templates/Operating systems",
                    "uuid": "846977d1dfed4968bc5f8bdb363285bc"
                }
            ]
        }
    ],
    "id": 1
}
```

Récupération des hôtes par modèle

Récupérez les hôtes auxquels le modèle « 10001 » (*Linux by Zabbix agent*) est lié.

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "template.get",
    "params": {
        "output": "templateid",
        "templateids": "10001",
```

```
        "selectHosts": ["hostid", "name"]
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "templateid": "10001",
            "hosts": [
                {
                    "hostid": "10084",
                    "name": "Zabbix server"
                },
                {
                    "hostid": "10603",
                    "name": "Host 1"
                },
                {
                    "hostid": "10604",
                    "name": "Host 2"
                }
            ]
        }
    ],
    "id": 1
}
```

Recherche par balises de modèle

Récupérez les modèles ayant la balise ”host-name” égale à ”{HOST.NAME}”.

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "template.get",
    "params": {
        "output": ["hostid"],
        "selectTags": "extend",
        "evaltype": 0,
        "tags": [
            {
                "tag": "host-name",
                "value": "{HOST.NAME}",
                "operator": 1
            }
        ]
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "templateid": "10402",
            "tags": [
                {
                    "tag": "host-name",
                    "value": "{HOST.NAME}"
```

```
                }
            ]
        }
    ],
    "id": 1
}
```

Voir aussi

- Groupe de modèles
- Modèle
- Macro utilisateur
- Interface d'hôte

Source

CTemplate::get() dans *ui/include/classes/api/services/CTemplate.php*.

template.massadd

Description

`object template.massadd(object parameters)`

Cette méthode permet d'ajouter simultanément plusieurs objets associés aux modèles donnés.

> **Note:**
> Cette méthode est disponible uniquement pour les types d'utilisateur *Admin* et *Super admin*. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres des rôles utilisateur. Voir Rôles utilisateur pour plus d'informations.

Paramètres

`(object)` Paramètres contenant les ID des modèles à mettre à jour et les objets à ajouter aux modèles donnés.

La méthode accepte les paramètres suivants.

Paramètre	Type	Description
templates	object/array	Modèles à mettre à jour. Les modèles doivent avoir uniquement la propriété `templateid` définie. Comportement des paramètres: - *obligatoire*
groups	object/array	Groupes de modèles à ajouter aux modèles donnés. Les groupes de modèles doivent avoir uniquement la propriété `groupid` définie.
macros	object/array	Macros utilisateur à créer pour les modèles donnés.
templates_link	object/array	Modèles à lier aux modèles donnés. Les modèles doivent avoir uniquement la propriété `templateid` définie.

Valeurs retournées

`(object)` Retourne un objet contenant les identifiants des modèles mis à jour sous la propriété `templateids`.

Exemples

Lier un groupe à des modèles

Ajouter le groupe de modèles « 2 » à deux modèles.

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "template.massadd",
    "params": {
        "templates": [
            {
                "templateid": "10085"
            },
            {
                "templateid": "10086"
            }
        ],
        "groups": [
            {
                "groupid": "2"
            }
        ]
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "templateids": [
            "10085",
            "10086"
        ]
    },
    "id": 1
}
```

Lier deux modèles à un modèle

Liez les modèles « 10106 » et « 10104 » au modèle « 10073 ».

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "template.massadd",
    "params": {
        "templates": [
            {
                "templateid": "10073"
            }
        ],
        "templates_link": [
            {
                "templateid": "10106"
            },
            {
                "templateid": "10104"
            }
        ]
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
```

```
        "templateids": [
            "10073"
        ]
    },
    "id": 1
}
```

Voir aussi

- template.update
- Hôte
- Groupe de modèles
- Macro utilisateur

Source

CTemplate::massAdd() dans *ui/include/classes/api/services/CTemplate.php*.

template.massremove

Description

`object template.massremove(object parameters)`

Cette méthode vous permet de supprimer des objets associés de plusieurs modèles.

Note:
Cette méthode est disponible uniquement pour les types d'utilisateur *Admin* et *Super admin*. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres des rôles utilisateur. Consultez Rôles utilisateur pour plus d'informations.

Paramètres

`(object)` Paramètres contenant les ID des modèles à mettre à jour et les objets qui doivent être supprimés.

Paramètre	Type	Description
templateids	ID/array	ID des modèles à mettre à jour. Comportement des paramètres : - *obligatoire*
groupids	ID/array	ID des groupes de modèles desquels supprimer les modèles donnés.
macros	string/array	ID des macros utilisateur à supprimer des modèles donnés.
templateids_clear	ID/array	ID des modèles à délier et effacer des modèles donnés (en amont).
templateids_link	ID/array	ID des modèles à délier des modèles donnés (en amont).

Valeurs retournées

`(object)` Retourne un objet contenant les identifiants des modèles mis à jour sous la propriété `templateids`.

Exemples

Suppression de modèles d'un groupe

Supprimez deux modèles du groupe « 2 ».

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "template.massremove",
    "params": {
        "templateids": [
            "10085",
            "10086"
        ],
        "groupids": "2"
```

```
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "templateids": [
            "10085",
            "10086"
        ]
    },
    "id": 1
}
```

Dissociation de modèles d'un hôte

Dissociez les modèles « 10106 » et « 10104 » du modèle « 10085 ».

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "template.massremove",
    "params": {
        "templateids": "10085",
        "templateids_link": [
            "10106",
            "10104"
        ]
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "templateids": [
            "10085"
        ]
    },
    "id": 1
}
```

Voir aussi

- template.update
- Macro utilisateur

Source

CTemplate::massRemove() dans *ui/include/classes/api/services/CTemplate.php*.

template.update

Description

`object template.update(object/array templates)`

Cette méthode vous permet de mettre à jour des modèles existants.

Note:
Cette méthode est disponible uniquement pour les types d'utilisateur *Admin* et *Super admin*. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres des rôles utilisateur. Consultez Rôles utilisateur pour plus d'informations.

Paramètres

`(object/array)` Propriétés du modèle à mettre à jour.

La propriété `templateid` doit être définie pour chaque modèle, toutes les autres propriétés sont facultatives. Seules les propriétés indiquées seront mises à jour, toutes les autres resteront inchangées.

En plus des propriétés de modèle standard, la méthode accepte les paramètres suivants.

Paramètre	Type	Description
groups	object/array	Groupes de modèles pour remplacer les groupes de modèles actuels auxquels les modèles appartiennent. Les groupes de modèles doivent avoir uniquement la propriété `groupid` définie.
tags	object/array	Balises de modèle pour remplacer les balises de modèle actuelles.
macros	object/array	Macros utilisateur pour remplacer les macros utilisateur actuelles sur les modèles donnés.
templates	object/array	Modèles pour remplacer les modèles actuellement liés. Les modèles qui ne sont pas transmis seront uniquement dissociés. Les modèles doivent avoir uniquement la propriété `templateid` définie.
templates_clear	object/array	Modèles à dissocier et supprimer des modèles donnés. Les modèles doivent avoir uniquement la propriété `templateid` définie.

Valeurs retournées

`(object)` Retourne un objet contenant les identifiants des modèles mis à jour sous la propriété `templateids`.

Exemples

Modification des propriétés standard du modèle

Modifiez le nom technique du modèle en "Linux by Zabbix agent Custom", le nom visible en "My template", et mettez à jour la description du modèle.

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "template.update",
    "params": {
        "templateid": "10086",
        "host": "Linux by Zabbix agent Custom",
        "name": "My template",
        "description": "This is a custom Linux template."
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "templateids": [
            "10086"
        ]
```

```
    },
    "id": 1
}
```

Mise à jour des groupes de modèles

Remplace tous les groupes de modèles du modèle donné par un autre.

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "template.update",
    "params": {
        "templateid": "10086",
        "groups": [
            {
                "groupid": "24"
            }
        ]
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "templateids": [
            "10086"
        ]
    },
    "id": 1
}
```

Mise à jour de plusieurs groupes de modèles

Remplacez tous les groupes de modèles de plusieurs modèles par d'autres groupes différents.

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "template.update",
    "params": [
        {
            "templateid": "10086",
            "groups": [
                {
                    "groupid": "24"
                }
            ]
        },
        {
            "templateid": "10087",
            "groups": [
                {
                    "groupid": "1"
                },
                {
                    "groupid": "12"
                }
            ]
        },
        {
            "templateid": "10088",
```

```
            "groups": [
                {
                    "groupid": "1"
                },
                {
                    "groupid": "12"
                }
            ]
        }
    ],
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "templateids": [
            "10086",
            "10087",
            "10088"
        ]
    },
    "id": 1
}
```

Mise à jour des tags de modèle

Remplacez tous les tags du modèle par un autre.

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "template.update",
    "params": {
        "templateid": "10086",
        "tags": [
            {
                "tag": "host-name",
                "value": "{HOST.NAME}"
            }
        ]
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "templateids": [
            "10086"
        ]
    },
    "id": 1
}
```

Mise à jour des macros de modèle

Remplacez toutes les macros du modèle par une autre.

Requête:

```
{
    "jsonrpc": "2.0",
```

```
    "method": "template.update",
    "params": {
        "templateid": "10086",
        "macros": [
            {
                "macro": "{$MY_MACRO}",
                "value": "new_value"
            }
        ]
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "templateids": [
            "10086"
        ]
    },
    "id": 1
}
```

Mise à jour de plusieurs macros de modèle

Remplacez toutes les macros utilisateur par la macro utilisateur donnée sur plusieurs modèles.

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "template.update",
    "params": [
            {
                "templateid": "10087",
                "macros": [
                    {
                        "macro": "{$AGENT.TIMEOUT}",
                        "value": "5m",
                        "description": "Délai après lequel l'agent est considéré comme indisponible. Fonct
                    }
                ]
            },
            {
                "templateid": "10088",
                "macros": [
                    {
                        "macro": "{$AGENT.TIMEOUT}",
                        "value": "5m",
                        "description": "Délai après lequel l'agent est considéré comme indisponible. Fonct
                    }
                ]
            }
        ],
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "templateids": [
            "10087",
```

```
            "10088"
        ]
    },
    "id": 1
}
```

Mise à jour des modèles liés à un modèle

Dissociez (sans effacer) tous les modèles du modèle donné et liez-lui un autre modèle.

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "template.update",
    "params": {
        "templateid": "10086",
        "templates": [
            {
                "templateid": "10001"
            }
        ]
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "templateids": [
            "10086"
        ]
    },
    "id": 1
}
```

Suppression de plusieurs modèles liés à des modèles

Déliez et supprimez un modèle lié spécifique des modèles donnés.

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "template.update",
    "params": [
        {
            "templateid": "10087",
            "templates_clear": [
                {
                    "templateid": "10001"
                }
            ]
        },
        {
            "templateid": "10088",
            "templates_clear": [
                {
                    "templateid": "10001"
                }
            ]
        }
    ],
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "templateids": [
            "10087",
            "10088"
        ]
    },
    "id": 1
}
```

Source

CTemplate::update() dans *ui/include/classes/api/services/CTemplate.php*.

Métrique APM

Cette classe est conçue pour fonctionner avec les métriques collectées pour la surveillance des performances des applications (APM).

Références des objets :

- Métrique APM
- Exemplaire

Méthodes disponibles :

- apmmetric.get - récupérer les points de données de la métrique

Objet de métrique APM

Les objets suivants sont directement liés à l'API `apmmetric`.

> **Note:**
> Chaque propriété, à l'exception de `type`, correspond à une colonne des tables de métriques (`otel_metrics_gauge`, `otel_metrics_sum`, `otel_metrics_histogram` et `otel_metrics_exponential_histogram`) de la source de données de télémétrie. La plupart des propriétés correspondent à la colonne portant le même nom (par exemple, la propriété `metric_name` correspond à la colonne `MetricName`). Lorsque les noms diffèrent, les colonnes de la source de données sont indiquées dans les tableaux ci-dessous. Pour obtenir la liste des colonnes, consultez Colonnes de données de télémétrie prises en charge.

Métrique APM

L'objet de métrique APM représente un point de données d'une métrique (la valeur, ou les valeurs agrégées, d'une mesure à un moment donné).

L'objet de métrique APM possède les propriétés suivantes :

Property	Type	Description
type	integer	Type du point de données. Valeurs possibles : 0 - Jauge ; 1 - Somme ; 2 - Histogramme ; 3 - Histogramme exponentiel. Le type est déterminé par la table qui contient le point de données : `otel_metrics_gauge`, `otel_metrics_sum`, `otel_metrics_histogram` ou `otel_metrics_exponential_histogram`. Comportement de la propriété : - *lecture seule*
resource_attributes	object	Attributs de ressource clé-valeur du point de données (par exemple, `host.name: eabf8c2a5ddc`). Comportement de la propriété : - *lecture seule*
resource_schema_url	string	URL, le cas échéant, du schéma OpenTelemetry des attributs de la ressource. Comportement de la propriété : - *lecture seule*
scope_name	string	Nom du périmètre d'instrumentation ayant produit le point de données. Comportement de la propriété : - *lecture seule*
scope_version	string	Version, le cas échéant, du périmètre d'instrumentation. Comportement de la propriété : - *lecture seule*
scope_attributes	object	Attributs clé-valeur du périmètre d'instrumentation. Comportement de la propriété : - *lecture seule*
scope_schema_url	string	URL, le cas échéant, du schéma OpenTelemetry du périmètre d'instrumentation et de ses attributs. Comportement de la propriété : - *lecture seule*
service_name	string	Nom du service ayant produit le point de données (par exemple, `zabbix-frontend`), extrait de l'attribut de ressource OpenTelemetry `service.name`. Comportement de la propriété : - *lecture seule*
metric_name	string	Nom de la métrique à laquelle appartient le point de données (par exemple, `http.server.request.duration`). Comportement de la propriété : - *lecture seule*
metric_description	string	Description, le cas échéant, de la métrique. Comportement de la propriété : - *lecture seule*
metric_unit	string	Unité, le cas échéant, des valeurs de la métrique (par exemple, `s` ou `By` ; voir UCUM). Comportement de la propriété : - *lecture seule*

Property	Type	Description
attributes	object	Attributs du point de données clé-valeur (par exemple, `http.request.method: POST`). Comportement de la propriété : - *lecture seule*
start_time_unix	integer	Heure Unix (en nanosecondes) à laquelle a commencé la période de mesure du point de données. Comportement de la propriété : - *lecture seule*
time_unix	integer	Heure Unix (en nanosecondes) du point de données. Pour les points de données de type *Somme*, *Histogramme* et *Histogramme exponentiel*, il s'agit de la fin de la période de mesure. Comportement de la propriété : - *lecture seule*
value	float	Valeur du point de données. S'applique aux points de données de type *Jauge* et *Somme*. Valeur par défaut : 0. Comportement de la propriété : - *lecture seule*
flags	integer	Masque de bits des indicateurs du point de données. Si le bit de poids faible est défini (valeur 1), aucune valeur n'a été enregistrée pour le point de données. Comportement de la propriété : - *lecture seule*
exemplars	array	Tableau d'objets exemplaire, c'est-à-dire de mesures individuelles conservées comme exemples, avec la trace et le span pendant lesquels elles ont été enregistrées. Comportement de la propriété : - *lecture seule*
aggregation_temporality	integer	Manière dont les valeurs du point de données sont agrégées au fil du temps. S'applique aux points de données de type *Somme*, *Histogramme* et *Histogramme exponentiel*. Valeurs possibles : 0 - *(par défaut)* Non spécifiée ; 1 - Delta (agrégée sur la période depuis le point de données précédent) ; 2 - Cumulative (agrégée sur la période depuis `start_time_unix`). Comportement de la propriété : - *lecture seule*
is_monotonic	boolean	Indique si la somme peut uniquement augmenter ou rester constante au fil du temps. S'applique aux points de données de type *Somme*. Valeur par défaut : `false`. Comportement de la propriété : - *lecture seule*

Property	Type	Description
count	integer	Nombre de mesures dans le point de données. S'applique aux points de données de type *Histogramme* et *Histogramme exponentiel*. Valeur par défaut : 0. Comportement de la propriété : - *lecture seule*
sum	float	Somme de toutes les mesures du point de données. S'applique aux points de données de type *Histogramme* et *Histogramme exponentiel*. Valeur par défaut : 0. Comportement de la propriété : - *lecture seule*
bucket_counts	array	Tableau d'entiers indiquant le nombre de mesures dans chaque compartiment de l'histogramme. Les compartiments sont définis par `explicit_bounds`. S'applique aux points de données de type *Histogramme*. Valeur par défaut : `[]`. Comportement de la propriété : - *lecture seule*
explicit_bounds	array	Tableau de nombres flottants indiquant les limites entre les compartiments de l'histogramme. S'applique aux points de données de type *Histogramme*. Valeur par défaut : `[]`. Comportement de la propriété : - *lecture seule*
min	float	Plus petite des mesures du point de données. S'applique aux points de données de type *Histogramme* et *Histogramme exponentiel*. Valeur par défaut : 0. Comportement de la propriété : - *lecture seule*
max	float	Plus grande des mesures du point de données. S'applique aux points de données de type *Histogramme* et *Histogramme exponentiel*. Valeur par défaut : 0. Comportement de la propriété : - *lecture seule*
scale	integer	Échelle de l'histogramme exponentiel, qui détermine les limites et la résolution des compartiments. Une échelle plus élevée produit des compartiments plus étroits et une résolution plus fine ; une échelle plus faible produit des compartiments plus larges et une résolution plus grossière. S'applique aux points de données de type *Histogramme exponentiel*. Valeur par défaut : 0. Comportement de la propriété : - *lecture seule*

Property	Type	Description
zero_count	integer	Nombre de mesures dans le compartiment zéro de l'histogramme exponentiel. S'applique aux points de données de type *Histogramme exponentiel*. Valeur par défaut : 0. Comportement de la propriété : - *lecture seule*
positive_bucket_counts	array	Tableau d'entiers indiquant le nombre de mesures positives dans chaque compartiment de l'histogramme exponentiel. Les limites des compartiments sont déterminées par `scale` ; l'index du premier compartiment est défini dans `positive_offset`. S'applique aux points de données de type *Histogramme exponentiel*. Valeur par défaut : `[]`. Comportement de la propriété : - *lecture seule*
positive_offset	integer	Index du premier compartiment dans `positive_bucket_counts`. S'applique aux points de données de type *Histogramme exponentiel*. Valeur par défaut : 0. Comportement de la propriété : - *lecture seule*
negative_bucket_counts	array	Tableau d'entiers indiquant le nombre de mesures négatives dans chaque compartiment de l'histogramme exponentiel. Les limites des compartiments sont déterminées par `scale` ; l'index du premier compartiment est défini dans `negative_offset`. S'applique aux points de données de type *Histogramme exponentiel*. Valeur par défaut : `[]`. Comportement de la propriété : - *lecture seule*
negative_offset	integer	Index du premier compartiment dans `negative_bucket_counts`. S'applique aux points de données de type *Histogramme exponentiel*. Valeur par défaut : 0. Comportement de la propriété : - *lecture seule*

Exemple

L'objet exemple possède les propriétés suivantes :

Propriété	Type	Description
traceid	string	ID, le cas échéant, de la trace pendant laquelle la mesure a été enregistrée. Colonne de la source de données : `Exemplars.TraceId`.
spanid	string	ID, le cas échéant, du span pendant lequel la mesure a été enregistrée. Colonne de la source de données : `Exemplars.SpanId`.
value	float	Valeur de la mesure. Colonne de la source de données : `Exemplars.Value`.
time_unix	integer	Heure Unix (en nanosecondes) à laquelle la mesure a été enregistrée. Colonne de la source de données : `Exemplars.TimeUnix`.

Propriété	Type	Description
filtered_attributes	object	Attributs clé-valeur de la mesure qui ne sont pas inclus dans les `attributes` du point de données. Colonne de la source de données : `Exemplars.FilteredAttributes`.

apmmetric.get

Description

`integer/array apmmetric.get(object parameters)`

Cette méthode vous permet de récupérer des points de données de métriques selon les paramètres indiqués.

> **Note:**
> Cette méthode est disponible pour les utilisateurs de tout type. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres du rôle utilisateur. Consultez la page Rôles utilisateur pour plus d'informations.

Paramètres

`(object)` Paramètres définissant le résultat souhaité.

La méthode prend en charge les paramètres suivants.

Paramètre	Type	Description
time_from	timestamp	Retourne uniquement les points de données dont l'heure est égale ou postérieure à l'heure indiquée. Comportement du paramètre : - *requis*
time_till	timestamp	Retourne uniquement les points de données dont l'heure est antérieure à l'heure indiquée. Comportement du paramètre : - *requis*
types	integer/array	Retourne uniquement les points de données des types indiqués. Consultez la page de l'objet de métrique APM pour obtenir la liste des types de métriques pris en charge.
with_flags_on	integer	Retourne uniquement les points de données pour lesquels tous les indicateurs indiqués sont définis. Pour spécifier plusieurs indicateurs, utilisez la somme de leurs valeurs.
with_flags_off	integer	Retourne uniquement les points de données pour lesquels aucun des indicateurs indiqués n'est défini. Pour spécifier plusieurs indicateurs, utilisez la somme de leurs valeurs.
resource_attributes	object/array	Retourne uniquement les points de données possédant les attributs de ressource indiqués. Format : `[{"key": "<key>", "value": "<value>", "operator": "<operator>"}, ...]`. Un tableau vide retourne tous les points de données. Valeurs possibles de l'opérateur : 0 - *(par défaut)* Contient ; 1 - Est égal à ; 2 - Ne contient pas ; 3 - N'est pas égal à ; 4 - Existe ; 5 - N'existe pas.

Paramètre	Type	Description
resource_attributes_evaltype	integer	Méthode d'évaluation des conditions dans `resource_attributes`. Valeurs possibles : 0 - *(par défaut)* Et/Ou ; 2 - Ou.
scope_attributes	object/array	Retourne uniquement les points de données possédant les attributs de portée indiqués. Format : `[{"key": "<key>", "value": "<value>", "operator": "<operator>"}, ...]`. Un tableau vide retourne tous les points de données. Valeurs possibles de l'opérateur : 0 - *(par défaut)* Contient ; 1 - Est égal à ; 2 - Ne contient pas ; 3 - N'est pas égal à ; 4 - Existe ; 5 - N'existe pas.
scope_attributes_evaltype	integer	Méthode d'évaluation des conditions dans `scope_attributes`. Valeurs possibles : 0 - *(par défaut)* Et/Ou ; 2 - Ou.
attributes	object/array	Retourne uniquement les points de données possédant les attributs indiqués. Format : `[{"key": "<key>", "value": "<value>", "operator": "<operator>"}, ...]`. Un tableau vide retourne tous les points de données. Valeurs possibles de l'opérateur : 0 - *(par défaut)* Contient ; 1 - Est égal à ; 2 - Ne contient pas ; 3 - N'est pas égal à ; 4 - Existe ; 5 - N'existe pas.
attributes_evaltype	integer	Méthode d'évaluation des conditions dans `attributes`. Valeurs possibles : 0 - *(par défaut)* Et/Ou ; 2 - Ou.
filter	object	Retourne uniquement les résultats correspondant exactement au filtre indiqué. Accepte un objet dont les clés sont des noms de propriétés et dont les valeurs sont soit une valeur unique, soit un tableau de valeurs auxquelles effectuer la correspondance. Propriétés prises en charge : `resource_schema_url`, `scope_schema_url`, `scope_name`, `scope_version`, `service_name`, `metric_name`, `metric_unit`, `flags`.

Paramètre	Type	Description
search	object	Retourne les résultats correspondant au modèle indiqué (sans distinction entre majuscules et minuscules). Accepte un objet dont les clés sont des noms de propriétés et dont les valeurs sont des chaînes à rechercher. Si aucune option supplémentaire n'est indiquée, une recherche `LIKE "%…%"` est effectuée. Propriétés prises en charge : `resource_schema_url`, `scope_schema_url`, `scope_name`, `scope_version`, `service_name`, `metric_name`, `metric_description`, `metric_unit`.
sortfield	string/array	Trie le résultat selon les propriétés indiquées. Valeurs possibles : `type`, `resource_schema_url`, `scope_schema_url`, `scope_name`, `scope_version`, `service_name`, `metric_name`, `metric_unit`, `start_time_unix`, `time_unix`.
offset	integer	Nombre d'enregistrements à ignorer avant de retourner le résultat.
countOutput	boolean	Ces paramètres sont décrits dans le commentaire de référence.
excludeSearch	boolean	
limit	integer	
output	query	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Valeurs de retour

`(integer/array)` Renvoie soit :

- un tableau d'objets ;
- le nombre d'objets récupérés, si le paramètre `countOutput` a été utilisé.

Exemples

Récupération des points de données d'un histogramme

Récupérer le dernier point de données de la métrique d'histogramme `http.server.request.duration` du service `zabbix-frontend`.

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "apmmetric.get",
    "params": {
        "output": ["metric_name", "metric_unit", "time_unix", "count", "sum", "min", "max", "explicit_boun
        "time_from": 1790848800,
        "time_till": 1790852400,
        "types": 2,
        "filter": {
            "service_name": "zabbix-frontend",
            "metric_name": "http.server.request.duration"
        },
        "sortfield": "time_unix",
        "sortorder": "DESC",
        "limit": 1
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "metric_name": "http.server.request.duration",
            "metric_unit": "s",
            "time_unix": "1790852390000000000",
            "count": "1842",
            "sum": "412.773",
            "min": "0.0061",
            "max": "1.098",
            "explicit_bounds": ["0.005", "0.05", "0.1", "0.25", "0.5", "1"],
            "bucket_counts": ["0", "211", "904", "512", "167", "41", "7"],
            "exemplars": [
                {
                    "traceid": "dee625f33cb0994e86c5cd4640e3b09c",
                    "spanid": "3f8b2c6d1e9a7054",
                    "value": "1.098",
                    "time_unix": "1790850513439147623",
                    "filtered_attributes": {}
                }
            ]
        }
    ],
    "id": 1
}
```


Voir aussi

- Trace APM
- Span APM
- Journal APM

Source

CApmMetric::get() dans *ui/include/classes/api/services/CApmMetric.php*.

Nettoyage interne

Cette classe est conçue pour fonctionner avec le nettoyage interne.

Références d'objet :

- Housekeeping

Méthodes disponibles :

- housekeeping.get - récupérer le nettoyage interne
- housekeeping.update - mettre à jour le nettoyage interne

Objet de nettoyage interne

Les objets suivants sont directement liés à l'API `housekeeping`.

Nettoyage

L'objet des paramètres possède les propriétés suivantes.

Propriété	Type	Description
hk_events_mode	integer	Activer le nettoyage interne des événements et des alertes. Valeurs possibles : 0 - Désactiver ; 1 - *(par défaut)* Activer.

Propriété	Type	Description
hk_events_trigger	string	Période de conservation des données des déclencheurs. Accepte un nombre de secondes (par ex. 1, 60, 300) ou une valeur avec un suffixe temporel (par ex. 30s, 1m, 2h, 1d). Valeur par défaut : 365d.
hk_events_service	string	Période de conservation des données des services. Accepte un nombre de secondes (par ex. 1, 60, 300) ou une valeur avec un suffixe temporel (par ex. 30s, 1m, 2h, 1d). Valeur par défaut : 1d.
hk_events_internal	string	Période de conservation des données internes. Accepte un nombre de secondes (par ex. 1, 60, 300) ou une valeur avec un suffixe temporel (par ex. 30s, 1m, 2h, 1d). Valeur par défaut : 1d.
hk_events_discovery	string	Période de conservation des données de découverte réseau. Accepte un nombre de secondes (par ex. 1, 60, 300) ou une valeur avec un suffixe temporel (par ex. 30s, 1m, 2h, 1d). Valeur par défaut : 1d.
hk_events_autoreg	string	Période de conservation des données d'enregistrement automatique. Accepte un nombre de secondes (par ex. 1, 60, 300) ou une valeur avec un suffixe temporel (par ex. 30s, 1m, 2h, 1d). Valeur par défaut : 1d.
hk_services_mode	integer	Activer le nettoyage interne des services. Valeurs possibles : 0 - Désactiver ; 1 - *(par défaut)* Activer.
hk_services	string	Période de conservation des données des services. Accepte un nombre de secondes (par ex. 1, 60, 300) ou une valeur avec un suffixe temporel (par ex. 30s, 1m, 2h, 1d). Valeur par défaut : 365d.
hk_audit_mode	integer	Activer le nettoyage interne des audits. Valeurs possibles : 0 - Désactiver ; 1 - *(par défaut)* Activer.
hk_audit	string	Période de conservation des données d'audit. Accepte un nombre de secondes (par ex. 1, 60, 300) ou une valeur avec un suffixe temporel (par ex. 30s, 1m, 2h, 1d). Valeur par défaut : 31d.
hk_sessions_mode	integer	Activer le nettoyage interne des sessions. Valeurs possibles : 0 - Désactiver ; 1 - *(par défaut)* Activer.
hk_sessions	string	Période de conservation des données des sessions. Accepte un nombre de secondes (par ex. 1, 60, 300) ou une valeur avec un suffixe temporel (par ex. 30s, 1m, 2h, 1d). Valeur par défaut : 31d.
hk_history_mode	integer	Activer le nettoyage interne de l'historique. Valeurs possibles : 0 - Désactiver ; 1 - *(par défaut)* Activer.

Propriété	Type	Description
hk_history_global	integer	Remplacer la période d'historique des éléments. Valeurs possibles : 0 - Ne pas remplacer ; 1 - *(par défaut)* Remplacer.
hk_history	string	Période de conservation des données d'historique. Accepte un nombre de secondes (par ex. 1, 60, 300) ou une valeur avec un suffixe temporel (par ex. 30s, 1m, 2h, 1d). Valeur par défaut : 31d.
hk_trends_mode	integer	Activer le nettoyage interne des tendances. Valeurs possibles : 0 - Désactiver ; 1 - *(par défaut)* Activer.
hk_trends_global	integer	Remplacer la période des tendances des éléments. Valeurs possibles : 0 - Ne pas remplacer ; 1 - *(par défaut)* Remplacer.
hk_trends	string	Période de conservation des données de tendances. Accepte un nombre de secondes (par ex. 1, 60, 300) ou une valeur avec un suffixe temporel (par ex. 30s, 1m, 2h, 1d). Valeur par défaut : 365d.
db_extension	string	Indicateur de configuration de l'extension DB. Si cet indicateur est défini sur « timescaledb », le serveur modifie son comportement pour le nettoyage et la suppression des éléments. Comportement de la propriété : - *lecture seule*
compression_availability	integer	Indique si la compression des données est prise en charge par la base de données (ou son extension). Valeurs possibles : 0 - Non disponible ; 1 - Disponible. Comportement de la propriété : - *lecture seule*
compression_status	integer	Activer la compression TimescaleDB pour l'historique et les tendances. Valeurs possibles : 0 - *(par défaut)* Désactivée ; 1 - Activée.
compress_older	string	Compresser les enregistrements d'historique et de tendances antérieurs à la période spécifiée. Accepte un nombre de secondes (par ex. 1, 60, 300) ou une valeur avec un suffixe temporel (par ex. 30s, 1m, 2h, 1d). Valeur par défaut : 7d.

housekeeping.get

Description

`object housekeeping.get(object parameters)`

La méthode permet de récupérer un objet de maintenance selon les paramètres fournis.

Note:
Cette méthode est disponible pour les utilisateurs de tout type. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres des rôles utilisateur. Consultez Rôles utilisateur pour plus d'informations.

Paramètres

`(object)` Paramètres définissant la sortie souhaitée.

La méthode ne prend en charge qu'un seul paramètre.

Paramètre	Type	Description
output	query	Ce paramètre est décrit dans le commentaire de référence.

Valeurs retournées

`(object)` Renvoie l'objet de procédure de nettoyage.

Exemples

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "housekeeping.get",
    "params": {
        "output": "extend"
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "hk_events_mode": "1",
        "hk_events_trigger": "365d",
        "hk_events_service": "1d",
        "hk_events_internal": "1d",
        "hk_events_discovery": "1d",
        "hk_events_autoreg": "1d",
        "hk_services_mode": "1",
        "hk_services": "365d",
        "hk_audit_mode": "1",
        "hk_audit": "31d",
        "hk_sessions_mode": "1",
        "hk_sessions": "365d",
        "hk_history_mode": "1",
        "hk_history_global": "0",
        "hk_history": "31d",
        "hk_trends_mode": "1",
        "hk_trends_global": "0",
        "hk_trends": "365d",
        "db_extension": "",
        "compression_status": "0",
        "compress_older": "7d"
    },
    "id": 1
}
```

Source

CHousekeeping ::get() dans *ui/include/classes/api/services/CHousekeeping.php*.

housekeeping.update

Description

`object housekeeping.update(object housekeeping)`

Cette méthode vous permet de mettre à jour des paramètres de maintenance existants.

> **Note:**
> Cette méthode est disponible uniquement pour le type d'utilisateur *Super admin*. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres des rôles utilisateur. Voir User roles pour plus d'informations.

Paramètres

`(object)` Propriétés de nettoyage interne à mettre à jour.

Valeurs de retour

`(array)` Renvoie un tableau contenant les noms des paramètres mis à jour.

Exemples

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "housekeeping.update",
    "params": {
        "hk_events_mode": "1",
        "hk_events_trigger": "200d",
        "hk_events_internal": "2d",
        "hk_events_discovery": "2d"
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": [
        "hk_events_mode",
        "hk_events_trigger",
        "hk_events_internal",
        "hk_events_discovery"
    ],
    "id": 1
}
```

Source

CHousekeeping::update() dans *ui/include/classes/api/services/CHousekeeping.php*.

Nœud de haute disponibilité

Cette classe est conçue pour fonctionner avec des nœuds de serveur qui font partie d'un cluster de haute disponibilité ou d'une instance de serveur autonome.

Références d'objet :

- Nœud de haute disponibilité

Méthodes disponibles :

- hanode.get - récupérer les nœuds

Objet de nœud de haute disponibilité

L'objet suivant est lié à l'exploitation d'un cluster de haute disponibilité de serveurs Zabbix.

Noeud de haute disponibilité

> **Note:**
> Les noeuds sont créés par le serveur Zabbix et ne peuvent pas être modifiés via l'API.

L'objet Noeud de haute disponibilité possède les propriétés suivantes.

Property	Type	Description
ha_nodeid	ID	ID du noeud.
name	string	Nom attribué au noeud, à l'aide de l'entrée de configuration HANodeName de `zabbix_server.conf`. Vide pour un serveur exécuté en mode autonome.
address	string	Adresse IP ou nom DNS à partir duquel le noeud se connecte.
port	integer	Port sur lequel le noeud s'exécute.
lastaccess	integer	Heure du heartbeat, c'est-à-dire l'heure de la dernière mise à jour provenant du noeud. Horodatage UTC.
status	integer	État du noeud. Valeurs possibles : 0 - veille; 1 - arrêté manuellement; 2 - indisponible; 3 - actif.

hanode.get

Description

`integer/array hanode.get(object parameters)`

La méthode permet de récupérer une liste de nœuds de cluster de haute disponibilité selon les paramètres fournis.

> **Note:**
> Cette méthode est disponible uniquement pour les types d'utilisateurs *Super admin*. Consultez Rôles d'utilisateur pour plus d'informations.

Paramètres

`(object)` Paramètres définissant la sortie souhaitée.

La méthode prend en charge les paramètres suivants.

Paramètre	Type	Description
ha_nodeids	ID/array	Retourner uniquement les nœuds avec les ID de nœud donnés.
filter	object	Retourner uniquement les résultats qui correspondent exactement au filtre donné. Accepte un objet, où les clés sont des noms de propriété, et les valeurs sont soit une valeur unique, soit un tableau de valeurs à comparer.
sortfield	string/array	Prend en charge les propriétés suivantes : `name`, `address`, `status`. Trier le résultat selon les propriétés données. Valeurs possibles : `name`, `lastaccess`, `status`.
countOutput	flag	Ces paramètres sont décrits dans le commentaire de référence.
limit	integer	
output	query	
preservekeys	boolean	

Paramètre	Type	Description
sortorder	string/array	

Valeurs retournées

`(integer/array)` Retourne soit :

- un tableau d'objets ;
- le nombre d'objets récupérés, si le paramètre `countOutput` a été utilisé.

Exemples

Obtenir une liste de nœuds triés par statut

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "hanode.get",
    "params": {
        "preservekeys": true,
        "sortfield": "status",
        "sortorder": "DESC"
    },
    "id": 1
}
```

Réponse :

```
{
  "jsonrpc": "2.0",
  "result": {
    "ckuo7i1nw000h0sajj3l3hh8u": {
      "ha_nodeid": "ckuo7i1nw000h0sajj3l3hh8u",
      "name": "node-active",
      "address": "192.168.1.13",
      "port": "10051",
      "lastaccess": "1635335704",
      "status": "3"
    },
    "ckuo7i1nw000e0sajwfttc1mp": {
      "ha_nodeid": "ckuo7i1nw000e0sajwfttc1mp",
      "name": "node6",
      "address": "192.168.1.10",
      "port": "10053",
      "lastaccess": "1635332902",
      "status": "2"
    },
    "ckuo7i1nv000c0sajz85xcrtt": {
      "ha_nodeid": "ckuo7i1nv000c0sajz85xcrtt",
      "name": "node4",
      "address": "192.168.1.8",
      "port": "10052",
      "lastaccess": "1635334214",
      "status": "1"
    },
    "ckuo7i1nv000a0saj1fcdkeu4": {
      "ha_nodeid": "ckuo7i1nv000a0saj1fcdkeu4",
      "name": "node2",
      "address": "192.168.1.6",
      "port": "10051",
      "lastaccess": "1635335705",
      "status": "0"
    }
  },
```

```
  "id": 1
}
```

Obtenir une liste de nœuds spécifiques par leurs ID

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "hanode.get",
    "params": {
        "ha_nodeids": ["ckuo7i1nw000e0sajwfttc1mp", "ckuo7i1nv000c0sajz85xcrtt"]
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "ha_nodeid": "ckuo7i1nv000c0sajz85xcrtt",
            "name": "node4",
            "address": "192.168.1.8",
            "port": "10052",
            "lastaccess": "1635334214",
            "status": "1"
        },
        {
            "ha_nodeid": "ckuo7i1nw000e0sajwfttc1mp",
            "name": "node6",
            "address": "192.168.1.10",
            "port": "10053",
            "lastaccess": "1635332902",
            "status": "2"
        }
    ],
    "id": 1
}
```

Obtenir une liste des nœuds arrêtés

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "hanode.get",
    "params": {
        "output": ["ha_nodeid", "address", "port"],
        "filter": {
            "status": 1
        }
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "ha_nodeid": "ckuo7i1nw000g0sajjsjre7e3",
            "address": "192.168.1.12",
            "port": "10051"
        },
```

```
        {
            "ha_nodeid": "ckuo7i1nv000c0sajz85xcrtt",
            "address": "192.168.1.8",
            "port": "10052"
        },
        {
            "ha_nodeid": "ckuo7i1nv000d0sajd95y1b6x",
            "address": "192.168.1.9",
            "port": "10053"
        }
    ],
    "id": 1
}
```

Obtenir le nombre de nœuds de secours

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "hanode.get",
    "params": {
        "countOutput": true,
        "filter": {
            "status": 0
        }
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": "3",
    "id": 1
}
```

Vérifier l'état des nœuds à des adresses IP spécifiques

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "hanode.get",
    "params": {
        "output": ["name", "status"],
        "filter": {
            "address": ["192.168.1.7", "192.168.1.13"]
        }
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "name": "node3",
            "status": "0"
        },
        {
            "name": "node-active",
            "status": "3"
        }
```

```
    ],
    "id": 1
}
```

Source

CHaNode::get() dans *ui/include/classes/api/services/CHaNode.php*.

Paramètres

Cette classe est conçue pour fonctionner avec les paramètres d'administration courants.

Références d'objet :

- Settings

Méthodes disponibles :

- settings.get - récupérer les paramètres
- settings.update - mettre à jour les paramètres

Objet settings

Les objets suivants sont directement liés à l'API `settings`.

Paramètres

L'objet des paramètres possède les propriétés suivantes.

Propriété	Type	Description
default_lang	string	Langue du système par défaut. Par défaut : `en_US`.
default_timezone	string	Fuseau horaire du système par défaut. Par défaut : `system` - valeur par défaut du système. Pour obtenir la liste complète des fuseaux horaires pris en charge, consultez la documentation PHP.
default_theme	string	Thème par défaut. Valeurs possibles : `blue-theme` - *(par défaut)* Bleu ; `blue-classic-theme` - Bleu (classique) ; `dark-theme` - Sombre ; `dark-blue-theme` - Bleu sombre ; `dark-classic-theme` - Sombre (classique) ; `hc-light` - Clair à contraste élevé ; `hc-dark` - Sombre à contraste élevé.
search_limit	integer	Limite pour les résultats de recherche et de filtrage. Par défaut : 1000.
max_overview_table_size	integer	Nombre maximal de colonnes et de lignes dans les widgets de tableau de bord Aperçu des données et Aperçu des déclencheurs. Par défaut : 50.
max_in_table	integer	Nombre maximal d'éléments à afficher dans une cellule de tableau. Par défaut : 50.
server_check_interval	integer	Afficher un avertissement si le serveur Zabbix est arrêté. Valeurs possibles : 0 - Ne pas afficher d'avertissement ; 10 - *(par défaut)* Afficher un avertissement.

Propriété	Type	Description
work_period	string	Heures de travail. Par défaut : 1-5,09:00-18:00.
show_technical_errors	integer	Afficher les erreurs techniques (PHP/SQL) aux utilisateurs qui ne sont pas des super administrateurs et aux utilisateurs qui ne font pas partie de groupes d'utilisateurs pour lesquels le mode débogage est activé. Valeurs possibles : 0 - *(par défaut)* Ne pas afficher les erreurs techniques ; 1 - Afficher les erreurs techniques.
history_period	string	Période maximale d'affichage des données d'historique dans les widgets de tableau de bord Dernières données, Web et Aperçu des données. Accepte un nombre de secondes (par exemple, 1, 60, 300) ou une valeur avec un suffixe temporel (par exemple, 30s, 1m, 2h, 1d). Par défaut : 24h.
period_default	string	Période par défaut du filtre temporel. Accepte un nombre de secondes (par exemple, 1, 60, 300) ou une valeur avec un suffixe temporel (par exemple, 30s, 1m, 2h, 1d, 1M, 1y), avec prise en charge des mois et des années. Par défaut : 1h.
max_period	string	Période maximale du filtre temporel. Accepte un nombre de secondes (par exemple, 1, 60, 300) ou une valeur avec un suffixe temporel (par exemple, 30s, 1m, 2h, 1d, 1M, 1y), avec prise en charge des mois et des années. Par défaut : 2y.
severity_color_0	string	Couleur de la sévérité « Non classé » sous forme de code couleur hexadécimal. Par défaut : 97AAB3.
severity_color_1	string	Couleur de la sévérité « Information » sous forme de code couleur hexadécimal. Par défaut : 7499FF.
severity_color_2	string	Couleur de la sévérité « Avertissement » sous forme de code couleur hexadécimal. Par défaut : FFC859.
severity_color_3	string	Couleur de la sévérité « Moyenne » sous forme de code couleur hexadécimal. Par défaut : FFA059.
severity_color_4	string	Couleur de la sévérité « Élevée » sous forme de code couleur hexadécimal. Par défaut : E97659.
severity_color_5	string	Couleur de la sévérité « Désastre » sous forme de code couleur hexadécimal. Par défaut : E45959.
severity_name_0	string	Nom de la sévérité « Non classé ». Par défaut : Non classé.
severity_name_1	string	Nom de la sévérité « Information ». Par défaut : Information.
severity_name_2	string	Nom de la sévérité « Avertissement ». Par défaut : Avertissement.

Propriété	Type	Description
severity_name_3	string	Nom de la sévérité « Moyenne ». Par défaut : Moyenne.
severity_name_4	string	Nom de la sévérité « Élevée ». Par défaut : Élevée.
severity_name_5	string	Nom de la sévérité « Désastre ». Par défaut : Désastre.
custom_color	integer	Utiliser des couleurs personnalisées pour l'état des événements. Valeurs possibles : 0 - *(par défaut)* Ne pas utiliser de couleurs personnalisées pour l'état des événements ; 1 - Utiliser des couleurs personnalisées pour l'état des événements.
ok_period	string	Afficher la période des déclencheurs OK. Accepte un nombre de secondes (par exemple, 1, 60, 300) ou une valeur avec un suffixe temporel (par exemple, 30s, 1m, 2h). Par défaut : 5m.
blink_period	string	Période de clignotement lors d'un changement d'état du déclencheur. Accepte un nombre de secondes (par exemple, 1, 60, 300) ou une valeur avec un suffixe temporel (par exemple, 30s, 1m, 2h). Par défaut : 2m.
problem_unack_color	string	Couleur des événements PROBLEM non confirmés sous forme de code couleur hexadécimal. Par défaut : CC0000.
problem_ack_color	string	Couleur des événements PROBLEM confirmés sous forme de code couleur hexadécimal. Par défaut : CC0000.
ok_unack_color	string	Couleur des événements RESOLVED non confirmés sous forme de code couleur hexadécimal. Par défaut : 009900.
ok_ack_color	string	Couleur des événements RESOLVED confirmés sous forme de code couleur hexadécimal. Par défaut : 009900.
problem_unack_style	integer	Clignotement des événements PROBLEM non confirmés. Valeurs possibles : 0 - Ne pas afficher de clignotement ; 1 - *(par défaut)* Afficher un clignotement.
problem_ack_style	integer	Clignotement des événements PROBLEM confirmés. Valeurs possibles : 0 - Ne pas afficher de clignotement ; 1 - *(par défaut)* Afficher un clignotement.
ok_unack_style	integer	Clignotement des événements RESOLVED non confirmés. Valeurs possibles : 0 - Ne pas afficher de clignotement ; 1 - *(par défaut)* Afficher un clignotement.
ok_ack_style	integer	Clignotement des événements RESOLVED confirmés. Valeurs possibles : 0 - Ne pas afficher de clignotement ; 1 - *(par défaut)* Afficher un clignotement.

Propriété	Type	Description
url	string	URL de l'interface.
discovery_groupid	ID	ID du groupe d'hôtes dans lequel les hôtes découverts seront automatiquement placés.
default_inventory_mode	integer	Mode d'inventaire des hôtes par défaut. Valeurs possibles : -1 - *(par défaut)* Désactivé ; 0 - Manuel ; 1 - Automatique.
alert_usrgrpid	ID	ID du groupe d'utilisateurs auquel le message d'alarme d'indisponibilité de la base de données sera envoyé. Si la valeur « 0 » est définie, le message d'alarme ne sera pas envoyé.
snmptrap_logging	integer	Consigner les traps SNMP non correspondants. Valeurs possibles : 0 - Ne pas consigner les traps SNMP non correspondants ; 1 - *(par défaut)* Consigner les traps SNMP non correspondants.
login_attempts	integer	Nombre de tentatives de connexion échouées après lequel le formulaire de connexion sera bloqué. Par défaut : 5.
login_block	string	Intervalle pendant lequel le formulaire de connexion sera bloqué si le nombre de tentatives de connexion échouées dépasse la valeur définie dans le champ login_attempts. Accepte un nombre de secondes (par exemple, 1, 60) ou une valeur avec un suffixe temporel (par exemple, 30s, 1m). Par défaut : 30s.
validate_uri_schemes	integer	Valider les schémas URI. Valeurs possibles : 0 - Ne pas valider ; 1 - *(par défaut)* Valider.
uri_valid_schemes	string	Schémas URI valides. Par défaut : http,https,ftp,file,mailto,tel,ssh.
x_frame_options	string	En-tête HTTP X-Frame-Options. Par défaut : SAMEORIGIN.
iframe_sandboxing_enabled	integer	Utiliser le bac à sable des iframe. Valeurs possibles : 0 - Ne pas utiliser ; 1 - *(par défaut)* Utiliser.
iframe_sandboxing_exceptions	string	Exceptions du bac à sable des iframe.
connect_timeout	string	Délai d'expiration de la connexion avec le serveur Zabbix. Plage de valeurs possibles : 1-30s. Par défaut : 3s. Comportement de la propriété : - *required*
socket_timeout	string	Délai d'expiration réseau par défaut. Plage de valeurs possibles : 1-300s. Par défaut : 3s. Comportement de la propriété : - *required*

Propriété	Type	Description
media_type_test_timeout	string	Délai d'expiration réseau pour le test du type de média. Plage de valeurs possibles : 1-300s. Par défaut : 65s. Comportement de la propriété : - *required*
item_test_timeout	string	Délai d'expiration réseau pour les tests des éléments. Plage de valeurs possibles : 1-600s. Par défaut : 60s. Comportement de la propriété : - *required*
script_timeout	string	Délai d'expiration réseau pour l'exécution des scripts. Plage de valeurs possibles : 1-300s. Par défaut : 60s. Comportement de la propriété : - *required*
report_test_timeout	string	Délai d'expiration réseau pour le test d'un rapport planifié. Plage de valeurs possibles : 1-300s. Par défaut : 60s. Comportement de la propriété : - *required*
device_link_timeout	string	Délai d'expiration pour l'ajout d'un appareil mobile à utiliser avec l'application mobile Zabbix ; s'applique à la communication entre l'API Zabbix et l'adaptateur bridge. Plage de valeurs possibles : 1-300s. Par défaut : 60s. Comportement de la propriété : - *required*
auditlog_enabled	integer	Indique s'il faut activer la journalisation d'audit. Valeurs possibles : 0 - Désactiver ; 1 - *(par défaut)* Activer.
auditlog_mode	integer	Indique s'il faut activer la journalisation d'audit des activités de découverte de bas niveau, de découverte réseau et d'enregistrement automatique effectuées par le serveur (utilisateur System). Valeurs possibles : 0 - Désactiver ; 1 - *(par défaut)* Activer.
ha_failover_delay	string	Délai de basculement en secondes. Par défaut : 1m. Comportement de la propriété : - *read-only*

Propriété	Type	Description
geomaps_tile_provider	string	Fournisseur de tuiles de carte géographique. Valeurs possibles : `OpenStreetMap.Mapnik` - *(par défaut)* OpenStreetMap Mapnik ; `OpenTopoMap` - OpenTopoMap ; `Stamen.TonerLite` - Stamen Toner Lite ; `Stamen.Terrain` - Stamen Terrain ; `USGS.USTopo` - USGS US Topo ; `USGS.USImagery` - USGS US Imagery. Prend en charge une chaîne vide pour spécifier des valeurs personnalisées de `geomaps_tile_url`, `geomaps_max_zoom` et `geomaps_attribution`.
geomaps_tile_url	string	URL des tuiles de carte géographique. Comportement de la propriété : - *supported* si `geomaps_tile_provider` est défini sur une chaîne vide
geomaps_max_zoom	integer	Niveau de zoom maximal de la carte géographique. Valeurs possibles : 0-30. Comportement de la propriété : - *supported* si `geomaps_tile_provider` est défini sur une chaîne vide
geomaps_attribution	string	Texte d'attribution de la carte géographique. Comportement de la propriété : - *supported* si `geomaps_tile_provider` est défini sur une chaîne vide
vault_provider	integer	Fournisseur Vault. Valeurs possibles : 0 - *(par défaut)* HashiCorp Vault ; 1 - CyberArk Vault.
proxy_secrets_provider	integer	Résoudre les valeurs des macros secrètes par : Valeurs possibles : 0 - *(par défaut)* le serveur uniquement ; 1 - le serveur et les proxy indépendamment.
timeout_zabbix_agent	string	Ne pas consacrer plus de `timeout_*` secondes au traitement. Accepte un nombre de secondes (par exemple, `1`, `60`, `300`) ou une valeur avec un suffixe temporel (par exemple, `30s`, `1m`). Accepte également les macros utilisateur. Plage de valeurs possibles : 1-600s. Par défaut : 3s. Valeur par défaut pour `timeout_browser` : 60s. Comportement de la propriété : - *required*
timeout_simple_check		
timeout_snmp_agent		
timeout_external_check		
timeout_db_monitor		
timeout_http_agent		
timeout_ssh_agent		
timeout_telnet_agent		
timeout_script		
timeout_browser		

Propriété	Type	Description
timeout_telemetry_query		
apm_global_db	object	Configuration de la source de données globale.

Source de données globale

L'objet de source de données globale possède les propriétés suivantes :

Propriété	Type	Description
status	integer	État de la source de données globale. Valeurs possibles : 0 - *(par défaut)* non configurée ; 1 - configurée.
url	string	URL utilisée pour se connecter à la source de données. Comportement de la propriété : - *obligatoire* si l'utilisateur modifie `status` sur 1 (configurée).
authentication_type	integer	Type d'authentification. Valeurs possibles : 0 - *(par défaut)* utilisation d'un nom d'utilisateur et d'un mot de passe ; 1 - aucune authentification utilisée. Comportement de la propriété : - *obligatoire* si l'utilisateur modifie `status` sur 1 (configurée).
username	string	Nom d'utilisateur utilisé pour se connecter à la source de données. Comportement de la propriété : - *pris en charge* si `status` est défini sur 1 (configurée) et `authentication_type` sur 0 (nom d'utilisateur et mot de passe). - *obligatoire* si l'utilisateur modifie `authentication_type` sur 0 (nom d'utilisateur et mot de passe).
password	string	Mot de passe utilisé pour se connecter à la source de données. La valeur de ce paramètre peut être vide. Comportement de la propriété : - *pris en charge* si `status` est défini sur 1 (configurée) et `authentication_type` sur 0 (nom d'utilisateur et mot de passe).
db	string	Nom de la base de données. La valeur de ce paramètre peut être vide. Comportement de la propriété : - *pris en charge* si `status` est défini sur 1 (configurée).
ssl_verify_peer	integer	Vérifier le certificat du serveur. Valeurs possibles : 0 - *(par défaut)* désactivé ; 1 - activé. Comportement de la propriété : - *pris en charge* si `status` est défini sur 1 (configurée) et si `url` utilise le schéma `https://`.

Propriété	Type	Description
ssl_verify_host	integer	Vérifier l'hôte. Valeurs possibles : 0 - désactivé ; 1 - *(par défaut)* activé. Comportement de la propriété : - *pris en charge* si `status` est défini sur 1 (configurée) et si `url` utilise le schéma `https://`.

settings.get

Description

`object settings.get(object parameters)`

La méthode permet de récupérer l'objet des paramètres selon les paramètres fournis.

> **Note:**
> Cette méthode est disponible pour les utilisateurs de tout type. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres du rôle utilisateur. Consultez Rôles d'utilisateur pour plus d'informations.

Paramètres

`(object)` Paramètres définissant la sortie souhaitée.

La méthode ne prend en charge qu'un seul paramètre.

Paramètre	Type	Description
output	query	Ce paramètre est décrit dans le commentaire de référence.

Valeurs retournées

`(object)` Renvoie l'objet paramètres.

Exemples

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "settings.get",
    "params": {
        "output": "extend"
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "alert_usrgrpid": "7",
        "apm_global_db": {
            "status": 1,
            "url": "https://localhost:8443",
            "authentication_type": 0,
            "username": "zabbix",
            "db": "",
            "ssl_verify_peer": 1,
            "ssl_verify_host": 1
        },
```

```
        "auditlog_enabled": "1",
        "auditlog_mode": "1",
        "blink_period": "2m",
        "connect_timeout": "3s",
        "custom_color": "0",
        "default_inventory_mode": "-1",
        "default_lang": "en_US",
        "default_theme": "blue-theme",
        "default_timezone": "system",
        "device_link_timeout": "60s",
        "discovery_groupid": "5",
        "geomaps_attribution": "",
        "geomaps_max_zoom": "0",
        "geomaps_tile_provider": "OpenStreetMap.Mapnik",
        "geomaps_tile_url": "",
        "ha_failover_delay": "1m",
        "history_period": "24h",
        "iframe_sandboxing_enabled": "1",
        "iframe_sandboxing_exceptions": "",
        "item_test_timeout": "60s",
        "login_attempts": "5",
        "login_block": "30s",
        "max_in_table": "50",
        "max_overview_table_size": "50",
        "max_period": "2y",
        "media_type_test_timeout": "65s",
        "ok_ack_color": "009900",
        "ok_ack_style": "1",
        "ok_period": "5m",
        "ok_unack_color": "009900",
        "ok_unack_style": "1",
        "period_default": "1h",
        "problem_ack_color": "CC0000",
        "problem_ack_style": "1",
        "problem_unack_color": "CC0000",
        "problem_unack_style": "1",
        "proxy_secrets_provider": "0",
        "report_test_timeout": "60s",
        "script_timeout": "60s",
        "search_limit": "1000",
        "server_check_interval": "10",
        "serverid": "019f4707-1a3e-70e6-aca2-286a2b696358",
        "severity_color_0": "97AAB3",
        "severity_color_1": "7499FF",
        "severity_color_2": "FFC859",
        "severity_color_3": "FFA059",
        "severity_color_4": "E97659",
        "severity_color_5": "E45959",
        "severity_name_0": "Non classé",
        "severity_name_1": "Information",
        "severity_name_2": "Avertissement",
        "severity_name_3": "Moyenne",
        "severity_name_4": "Élevée",
        "severity_name_5": "Désastre",
        "show_technical_errors": "0",
        "snmptrap_logging": "1",
        "socket_timeout": "3s",
        "timeout_browser": "60s",
        "timeout_db_monitor": "3s",
        "timeout_external_check": "3s",
        "timeout_http_agent": "3s",
        "timeout_script": "3s",
```

```
        "timeout_simple_check": "3s",
        "timeout_snmp_agent": "3s",
        "timeout_ssh_agent": "3s",
        "timeout_telemetry_query": "3s",
        "timeout_telnet_agent": "3s",
        "timeout_zabbix_agent": "3s",
        "uri_valid_schemes": "http,https,ftp,file,mailto,tel,ssh",
        "url": "",
        "validate_uri_schemes": "1",
        "vault_provider": "0",
        "work_period": "1-5,09:00-18:00",
        "x_frame_options": "SAMEORIGIN"
    },
    "id": 1
}
```

Source

CSettings::get() dans *ui/include/classes/api/services/CSettings.php*.

settings.update

Description

`object settings.update(object settings)`

Cette méthode vous permet de mettre à jour des paramètres communs existants.

> **Note:**
> Cette méthode est disponible uniquement pour le type d'utilisateur *Super admin*. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres du rôle utilisateur. Consultez User roles pour plus d'informations.

Paramètres

`(object)` Propriétés des paramètres à mettre à jour.

Valeurs de retour

`(array)` Renvoie un tableau contenant les noms des paramètres mis à jour.

Exemples

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "settings.update",
    "params": {
        "login_attempts": "1",
        "login_block": "1m"
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": [
        "login_attempts",
        "login_block"
    ],
    "id": 1
}
```

Source

CSettings::update() dans *ui/include/classes/api/services/CSettings.php*.

Problème

Cette classe est conçue pour fonctionner avec les problèmes.

Références d'objet :

- Problem
  - URL du type de média
- Balise de problème

Méthodes disponibles :

- problem.get - récupérer les problèmes

Objet problème

Les objets suivants sont directement liés à l'API `problem`.

Problème

Note:
Les problèmes sont créés par le serveur Zabbix et ne peuvent pas être modifiés via l'API.

L'objet problème possède les propriétés suivantes.

Property	Type	Description
eventid	ID	ID de l'événement de problème.
source	integer	Type de l'événement de problème. Valeurs possibles : 0 - événement créé par un déclencheur ; 3 - événement interne ; 4 - événement créé lors de la mise à jour de l'état du service.
object	integer	Type d'objet lié à l'événement de problème. Valeurs possibles si `source` est défini sur « événement créé par un déclencheur » : 0 - déclencheur. Valeurs possibles si `source` est défini sur « événement interne » : 0 - déclencheur ; 4 - élément ; 5 - règle LLD. Valeurs possibles si `source` est défini sur « événement créé lors de la mise à jour de l'état du service » : 6 - service.
objectid	ID	ID de l'objet lié.
clock	timestamp	Heure de création de l'événement de problème.
ns	integer	Nanosecondes au moment de la création de l'événement de problème.
r_eventid	ID	ID de l'événement de rétablissement.
r_clock	timestamp	Heure de création de l'événement de rétablissement.
r_ns	integer	Nanosecondes au moment de la création de l'événement de rétablissement.
cause_eventid	ID	ID de l'événement cause.
correlationid	ID	ID de la règle de corrélation si cet événement a été rétabli par une règle de corrélation globale.
userid	ID	ID de l'utilisateur qui a fermé le problème (si le problème a été fermé manuellement).
name	string	Nom du problème résolu.

Property	Type	Description
acknowledged	integer	État d'acquittement du problème. Valeurs possibles : 0 - non acquitté ; 1 - acquitté.
severity	integer	Gravité actuelle du problème. Valeurs possibles : 0 - non classé ; 1 - information ; 2 - avertissement ; 3 - moyen ; 4 - élevé ; 5 - désastre.
suppressed	integer	Indique si le problème est supprimé. Valeurs possibles : 0 - le problème est à l'état normal ; 1 - le problème est supprimé.
opdata	string	Données opérationnelles avec macros développées.
urls	array	URLs de type de média actives.

URL du type de média

L'objet URL du type de média possède les propriétés suivantes.

Propriété	Type	Description
name	string	Nom de l'URL définie pour le type de média.
url	string	Valeur de l'URL définie pour le type de média.

Les résultats contiendront des entrées uniquement pour les types de média actifs avec une entrée de menu d'événement activée. Les macros utilisées dans les propriétés seront développées, mais si l'une des propriétés contient une macro non développée, les deux propriétés seront exclues des résultats. Pour les macros prises en charge, voir *Macros prises en charge*.

Tag de problème

L'objet de tag de problème possède les propriétés suivantes.

Propriété	Type	Description
tag	string	Nom du tag de problème.
value	string	Valeur du tag de problème.

problem.get

Description

`integer/array problem.get(object parameters)`

La méthode vous permet de récupérer les problèmes selon les paramètres fournis.

Cette méthode sert à récupérer les problèmes non résolus. Il est également possible, si cela est spécifié, de récupérer en plus les problèmes récemment résolus. La période qui détermine ce que signifie "récemment" est définie dans *Administration* > *Général*. Les problèmes résolus avant cette période ne sont pas conservés dans la table des problèmes. Pour récupérer des problèmes résolus plus loin dans le passé, utilisez la méthode `event.get`.

Attention:
Cette méthode peut renvoyer des problèmes d'une entité supprimée si ces problèmes n'ont pas encore été supprimés par le housekeeper.

Note:
Cette méthode est disponible pour les utilisateurs de tout type. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres du rôle utilisateur. Consultez Rôles utilisateur pour plus d'informations.

Paramètres

`(object)` Paramètres définissant la sortie souhaitée.

La méthode prend en charge les paramètres suivants.

Parameter	Type	Description
eventids	ID/array	Ne retourner que les problèmes avec les IDs donnés.
groupids	ID/array	Ne retourner que les problèmes créés par des objets appartenant aux groupes d'hôtes donnés.
hostids	ID/array	Ne retourner que les problèmes créés par des objets appartenant aux hôtes donnés.
objectids	ID/array	Ne retourner que les problèmes créés par les objets donnés.
source	integer	Ne retourner que les problèmes du type donné. Consultez la page de l'objet d'événement de problème pour obtenir la liste des types d'événements pris en charge. Par défaut: 0 - problème créé par un déclencheur.
object	integer	Ne retourner que les problèmes créés par des objets du type donné. Consultez la page de l'objet d'événement de problème pour obtenir la liste des types d'objets pris en charge. Par défaut: 0 - déclencheur.
acknowledged	boolean	Si défini sur `true`, ne retourner que les problèmes reconnus.
action	integer	Ne retourner que les problèmes pour lesquels les actions de mise à jour d'événement données ont été effectuées. Pour plusieurs actions, utilisez une somme de toutes les valeurs de bitmap acceptables comme masque de bits (par exemple, 34 pour reconnaître et supprimer l'événement).
action_userids	ID/array	Ne retourner que les problèmes avec les IDs donnés des utilisateurs qui ont effectué les actions de mise à jour de l'événement de problème.
suppressed	boolean	Si défini sur `true`, ne retourner que les problèmes supprimés.
symptom	boolean	Si défini sur `true`, ne retourner que les événements de problème symptomatiques.
severities	integer/array	Ne retourner que les problèmes avec les sévérités d'événement données. S'applique uniquement si object est déclencheur.
evaltype	integer	Méthode d'évaluation des balises. Valeurs possibles: 0 - *(par défaut)* Et/Ou; 2 - Ou.
tags	array	Ne retourner que les problèmes avec les balises données. Format: `[{"tag": "<tag>", "value": "<value>", "operator": "<operator>"}, ...]`. Un tableau vide retourne tous les problèmes. Valeurs possibles de operator: 0 - *(par défaut)* Contient; 1 - Égal à; 2 - Ne contient pas; 3 - N'est pas égal à; 4 - Existe; 5 - N'existe pas.
recent	boolean	Si défini sur `true`, retourner les problèmes récents, y compris ceux résolus dans la période `ok_period` et tous les problèmes non résolus. Si défini sur `false`, ne retourner que les problèmes non résolus.

Parameter	Type	Description
eventid_from	string	Ne retourner que les problèmes avec des IDs supérieurs ou égaux à l'ID donné.
eventid_till	string	Ne retourner que les problèmes avec des IDs inférieurs ou égaux à l'ID donné.
time_from	timestamp	Ne retourner que les problèmes qui ont été créés après ou à l'heure donnée.
time_till	timestamp	Ne retourner que les problèmes qui ont été créés avant ou à l'heure donnée.
selectAcknowledges	query	Retourner une propriété `acknowledges` avec les mises à jour du problème. Les mises à jour du problème sont triées en ordre chronologique inverse. L'objet de mise à jour du problème possède les propriétés suivantes: `acknowledgeid` - `(ID)` ID de la mise à jour; `userid` - `(ID)` ID de l'utilisateur qui a mis à jour l'événement; `eventid` - `(ID)` ID de l'événement mis à jour; `clock` - `(timestamp)` heure à laquelle l'événement a été mis à jour; `message` - `(string)` texte du message; `action` - `(integer)` type d'action de mise à jour (voir `event.acknowledge`); `old_severity` - `(integer)` sévérité de l'événement avant cette action de mise à jour; `new_severity` - `(integer)` sévérité de l'événement après cette action de mise à jour; `suppress_until` - `(timestamp)` heure jusqu'à laquelle l'événement sera supprimé; `taskid` - `(ID)` ID de la tâche si l'événement actuel est en cours de changement de rang; `maintenanceid` - `(ID)` ID de la maintenance. Prend en charge `count`.
selectTags	query	Retourner une propriété `tags` avec les balises du problème. Format de sortie: `[{"tag": "<tag>", "value": "<value>"}, ...]`.
selectSuppressionData	query	Retourner une propriété `suppression_data` avec la liste des maintenances actives et des suppressions manuelles: `maintenanceid` - `(ID)` ID de la maintenance; `userid` - `(ID)` ID de l'utilisateur qui a supprimé le problème; `suppress_until` - `(integer)` heure jusqu'à laquelle le problème est supprimé.
filter	object	Ne retourner que les résultats qui correspondent exactement au filtre donné. Accepte un objet, où les clés sont des noms de propriétés et les valeurs sont soit une seule valeur, soit un tableau de valeurs à comparer. Ne prend pas en charge les propriétés de type de données data type `text`.
sortfield	string/array	Trier le résultat selon les propriétés données. Valeurs possibles: `eventid`.
countOutput	boolean	Ces paramètres sont décrits dans la commentaire de référence.
editable	boolean	
excludeSearch	boolean	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Valeurs de retour

`(integer/array)` Renvoie soit :

- un tableau d'objets ;
- le nombre d'objets récupérés, si le paramètre `countOutput` a été utilisé.

Exemples

Récupération des événements de problème du déclencheur

Récupérer les événements récents du déclencheur "15112."

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "problem.get",
    "params": {
        "output": "extend",
        "selectAcknowledges": "extend",
        "selectTags": "extend",
        "selectSuppressionData": "extend",
        "objectids": "15112",
        "recent": true,
        "sortfield": ["eventid"],
        "sortorder": "DESC"
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "eventid": "1245463",
            "source": "0",
            "object": "0",
            "objectid": "15112",
            "clock": "1472457242",
            "ns": "209442442",
            "r_eventid": "1245468",
            "r_clock": "1472457285",
            "r_ns": "125644870",
            "correlationid": "0",
            "userid": "1",
            "name": "Zabbix agent on localhost is unreachable for 5 minutes",
            "acknowledged": "1",
            "severity": "3",
            "cause_eventid": "0",
            "suppressed": "1",
            "acknowledges": [
                {
                    "acknowledgeid": "14443",
                    "userid": "1",
                    "eventid": "1245463",
                    "clock": "1472457281",
                    "message": "problem solved",
                    "action": "6",
                    "old_severity": "0",
                    "new_severity": "0",
                    "suppress_until": "1472511600",
                    "taskid": "0",
                    "maintenanceid": "0"
                }
```

```
            ],
            "opdata": "",
            "suppression_data": [
                {
                    "maintenanceid": "15",
                    "suppress_until": "1472511600",
                    "userid": "0"
                }
            ],
            "tags": [
                {
                    "tag": "test-tag",
                    "value": "test-value"
                }
            ],
            "urls": []
        }
    ],
    "id": 1
}
```

Récupération des problèmes reconnus par un utilisateur spécifié

Récupération des problèmes reconnus par l'utilisateur avec l'ID=10

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "problem.get",
    "params": {
        "output": "extend",
        "action": 2,
        "action_userids": [10],
        "selectAcknowledges": ["userid", "action"],
        "sortfield": ["eventid"],
        "sortorder": "DESC"
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "eventid": "1248566",
            "source": "0",
            "object": "0",
            "objectid": "15142",
            "clock": "1472457242",
            "ns": "209442442",
            "r_eventid": "1245468",
            "r_clock": "1472457285",
            "r_ns": "125644870",
            "correlationid": "0",
            "userid": "10",
            "name": "Zabbix agent on localhost is unreachable for 5 minutes",
            "acknowledged": "1",
            "severity": "3",
            "cause_eventid": "0",
            "suppressed": "0",
            "acknowledges": [
                {
```

```
                    "acknowledgeid": "9",
                    "userid": "10",
                    "action": "2"
                }
            ],
            "opdata": "",
            "urls": []
        }
    ],
    "id": 1
}
```


Voir aussi

- Alerte
- Élément
- Hôte
- Règle LLD
- Déclencheur

Source

CEvent::get() dans *ui/include/classes/api/services/CProblem.php*.

Prototype d'hôte

Cette classe est conçue pour fonctionner avec des prototypes d'hôte.

Références d'objet :

- Prototype d'hôte
- Lien de groupe
- Prototype de groupe
- Balise de prototype d'hôte
- Interface personnalisée
    - Détails de l'interface personnalisée

Méthodes disponibles :

- hostprototype.create - créer de nouveaux prototypes d'hôte
- hostprototype.delete - supprimer des prototypes d'hôte
- hostprototype.get - récupérer des prototypes d'hôte
- hostprototype.update - mettre à jour des prototypes d'hôte

Objet prototype d'hôte

Les objets suivants sont directement liés à l'API `hostprototype`.

Prototype d'hôte

L'objet prototype d'hôte possède les propriétés suivantes.

Propriété	Type	Description
hostid	ID	ID du prototype d'hôte. Comportement de la propriété : - *lecture seule* - *obligatoire* pour les opérations de mise à jour
host	string	Nom technique du prototype d'hôte. Comportement de la propriété : - *obligatoire* pour les opérations de création - *lecture seule* pour les objets hérités

Propriété	Type	Description
name	string	Nom visible du prototype d'hôte. Par défaut : valeur de la propriété `host`. Comportement de la propriété : - *lecture seule* pour les objets hérités
status	integer	Statut du prototype d'hôte. Valeurs possibles : 0 - *(par défaut)* hôte surveillé ; 1 - hôte non surveillé.
flags	integer	Origine du prototype d'hôte. Valeurs possibles : 2 - un prototype d'hôte ; 6 - un prototype d'hôte découvert Comportement de la propriété : - *lecture seule*
inventory_mode	integer	Mode de remplissage de l'inventaire de l'hôte. Valeurs possibles : -1 - *(par défaut)* désactivé ; 0 - manuel ; 1 - automatique.
templateid	ID	ID du modèle parent du prototype d'hôte. Comportement de la propriété : - *lecture seule*
discover	integer	Statut de découverte du prototype d'hôte. Valeurs possibles : 0 - *(par défaut)* les nouveaux hôtes seront découverts ; 1 - les nouveaux hôtes ne seront pas découverts et les hôtes existants seront marqués comme perdus.
custom_interfaces	integer	Source des interfaces personnalisées pour les hôtes créés par le prototype d'hôte. Valeurs possibles : 0 - *(par défaut)* hériter des interfaces de l'hôte parent ; 1 - utiliser les interfaces personnalisées des prototypes d'hôte. Comportement de la propriété : - *lecture seule* pour les objets hérités
uuid	string	Identifiant universel unique, utilisé pour lier les prototypes d'hôte importés à ceux qui existent déjà. Généré automatiquement s'il n'est pas fourni. Comportement de la propriété : - *pris en charge* si le prototype d'hôte appartient à un modèle

Lien de groupe

L'objet de lien de groupe relie un prototype d'hôte à un groupe d'hôtes. Il possède les propriétés suivantes.

Propriété	Type	Description
groupid	ID	ID du groupe d'hôtes. Comportement de la propriété : - *obligatoire*

Prototype de groupe

L'objet prototype de groupe définit un groupe qui sera créé pour un hôte découvert. Il possède les propriétés suivantes.

Propriété	Type	Description
group_prototypeid	ID	ID du prototype de groupe. Comportement de la propriété: - *lecture seule*
name	string	Nom du prototype de groupe. Comportement de la propriété: - *obligatoire* pour les opérations de création

Balise de prototype d'hôte

L'objet de balise de prototype d'hôte possède les propriétés suivantes.

Propriété	Type	Description
tag	string	Nom de la balise du prototype d'hôte. Comportement de la propriété: - *obligatoire*
value	string	Valeur de la balise du prototype d'hôte. Comportement de la propriété: - *lecture seule*.
object	integer	Type d'objet dont la balise a été héritée. Valeurs possibles : 0 - modèle. Comportement de la propriété: - *lecture seule*.
objectid	ID	ID de l'objet dont la balise a été héritée. Comportement de la propriété: - *lecture seule*.

Interface personnalisée

Les interfaces personnalisées sont prises en charge si `custom_interfaces` de l'objet prototype d'hôte est défini sur « use host prototypes custom interfaces ». L'objet d'interface personnalisée possède les propriétés suivantes.

Propriété	Type	Description
type	integer	Type d'interface. Valeurs possibles : 1 - Agent ; 2 - SNMP ; 3 - IPMI ; 4 - JMX. Comportement de la propriété : - *obligatoire*

Propriété	Type	Description
useip	integer	Indique si la connexion doit être établie via IP. Valeurs possibles : 0 - se connecter en utilisant le nom DNS de l'hôte ; 1 - se connecter en utilisant l'adresse IP de l'hôte. Comportement de la propriété : - *obligatoire*
ip	string	Adresse IP utilisée par l'interface. Peut contenir des macros. Comportement de la propriété : - *obligatoire* si `useip` est défini sur « connect using host IP address »
dns	string	Nom DNS utilisé par l'interface. Peut contenir des macros. Comportement de la propriété : - *obligatoire* si `useip` est défini sur « connect using host DNS name »
port	string	Numéro de port utilisé par l'interface. Peut contenir des macros utilisateur et LLD. Comportement de la propriété : - *obligatoire*
main	integer	Indique si l'interface est utilisée par défaut sur l'hôte. Une seule interface d'un certain type peut être définie par défaut sur un hôte. Valeurs possibles : 0 - non par défaut ; 1 - par défaut. Comportement de la propriété : - *obligatoire*
details	object	Objet supplémentaire pour les détails de l'interface personnalisée. Comportement de la propriété : - *obligatoire* si `type` est défini sur « SNMP »

Détails de l'interface personnalisée

L'objet details possède les propriétés suivantes.

Propriété	Type	Description
version	integer	Version de l'interface SNMP. Valeurs possibles : 1 - SNMPv1 ; 2 - SNMPv2c ; 3 - SNMPv3. Comportement de la propriété : - *obligatoire*
bulk	integer	Indique s'il faut utiliser des requêtes SNMP bulk. Valeurs possibles : 0 - ne pas utiliser de requêtes bulk ; 1 - *(par défaut)* - utiliser des requêtes bulk.

Propriété	Type	Description
community	string	Communauté SNMP. Comportement de la propriété : - *obligatoire* si `version` est définie sur ”SNMPv1” ou ”SNMPv2c”
max_repetitions	integer	Valeur de répétition maximale pour les requêtes SNMP bulk natives (GetBulkRequest-PDUs). Utilisée uniquement pour les éléments `discovery[]` et `walk[]` en SNMPv2 et v3. Par défaut : 10.
securityname	string	Nom de sécurité SNMPv3. Comportement de la propriété : - *pris en charge* si `version` est définie sur ”SNMPv3”
securitylevel	integer	Niveau de sécurité SNMPv3. Valeurs possibles : 0 - *(par défaut)* - noAuthNoPriv ; 1 - authNoPriv ; 2 - authPriv. Comportement de la propriété : - *pris en charge* si `version` est définie sur ”SNMPv3”
authpassphrase	string	Phrase secrète d’authentification SNMPv3. Comportement de la propriété : - *pris en charge* si `version` est définie sur ”SNMPv3” et `securitylevel` est défini sur ”authNoPriv” ou ”authPriv”
privpassphrase	string	Phrase secrète de confidentialité SNMPv3. Comportement de la propriété : - *pris en charge* si `version` est définie sur ”SNMPv3” et `securitylevel` est défini sur ”authPriv”
authprotocol	integer	Protocole d’authentification SNMPv3. Valeurs possibles : 0 - *(par défaut)* - MD5 ; 1 - SHA1 ; 2 - SHA224 ; 3 - SHA256 ; 4 - SHA384 ; 5 - SHA512. Comportement de la propriété : - *pris en charge* si `version` est définie sur ”SNMPv3” et `securitylevel` est défini sur ”authNoPriv” ou ”authPriv”
privprotocol	integer	Protocole de confidentialité SNMPv3. Utilisé uniquement par les interfaces SNMPv3. Valeurs possibles : 0 - *(par défaut)* - DES ; 1 - AES128 ; 2 - AES192 ; 3 - AES256 ; 4 - AES192C ; 5 - AES256C. Comportement de la propriété : - *pris en charge* si `version` est définie sur ”SNMPv3” et `securitylevel` est défini sur ”authPriv”

Propriété	Type	Description
contextname	string	Nom de contexte SNMPv3. Comportement de la propriété : - *pris en charge* si `version` est définie sur "SNMPv3"

hostprototype.create

Description

`object hostprototype.create(object/array hostPrototypes)`

Cette méthode vous permet de créer de nouveaux prototypes d'hôte.

> **Note:**
> Cette méthode est disponible uniquement pour les types d'utilisateur *Admin* et *Super admin*. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres des rôles utilisateur. Consultez Rôles utilisateur pour plus d'informations.

Paramètres

`(object/array)` Prototypes d'hôte à créer.

En plus des propriétés standard des prototypes d'hôte, la méthode accepte les paramètres suivants.

Paramètre	Type	Description
groupLinks	array	Liens de groupes à créer pour le prototype d'hôte. Comportement des paramètres: - *obligatoire*
ruleid	ID	ID de la règle LLD à laquelle appartient le prototype d'hôte. Comportement des paramètres: - *obligatoire*
groupPrototypes	array	Prototypes de groupes à créer pour le prototype d'hôte.
macros	object/array	Macros utilisateur à créer pour le prototype d'hôte.
tags	object/array	Tags de prototype d'hôte.
interfaces	object/array	Interfaces personnalisées du prototype d'hôte.
templates	object/array	Modèles à lier au prototype d'hôte. Seule la propriété `templateid` doit être définie pour les modèles.

Valeurs retournées

`(object)` Renvoie un objet contenant les IDs des prototypes d'hôtes créés sous la propriété `hostids`. L'ordre des ID renvoyés correspond à l'ordre des prototypes d'hôtes transmis.

Exemples

Création d'un prototype d'hôte

Créez un prototype d'hôte "{#VM.NAME}" sur la règle LLD "23542" avec un prototype de groupe "{#HV.NAME}", la paire de tags "datacenter": "{#DATACENTER.NAME}" et une interface SNMPv2 personnalisée 127.0.0.1:161 avec la communauté {$SNMP_COMMUNITY}. Liez-le au groupe d'hôtes "2".

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "hostprototype.create",
    "params": {
        "host": "{#VM.NAME}",
        "ruleid": "23542",
        "custom_interfaces": "1",
```

```
        "groupLinks": [
            {
                "groupid": "2"
            }
        ],
        "groupPrototypes": [
            {
                "name": "{#HV.NAME}"
            }
        ],
        "tags": [
            {
                "tag": "datacenter",
                "value": "{#DATACENTER.NAME}"
            }
        ],
        "interfaces": [
            {
                "main": "1",
                "type": "2",
                "useip": "1",
                "ip": "127.0.0.1",
                "dns": "",
                "port": "161",
                "details": {
                    "version": "2",
                    "bulk": "1",
                    "community": "{$SNMP_COMMUNITY}"
                }
            }
        ]
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "10103"
        ]
    },
    "id": 1
}
```

Voir aussi

- Lien de groupe
- Prototype de groupe
- Tag de prototype d'hôte
- Interface personnalisée
- Macro utilisateur

Source

CHostPrototype::create() dans *ui/include/classes/api/services/CHostPrototype.php*.

hostprototype.delete

Description

`object hostprototype.delete(array hostPrototypeIds)`

Cette méthode vous permet de supprimer des prototypes d'hôte.

Note:
Cette méthode est disponible uniquement pour les types d'utilisateur *Admin* et *Super admin*. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres des rôles utilisateur. Consultez Rôles utilisateur pour plus d'informations.

Paramètres

`(array)` ID des prototypes d'hôte à supprimer.

Valeurs de retour

`(object)` Renvoie un objet contenant les ID des prototypes d'hôte supprimés dans la propriété `hostids`.

Exemples

Suppression de plusieurs prototypes d'hôte

Supprimez deux prototypes d'hôte.

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "hostprototype.delete",
    "params": [
        "10103",
        "10105"
    ],
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "10103",
            "10105"
        ]
    },
    "id": 1
}
```

Source

CHostPrototype::delete() dans *ui/include/classes/api/services/CHostPrototype.php*.

hostprototype.get

Description

`integer/array hostprototype.get(object parameters)`

La méthode permet de récupérer des prototypes d'hôte selon les paramètres fournis.

Note:
Cette méthode est disponible pour les utilisateurs de tout type. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres des rôles utilisateur. Consultez Rôles utilisateur pour plus d'informations.

Paramètres

`(object)` Paramètres définissant la sortie souhaitée.

La méthode prend en charge les paramètres suivants.

Paramètre	Type	Description
hostids	ID/array	Retourner uniquement les prototypes d'hôte avec les ID donnés.

Paramètre	Type	Description
discoveryids	ID/array	Retourner uniquement les prototypes d'hôte qui appartiennent aux règles LLD données.
inherited	boolean	Si défini sur `true`, retourner uniquement les éléments hérités d'un modèle.
selectDiscoveryData	query	Retourner une propriété `discoveryData` avec les données de l'objet de découverte du prototype d'hôte. L'objet de découverte du prototype d'hôte relie un prototype d'hôte découvert à un prototype d'hôte à partir duquel il a été découvert. Il possède les propriétés suivantes : `host` - `(string)` ID de l'hôte ; `parent_hostid` - `(string)` ID du prototype d'hôte à partir duquel le prototype d'hôte a été créé ; `status` - `(int)` statut de découverte du prototype d'hôte : 0 - *(par défaut)* le prototype d'hôte est découvert, 1 - le prototype d'hôte n'est plus découvert ; `ts_delete` - `(timestamp)` heure à laquelle un prototype d'hôte qui n'est plus découvert sera supprimé ; `ts_disable` - `(timestamp)` heure à laquelle un prototype d'hôte qui n'est plus découvert sera désactivé ; `disable_source` - `(int)` indicateur précisant si le prototype d'hôte a été désactivé par une règle LLD ou manuellement : 0 - *(par défaut)* désactivé automatiquement, 1 - désactivé par une règle LLD.
selectDiscoveryRule	query	Retourner une propriété `discoveryRule` avec la règle LLD à laquelle appartient le prototype d'hôte.
selectDiscoveryRulePrototype	query	Retourner une propriété `discoveryRulePrototype` avec le prototype de règle LLD parent auquel appartient le prototype d'hôte.
selectInterfaces	query	Retourner une propriété `interfaces` avec les interfaces personnalisées du prototype d'hôte.
selectGroupLinks	query	Retourner une propriété `groupLinks` avec les liens de groupes du prototype d'hôte.
selectGroupPrototypes	query	Retourner une propriété `groupPrototypes` avec les prototypes de groupes du prototype d'hôte.
selectInheritedTags	query	Retourner une propriété `inheritedTags` avec les tags présents sur les modèles liés.
selectMacros	query	Retourner une propriété `macros` avec les macros du prototype d'hôte.
selectParentHost	query	Retourner une propriété `parentHost` avec l'hôte auquel appartient le prototype d'hôte.
selectTags	query	Retourner une propriété `tags` avec les tags du prototype d'hôte.
selectTemplates	query	Retourner une propriété `templates` avec les modèles liés au prototype d'hôte. Prend en charge `count`.
sortfield	string/array	Trier le résultat selon les propriétés données. Valeurs possibles : `hostid`, `host`, `name`, `status`, `discovered`.
countOutput	boolean	Ces paramètres sont décrits dans le commentaire de référence.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Valeurs de retour

(integer/array) Renvoie soit :

- un tableau d'objets ;
- le nombre d'objets récupérés, si le paramètre countOutput a été utilisé.

Exemples

Récupération des prototypes d'hôte à partir d'une règle LLD

Récupérez tous les prototypes d'hôte, leurs liens de groupe, prototypes de groupe et tags à partir d'une règle LLD.

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "hostprototype.get",
    "params": {
        "output": "extend",
        "selectInterfaces": "extend",
        "selectGroupLinks": "extend",
        "selectGroupPrototypes": "extend",
        "selectTags": "extend",
        "discoveryids": "23554"
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "hostid": "10092",
            "host": "{#HV.UUID}",
            "name": "{#HV.UUID}",
            "status": "0",
            "templateid": "0",
            "discover": "0",
            "custom_interfaces": "1",
            "uuid": "051a1469d4d045cbbf818fcc843a352e",
            "flags": "2",
            "inventory_mode": "-1",
            "tags": [
                {
                    "tag": "datacenter",
                    "value": "{#DATACENTER.NAME}"
                },
                {
                    "tag": "instance-type",
                    "value": "{#INSTANCE_TYPE}"
                }
            ],
            "groupLinks": [
                {
                    "group_prototypeid": "4",
                    "hostid": "10092",
                    "groupid": "7",
                    "templateid": "0"
                }
            ],
            "groupPrototypes": [
                {
                    "group_prototypeid": "7",
                    "hostid": "10092",
                    "name": "{#CLUSTER.NAME}",
                    "templateid": "0"
```

```
                }
            ],
            "interfaces": [
                {
                    "main": "1",
                    "type": "2",
                    "useip": "1",
                    "ip": "127.0.0.1",
                    "dns": "",
                    "port": "161",
                    "available": "0",
                    "error": "",
                    "errors_from": "0",
                    "disable_until": "0",
                    "details": {
                        "version": "2",
                        "bulk": "1",
                        "community": "{$SNMP_COMMUNITY}",
                        "max_repetitions": "10"
                    }
                }
            ]
        }
    ],
    "id": 1
}
```


Voir aussi

- Lien de groupe
- Prototype de groupe
- Macro utilisateur

Source

CHostPrototype::get() dans *ui/include/classes/api/services/CHostPrototype.php*.

hostprototype.update

Description

`object hostprototype.update(object/array hostPrototypes)`

Cette méthode vous permet de mettre à jour des prototypes d'hôte existants.

Note:
Cette méthode est disponible uniquement pour les types d'utilisateur *Admin* et *Super admin*. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres des rôles utilisateur. Consultez Rôles utilisateur pour plus d'informations.

Paramètres

`(object/array)` Propriétés du prototype d'hôte à mettre à jour.

La propriété `hostid` doit être définie pour chaque prototype d'hôte, toutes les autres propriétés sont facultatives. Seules les propriétés transmises seront mises à jour, toutes les autres resteront inchangées.

En plus des propriétés standard du prototype d'hôte, la méthode accepte les paramètres suivants.

Paramètre	Type	Description
groupLinks	array	Liens de groupes pour remplacer les liens de groupes actuels sur le prototype d'hôte. Comportement des paramètres : - *lecture seule* pour les objets hérités

Paramètre	Type	Description
groupPrototypes	array	Prototypes de groupes pour remplacer les prototypes de groupes existants sur le prototype d'hôte. Tous les prototypes de groupes qui ne sont pas listés dans la requête seront supprimés. Comportement des paramètres : - *lecture seule* pour les objets hérités
macros	object/array	Macros utilisateur pour remplacer les macros utilisateur actuelles. Toutes les macros qui ne sont pas listées dans la requête seront supprimées.
tags	object/array	Tags du prototype d'hôte pour remplacer les tags actuels. Tous les tags qui ne sont pas listés dans la requête seront supprimés. Comportement des paramètres : - *lecture seule* pour les objets hérités
interfaces	object/array	Interfaces personnalisées du prototype d'hôte pour remplacer les interfaces actuelles. L'objet d'interface personnalisée doit contenir tous ses paramètres. Toutes les interfaces qui ne sont pas listées dans la requête seront supprimées. Comportement des paramètres : - *pris en charge* si `custom_interfaces` de l'objet prototype d'hôte est défini sur "use host prototypes custom interfaces" - *lecture seule* pour les objets hérités
templates	object/array	Modèles pour remplacer les modèles actuellement liés. Seule la propriété `templateid` doit être définie pour les modèles.

Valeurs retournées

`(object)` Retourne un objet contenant les identifiants des prototypes hôte mis à jour sous la propriété `hostids`.

Exemples

Désactivation d'un prototype d'hôte

Désactivez un prototype d'hôte, c'est-à-dire définissez son statut sur « 1 ».

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "hostprototype.update",
    "params": {
        "hostid": "10092",
        "status": 1
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "10092"
        ]
    },
    "id": 1
}
```

Mise à jour des tags du prototype d'hôte

Remplacez les tags du prototype d'hôte par de nouveaux.

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "hostprototype.update",
    "params": {
        "hostid": "10092",
        "tags": [
            {
                "tag": "datacenter",
                "value": "{#DATACENTER.NAME}"
            },
            {
                "tag": "instance-type",
                "value": "{#INSTANCE_TYPE}"
            }
        ]
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "10092"
        ]
    },
    "id": 1
}
```

Mise à jour des interfaces personnalisées du prototype d'hôte

Remplacez les interfaces héritées par les interfaces personnalisées du prototype d'hôte.

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "hostprototype.update",
    "params": {
        "hostid": "10092",
        "custom_interfaces": "1",
        "interfaces": [
            {
                "main": "1",
                "type": "2",
                "useip": "1",
                "ip": "127.0.0.1",
                "dns": "",
                "port": "161",
                "details": {
                    "version": "2",
                    "bulk": "1",
                    "community": "{$SNMP_COMMUNITY}"
                }
            }
        ]
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "10092"
        ]
    },
    "id": 1
}
```

Voir aussi

- Lien de groupe
- Prototype de groupe
- Tag de prototype d'hôte
- Interface personnalisée
- Macro utilisateur

Source

CHostPrototype::update() dans *ui/include/classes/api/services/CHostPrototype.php*.

Prototype d'élément

Cette classe est conçue pour fonctionner avec des prototypes d'élément.

Références d'objet :

- Prototype d'élément
  - En-tête HTTP
  - Champ de requête HTTP
- Tag de prototype d'élément
- Prétraitement de prototype d'élément

Méthodes disponibles :

- itemprototype.create - créer de nouveaux prototypes d'élément
- itemprototype.delete - supprimer des prototypes d'élément
- itemprototype.get - récupérer des prototypes d'élément
- itemprototype.update - mettre à jour des prototypes d'élément

Objet prototype d'élément

Les objets suivants sont directement liés à l'API `itemprototype`.

Prototype d'élément

L'objet prototype d'élément possède les propriétés suivantes.

Property	Type	Description
itemid	ID	ID du prototype d'élément. Comportement de la propriété : - *en lecture seule* - *obligatoire* pour les opérations de mise à jour

Property	Type	Description
delay	string	Intervalle de mise à jour du prototype d'élément. Accepte un nombre de secondes (par exemple, 1, 60, 300), une valeur avec un suffixe temporel (par exemple, 30s, 1m, 2h, 1d) ou, éventuellement, un ou plusieurs intervalles personnalisés, tous séparés par des points-virgules. Les intervalles personnalisés peuvent combiner des intervalles flexibles et des intervalles de planification. Accepte les macros utilisateur et les macros LLD. Si elle est utilisée, la valeur doit être une macro unique. Plusieurs macros ou des macros combinées à du texte ne sont pas prises en charge. Les intervalles flexibles peuvent être écrits sous la forme de deux macros séparées par une barre oblique (par exemple, {$FLEX_INTERVAL}/{$FLEX_PERIOD}). Exemple : 1h;wd1-5h9-18;{$Macro1}/1-7,00:00-24:00;0/6-7,12:00-24:00;{$Macr Comportement de la propriété : - *obligatoire* si type est défini sur « Zabbix agent » (0), « Simple check » (3), « Zabbix internal » (5), « External check » (10), « Database monitor » (11), « IPMI agent » (12), « SSH agent » (13), « TELNET agent » (14), « Calculated » (15), « JMX agent » (16), « HTTP agent » (19), « SNMP agent » (20), « Script » (21), « Browser » (22), « Telemetry query » (24), ou si type est défini sur « Zabbix agent (active) » (7) et que key_ ne contient pas « mqtt.get »
hostid	ID	ID du hôte auquel appartient le prototype d'élément. Comportement de la propriété : - *constant* - *obligatoire* pour les opérations de création
interfaceid	ID	ID de l'interface du hôte du prototype d'élément. Comportement de la propriété : - *obligatoire* si le prototype d'élément appartient à un hôte et que type est défini sur « Zabbix agent », « IPMI agent », « JMX agent », « SNMP trap » ou « SNMP agent » - *pris en charge* si le prototype d'élément appartient à un hôte et que type est défini sur « Simple check », « External check », « SSH agent », « TELNET agent » ou « HTTP agent »
key_	string	Clé du prototype d'élément. Comportement de la propriété : - *obligatoire* pour les opérations de création - *en lecture seule* pour les objets hérités
name	string	Nom du prototype d'élément. Prend en charge les macros utilisateur. Comportement de la propriété : - *obligatoire* pour les opérations de création - *en lecture seule* pour les objets hérités

Property	Type	Description
type	integer	Type du prototype d'élément. Valeurs possibles : 0 - Zabbix agent ; 2 - Zabbix trapper ; 3 - Simple check ; 5 - Zabbix internal ; 7 - Zabbix agent (active) ; 10 - External check ; 11 - Database monitor ; 12 - IPMI agent ; 13 - SSH agent ; 14 - TELNET agent ; 15 - Calculated ; 16 - JMX agent ; 17 - SNMP trap ; 18 - Dependent item ; 19 - HTTP agent ; 20 - SNMP agent ; 21 - Script ; 22 - Browser ; 24 - Telemetry query. Comportement de la propriété : - *obligatoire* pour les opérations de création - *en lecture seule* pour les objets hérités
url	string	Chaîne URL. Prend en charge les macros LLD, les macros utilisateur, {HOST.IP}, {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.NAME}, {HOST.PORT}, {ITEM.ID}, {ITEM.KEY}. Comportement de la propriété : - *obligatoire* si `type` est défini sur « HTTP agent » - *en lecture seule* pour les objets hérités
value_type	integer	Type d'information du prototype d'élément. Valeurs possibles : 0 - nombre décimal ; 1 - caractère ; 2 - journal ; 3 - nombre entier non signé ; 4 - texte ; 5 - binaire ; 6 - JSON. Comportement de la propriété : - *obligatoire* pour les opérations de création - *en lecture seule* pour les objets hérités
allow_traps	integer	Autorise le remplissage de la valeur de manière similaire à l'élément trapper. 0 - *(par défaut)* Ne pas accepter les données entrantes ; 1 - Accepter les données entrantes. Comportement de la propriété : - *pris en charge* si `type` est défini sur « HTTP agent »

Property	Type	Description
authtype	integer	Méthode d'authentification. Valeurs possibles si `type` est défini sur « SSH agent » : 0 - *(par défaut)* mot de passe ; 1 - clé publique. Valeurs possibles si `type` est défini sur « HTTP agent » : 0 - *(par défaut)* aucune ; 1 - basic ; 2 - NTLM ; 3 - Kerberos ; 4 - Digest. Comportement de la propriété : - *pris en charge* si `type` est défini sur « SSH agent » ou « HTTP agent » - *en lecture seule* pour les objets hérités (si `type` est défini sur « HTTP agent »)
description	string	Description du prototype d'élément.
flags	integer	Origine du prototype d'élément. Valeurs possibles : 2 - un prototype d'élément ; 6 - un prototype d'élément découvert Comportement de la propriété : - *en lecture seule*
follow_redirects	integer	Suit les redirections de la réponse lors de l'interrogation des données. Valeurs possibles : 0 - Ne pas suivre les redirections ; 1 - *(par défaut)* Suivre les redirections. Comportement de la propriété : - *pris en charge* si `type` est défini sur « HTTP agent » - *en lecture seule* pour les objets hérités
headers	array	Tableau d'en-têtes envoyés lors de l'exécution d'une requête HTTP. Comportement de la propriété : - *pris en charge* si `type` est défini sur « HTTP agent » - *en lecture seule* pour les objets hérités
history	string	Unité de temps indiquant pendant combien de temps les données d'historique doivent être conservées. Accepte également une macro utilisateur et une macro LLD. Par défaut : 31d.
http_proxy	string	Chaîne de connexion au proxy HTTP(S). Comportement de la propriété : - *pris en charge* si `type` est défini sur « HTTP agent » - *en lecture seule* pour les objets hérités
ipmi_sensor	string	Capteur IPMI. Comportement de la propriété : - *obligatoire* si `type` est défini sur « IPMI agent » et que `key_` n'est pas défini sur « ipmi.get » - *pris en charge* si `type` est défini sur « IPMI agent » - *en lecture seule* pour les objets hérités

Property	Type	Description
jmx_endpoint	string	Chaîne de connexion personnalisée de Java gateway. Par défaut : service:jmx:rmi:///jndi/rmi://{HOST.CONN}:{HOST.PORT}/jmxrmi Comportement de la propriété : - *pris en charge* si `type` est défini sur « JMX agent »
logtimefmt	string	Format de l'heure dans les entrées de journal. Comportement de la propriété : - *pris en charge* si `value_type` est défini sur « log » - *en lecture seule* pour les objets hérités
master_itemid	ID	ID de l'élément maître. Comportement de la propriété : - *obligatoire* si `type` est défini sur « Dependent item » - *en lecture seule* pour les objets hérités
output_format	integer	Indique si la réponse doit être convertie en JSON. Valeurs possibles : 0 - *(par défaut)* Stocker les données brutes ; 1 - Convertir en JSON. Comportement de la propriété : - *pris en charge* si `type` est défini sur « HTTP agent » - *en lecture seule* pour les objets hérités
params	string	Paramètres supplémentaires selon le type du prototype d'élément : - script exécuté pour les prototypes d'éléments SSH agent et TELNET agent ; - requête SQL pour les prototypes d'éléments Database monitor ; - formule pour les prototypes d'éléments Calculated ; - script pour les prototypes d'éléments Script et Browser. Comportement de la propriété : - *obligatoire* si `type` est défini sur « Database monitor », « SSH agent », « TELNET agent », « Calculated », « Script » ou « Browser » - *en lecture seule* pour les objets hérités (si `type` est défini sur « Script » ou « Browser »)
parameters	object/array	Paramètres supplémentaires si `type` est défini sur « Script » ou « Browser ». Tableau d'objets possédant les propriétés `name` et `value`, où `name` doit être unique. Comportement de la propriété : - *pris en charge* si `type` est défini sur « Script » ou « Browser » - *en lecture seule* pour les objets hérités
password	string	Mot de passe d'authentification. Comportement de la propriété : - *obligatoire* si `type` est défini sur « JMX agent » et que `username` est défini - *pris en charge* si `type` est défini sur « Simple check », « SSH agent », « TELNET agent », « Database monitor » ou « HTTP agent » - *en lecture seule* pour les objets hérités (si `type` est défini sur « HTTP agent »)

Property	Type	Description
post_type	integer	Type du corps des données POST stocké dans la propriété `posts`. Valeurs possibles : 0 - *(par défaut)* Données brutes. 2 - Données JSON. 3 - Données XML. Comportement de la propriété : - *pris en charge* si `type` est défini sur « HTTP agent » - *en lecture seule* pour les objets hérités
posts	string	Données du corps de la requête HTTP(S). Comportement de la propriété : - *obligatoire* si `type` est défini sur « HTTP agent » et que `post_type` est défini sur « Données JSON » ou « Données XML » - *pris en charge* si `type` est défini sur « HTTP agent » et que `post_type` est défini sur « Données brutes » - *en lecture seule* pour les objets hérités
privatekey	string	Nom du fichier de clé privée. Comportement de la propriété : - *obligatoire* si `type` est défini sur « SSH agent » et que `authtype` est défini sur « clé publique »
publickey	string	Nom du fichier de clé publique. Comportement de la propriété : - *obligatoire* si `type` est défini sur « SSH agent » et que `authtype` est défini sur « clé publique »
query_fields	array	Tableau de champs de requête envoyés lors de l'exécution d'une requête HTTP. Comportement de la propriété : - *pris en charge* si `type` est défini sur « HTTP agent » - *en lecture seule* pour les objets hérités
request_method	integer	Type de méthode de requête. Valeurs possibles : 0 - *(par défaut)* GET ; 1 - POST ; 2 - PUT ; 3 - HEAD. Comportement de la propriété : - *pris en charge* si `type` est défini sur « HTTP agent » - *en lecture seule* pour les objets hérités
retrieve_mode	integer	Partie de la réponse à stocker. Valeurs possibles si `request_method` est défini sur « GET », « POST » ou « PUT » : 0 - *(par défaut)* Corps ; 1 - En-têtes ; 2 - Le corps et les en-têtes sont stockés. Valeurs possibles si `request_method` est défini sur « HEAD » : 1 - En-têtes. Comportement de la propriété : - *pris en charge* si `type` est défini sur « HTTP agent » - *en lecture seule* pour les objets hérités

Property	Type	Description
snmp_oid	string	OID SNMP. Comportement de la propriété : - *obligatoire* si `type` est défini sur « SNMP agent » - *en lecture seule* pour les objets hérités
ssl_cert_file	string	Chemin du fichier de clé SSL publique. Comportement de la propriété : - *pris en charge* si `type` est défini sur « HTTP agent » - *en lecture seule* pour les objets hérités
ssl_key_file	string	Chemin du fichier de clé SSL privée. Comportement de la propriété : - *pris en charge* si `type` est défini sur « HTTP agent » - *en lecture seule* pour les objets hérités
ssl_key_password	string	Mot de passe du fichier de clé SSL. Comportement de la propriété : - *pris en charge* si `type` est défini sur « HTTP agent » - *en lecture seule* pour les objets hérités
status	integer	État du prototype d'élément. Valeurs possibles : 0 - *(par défaut)* prototype d'élément activé ; 1 - prototype d'élément désactivé ; 3 - prototype d'élément non pris en charge.
status_codes	string	Plages de codes d'état HTTP requis, séparées par des virgules. Prend également en charge les macros utilisateur ou les macros LLD dans une liste séparée par des virgules. Exemple : 200,200-{$M},{$M},200-400 Comportement de la propriété : - *pris en charge* si `type` est défini sur « HTTP agent » - *en lecture seule* pour les objets hérités
templateid	ID	ID du prototype d'élément du modèle parent. Comportement de la propriété : - *en lecture seule*
timeout	string	Délai d'expiration de la requête d'interrogation des données de l'élément. Accepte un nombre de secondes (par exemple, 1, 60, 300) ou une valeur avec un suffixe temporel (par exemple, 30s, 1m). Accepte également les macros utilisateur et les macros LLD. Plage de valeurs possibles : 1-600s. Par défaut : "" - utiliser les paramètres du proxy ou les paramètres globaux. Comportement de la propriété : - *pris en charge* si `type` est défini sur « Zabbix agent » (0), « Simple check » (3) et que `key_` ne commence pas par « `vmware.` » ni « `icmpping` », « Zabbix agent (active) » (7), « External check » (10), « Database monitor » (11), « SSH agent » (13), « TELNET agent » (14), « HTTP agent » (19), « SNMP agent » (20) et que `snmp_oid` commence par « `walk[` » ou « `get[` », « Script » (21), « Browser » (22) - *en lecture seule* pour les objets hérités

Property	Type	Description
trapper_hosts	string	Hôtes autorisés. Comportement de la propriété : - *pris en charge* si `type` est défini sur « Zabbix trapper », ou si `type` est défini sur « HTTP agent » et que `allow_traps` est défini sur « Allow incoming data ». Si la propriété est omise, elle est automatiquement définie sur la macro utilisateur `{$TRAPPER.ALLOWED_HOSTS}` (valeur par défaut). Par défaut, `{$TRAPPER.ALLOWED_HOSTS}` vaut `127.0.0.1,::1`. Si la propriété est définie sur une chaîne vide, les connexions entrantes seront refusées pour tous les hôtes.
trends	string	Unité de temps indiquant pendant combien de temps les données de tendance doivent être conservées. Accepte également une macro utilisateur et une macro LLD. Par défaut : 365d. Comportement de la propriété : - *pris en charge* si `value_type` est défini sur « numeric float » ou « numeric unsigned »
units	string	Unités de la valeur. Comportement de la propriété : - *pris en charge* si `value_type` est défini sur « numeric float » ou « numeric unsigned » - *en lecture seule* pour les objets hérités
username	string	Nom d'utilisateur pour l'authentification. Comportement de la propriété : - *obligatoire* si `type` est défini sur « SSH agent » ou « TELNET agent », ou si `type` est défini sur « JMX agent » et que `password` est défini - *pris en charge* si `type` est défini sur « Simple check », « Database monitor » ou « HTTP agent » - *en lecture seule* pour les objets hérités (si `type` est défini sur « HTTP agent »)
uuid	string	Identifiant unique universel, utilisé pour associer les prototypes d'éléments importés à ceux qui existent déjà. Généré automatiquement s'il n'est pas fourni. Comportement de la propriété : - *pris en charge* si le prototype d'élément appartient à un modèle
valuemapid	ID	ID de la correspondance de valeurs associée. Comportement de la propriété : - *pris en charge* si `value_type` est défini sur « numeric float », « character » ou « numeric unsigned » - *en lecture seule* pour les objets hérités
verify_host	integer	Indique s'il faut vérifier que le nom du hôte de la connexion correspond à celui indiqué dans le certificat du hôte. Valeurs possibles : 0 - *(par défaut)* Ne pas vérifier ; 1 - Vérifier. Comportement de la propriété : - *pris en charge* si `type` est défini sur « HTTP agent » - *en lecture seule* pour les objets hérités

Property	Type	Description
verify_peer	integer	Indique s'il faut vérifier que le certificat du hôte est authentique. Valeurs possibles : 0 - *(par défaut)* Ne pas vérifier ; 1 - Vérifier. Comportement de la propriété : - *pris en charge* si `type` est défini sur « HTTP agent » - *en lecture seule* pour les objets hérités
discover	integer	État de découverte du prototype d'élément. Valeurs possibles : 0 - *(par défaut)* les nouveaux éléments seront découverts ; 1 - les nouveaux éléments ne seront pas découverts et les éléments existants seront marqués comme perdus.
query	object	Configuration de la requête de télémétrie. Définit ce que l'élément interroge dans la source de données de télémétrie, la manière dont il filtre les données et la manière dont il agrège le résultat. Comportement de la propriété : - *obligatoire* si `type` est défini sur « Telemetry query » - *en lecture seule* pour les objets hérités ou découverts
time_shift	string	Décalage vers le passé (avant l'heure actuelle) marquant la fin de la plage temporelle interrogée par l'élément. Accepte un nombre de secondes (par exemple, `1`, `60`, `300`) ou une valeur avec un suffixe temporel (par exemple, `30s`, `1m`, `2h`, `1d`). Accepte également les macros utilisateur et les macros LLD. Plage de valeurs possibles : 0-86400s. Par défaut : 15s. Comportement de la propriété : - *pris en charge* si `type` est défini sur « Telemetry query »
lookback_limit	string	Décalage vers le passé (avant l'heure actuelle) pendant lequel l'élément recherche des données lorsqu'il n'existe aucune requête précédente à poursuivre (par exemple, lors de la première exécution de l'élément), ou lorsque la requête précédente est plus ancienne que `delay` (par exemple, parce que le serveur Zabbix était arrêté). Accepte un nombre de secondes (par exemple, `1`, `60`, `300`) ou une valeur avec un suffixe temporel (par exemple, `30s`, `1m`, `2h`, `1d`). Accepte également les macros utilisateur et les macros LLD. Plage de valeurs possibles : 1-259200s. Par défaut : 10m. Comportement de la propriété : - *pris en charge* si `type` est défini sur « Telemetry query »

Property	Type	Description
granularity	string	Durée d'une portion en laquelle la plage temporelle complète de la requête est divisée. Accepte un nombre de secondes (par exemple, `1`, `60`, `300`) ou une valeur avec un suffixe temporel (par exemple, `30s`, `1m`, `2h`, `1d`). Accepte également les macros utilisateur et les macros LLD. Plage de valeurs possibles : 1-86400s. Par défaut : `15s`. Comportement de la propriété : - *pris en charge* si `type` est défini sur « Telemetry query »

En-tête HTTP

L'objet d'en-tête possède les propriétés suivantes :

Propriété	Type	Description
name	string	Nom de l'en-tête HTTP. Comportement de la propriété : - *obligatoire*
value	string	Valeur de l'en-tête. Comportement de la propriété : - *obligatoire*

Champ de requête HTTP

L'objet de champ de requête définit un nom et une valeur utilisés pour spécifier un paramètre d'URL. Il possède les propriétés suivantes :

Propriété	Type	Description
name	string	Nom du paramètre. Comportement de la propriété : - *obligatoire*
value	string	Valeur du paramètre. Comportement de la propriété : - *obligatoire*

Requête de télémétrie

L'objet de requête de télémétrie possède les propriétés suivantes.

Property	Type	Description
signal_type	integer	Type de signal interrogé par l'élément. Valeurs possibles : 0 - Traces ; 1 - Métriques ; 2 - Journaux. Comportement de la propriété : - *required*
metric_point_type	integer	Type de point de métrique interrogé par l'élément. Valeurs possibles : 0 - Somme ; 1 - Jauge ; 2 - Histogramme ; 3 - Histogramme exponentiel. Comportement de la propriété : - *required* si `signal_type` est défini sur « Metrics »
columns	array	Tableau de colonnes de données de télémétrie interrogées par l'élément. Peut être vide. Comportement de la propriété : - *required*
aggregated_columns	array	Tableau de colonnes agrégées de données de télémétrie interrogées par l'élément, ainsi que la manière dont il calcule une valeur à partir des enregistrements de colonnes. Comportement de la propriété : - *required*
filter	object	Filtre de requête de télémétrie pour les enregistrements de colonnes de la requête. Comportement de la propriété : - *required*

Colonnes de données de télémétrie

L'objet des colonnes de données de télémétrie possède les propriétés suivantes.

Property	Type	Description
column	string	Nom de la colonne de données de télémétrie interrogée par l'élément. Voir les colonnes de données de télémétrie prises en charge. Pour plusieurs colonnes à valeur unique, chaque valeur de `column` doit être unique. Pour plusieurs colonnes contenant des paires clé-valeur, chaque paire `column` et `attribute_key` doit être unique.
attribute_key	string	Nom de la clé interrogée par l'élément, dans la colonne contenant des paires clé-valeur. Pour plusieurs colonnes contenant des paires clé-valeur, chaque paire `column` et `attribute_key` doit être unique. Comportement de la propriété: - *required* si `column` est défini sur une colonne contenant des paires clé-valeur

Colonnes agrégées des données de télémétrie

L'objet des colonnes agrégées des données de télémétrie possède les propriétés suivantes.

Property	Type	Description
column	string	Nom de la colonne de données de télémétrie interrogée par l'élément. Voir colonnes de données de télémétrie prises en charge. Pour plusieurs colonnes, chaque valeur de `column` doit être unique. Cette propriété n'est pas prise en charge si `function` est défini sur "Count". Comportement de la propriété : - *obligatoire* si `function` est défini sur "Min", "Max", "Average", "Sum" ou "Percentile"
function	integer	Fonction que l'élément applique aux enregistrements de la colonne. Valeurs possibles : 1 - Min ; 2 - Max ; 3 - Average ; 4 - Count ; 5 - Sum ; 8 - Percentile. Comportement de la propriété : - *obligatoire*
parameters	array	Percentile que l'élément calcule à partir des enregistrements de la colonne. Valeur possible : un nombre compris entre 0 et 100, fourni sous forme de chaîne. Comportement de la propriété : - *obligatoire* si `function` est défini sur "Percentile"
alias	string	Nom unique de la valeur calculée, utilisé pour l'identifier dans la valeur de l'élément. Chaque valeur de `alias` doit être unique. Comportement de la propriété : - *obligatoire*

Filtre de données de télémétrie

L'objet de filtre de données de télémétrie possède les propriétés suivantes.

Property	Type	Description
conditions	array	Conditions du filtre de données de télémétrie auxquelles un enregistrement d'une colonne de données de télémétrie doit satisfaire. Comportement de la propriété : - *obligatoire*
evaltype	integer	Méthode d'évaluation des conditions du filtre. Valeurs possibles : 0 - Et/Ou ; 1 - Et ; 2 - Ou ; 3 - Expression personnalisée. Comportement de la propriété : - *obligatoire*

Property	Type	Description
eval_formula	string	Expression générée utilisée pour évaluer les conditions du filtre. L'expression contient des ID qui font référence à des conditions de filtre spécifiques via leur `formulaid`. La valeur de `eval_formula` est égale à la valeur de `formula` pour les filtres utilisant une expression personnalisée. Comportement de la propriété : - *en lecture seule*
formula	string	Expression définie par l'utilisateur utilisée pour évaluer les conditions des filtres utilisant une expression personnalisée. L'expression doit contenir des ID qui font référence à des conditions de filtre spécifiques via leur `formulaid`. Les ID utilisés dans l'expression doivent correspondre exactement à ceux définis dans les conditions du filtre : aucune condition ne peut rester inutilisée ou être omise. Comportement de la propriété : - *obligatoire* si `evaltype` est défini sur « Expression personnalisée »

Conditions de filtrage des données de télémétrie

L'objet des conditions de filtrage des données de télémétrie possède les propriétés suivantes.

Property	Type	Description
column	string	Nom de la colonne de données de télémétrie à laquelle s'applique la condition de filtrage. Voir les colonnes de données de télémétrie prises en charge. Comportement de la propriété : - *obligatoire*
attribute_key	string	Nom de la clé à laquelle s'applique la condition, dans la colonne contenant des paires clé-valeur. Comportement de la propriété : - *obligatoire* si `column` est défini sur une colonne contenant des paires clé-valeur
operator	integer	Opérateur de la condition de filtrage. Valeurs possibles pour les colonnes à valeur unique : 0 - Est égal à ; 1 - N'est pas égal à ; 2 - Contient ; 3 - Ne contient pas. Valeurs possibles pour les colonnes contenant des paires clé-valeur : 0 - Est égal à ; 1 - N'est pas égal à ; 12 - Existe. Comportement de la propriété : - *obligatoire*
value	string	Valeur à comparer. Comportement de la propriété : - *obligatoire* si `operator` est défini sur « Est égal à », « N'est pas égal à », « Contient » ou « Ne contient pas »

Property	Type	Description
formulaid	string	ID unique arbitraire utilisé pour référencer la condition depuis une expression personnalisée. Ne peut contenir que des lettres majuscules. L'ID doit être défini par l'utilisateur lors de la modification des conditions de filtrage, mais sera généré à nouveau lors de leur demande ultérieure. Comportement de la propriété : - *obligatoire* si `evaltype` de l'objet Filtre de requête de télémétrie est défini sur « Expression personnalisée »

Tag d'un prototype d'élément

L'objet de tag d'un prototype d'élément possède les propriétés suivantes.

Propriété	Type	Description
tag	string	Nom du tag du prototype d'élément. Comportement de la propriété: - *obligatoire*
value	string	Valeur du tag du prototype d'élément. Comportement de la propriété: - *lecture seule*.
object	integer	Type d'objet dont le tag a été hérité. Valeurs possibles : 0 - modèle; 1 - hôte. Comportement de la propriété: - *lecture seule*.
objectid	ID	ID de l'objet dont le tag a été hérité. Comportement de la propriété: - *lecture seule*.

Prétraitement du prototype d'élément

L'objet de prétraitement du prototype d'élément possède les propriétés suivantes.

Propriété	Type	Description
type	integer	Type de l'option de prétraitement. Valeurs possibles : 1 - Multiplicateur personnalisé ; 2 - Suppression des espaces à droite ; 3 - Suppression des espaces à gauche ; 4 - Suppression des espaces ; 5 - Expression régulière ; 6 - Conversion de booléen en décimal ; 7 - Conversion d'octal en décimal ; 8 - Conversion d'hexadécimal en décimal ; 9 - Modification simple ; 10 - Modification par seconde ; 11 - XPath XML ; 12 - JSONPath ; 13 - Dans l'intervalle ; 14 - Correspond à l'expression régulière ; 15 - Ne correspond pas à l'expression régulière ; 16 - Vérification des erreurs dans JSON ; 17 - Vérification des erreurs dans XML ; 18 - Vérification des erreurs à l'aide d'une expression régulière ; 19 - Ignorer les valeurs inchangées ; 20 - Ignorer les valeurs inchangées avec battement de cœur ; 21 - JavaScript ; 22 - Modèle Prometheus ; 23 - Conversion de Prometheus en JSON ; 24 - Conversion de CSV en JSON ; 25 - Remplacement ; 26 - Vérification des éléments non pris en charge ; 27 - Conversion de XML en JSON ; 28 - Valeur de parcours SNMP ; 29 - Conversion du parcours SNMP en JSON ; 30 - Valeur SNMP get. Comportement de la propriété : *- required*

Propriété	Type	Description
params	string	Paramètres supplémentaires utilisés par l'option de prétraitement. Plusieurs paramètres sont séparés par le caractère de nouvelle ligne (\n). Si `type` est défini sur « Vérification des éléments non pris en charge », les paramètres suivent la syntaxe `<scope>[\n<pattern>]`, où *pattern* est une expression régulière et *scope* est l'une des valeurs suivantes : -1 - correspond à toute erreur ; 0 - vérifie si le message d'erreur correspond à *pattern* ; 1 - vérifie si le message d'erreur ne correspond pas à *pattern*. Comportement de la propriété : - *required* si `type` est défini sur « Multiplicateur personnalisé » (1), « Suppression des espaces à droite » (2), « Suppression des espaces à gauche » (3), « Suppression des espaces » (4), « Expression régulière » (5), « XPath XML » (11), « JSONPath » (12), « Dans l'intervalle » (13), « Correspond à l'expression régulière » (14), « Ne correspond pas à l'expression régulière » (15), « Vérification des erreurs dans JSON » (16), « Vérification des erreurs dans XML » (17), « Vérification des erreurs à l'aide d'une expression régulière » (18), « Ignorer les valeurs inchangées avec battement de cœur » (20), « JavaScript » (21), « Modèle Prometheus » (22), « Conversion de Prometheus en JSON » (23), « Conversion de CSV en JSON » (24), « Remplacement » (25), « Vérification des éléments non pris en charge » (26), « Valeur de parcours SNMP » (28), « Conversion du parcours SNMP en JSON » (29) ou « Valeur SNMP get » (30)
error_handler	integer	Type d'action utilisé en cas d'échec d'une étape de prétraitement. Valeurs possibles : 0 - Le message d'erreur est défini par le serveur Zabbix ; 1 - Ignorer la valeur ; 2 - Définir une valeur personnalisée ; 3 - Définir un message d'erreur personnalisé. Valeurs possibles si `type` est défini sur « Vérification des éléments non pris en charge » : 1 - Ignorer la valeur ; 2 - Définir une valeur personnalisée ; 3 - Définir un message d'erreur personnalisé. Comportement de la propriété : - *required* si `type` est défini sur « Multiplicateur personnalisé » (1), « Expression régulière » (5), « Conversion de booléen en décimal » (6), « Conversion d'octal en décimal » (7), « Conversion d'hexadécimal en décimal » (8), « Modification simple » (9), « Modification par seconde » (10), « XPath XML » (11), « JSONPath » (12), « Dans l'intervalle » (13), « Correspond à l'expression régulière » (14), « Ne correspond pas à l'expression régulière » (15), « Vérification des erreurs dans JSON » (16), « Vérification des erreurs dans XML » (17), « Vérification des erreurs à l'aide d'une expression régulière » (18), « Modèle Prometheus » (22), « Conversion de Prometheus en JSON » (23), « Conversion de CSV en JSON » (24), « Vérification des éléments non pris en charge » (26), « Conversion de XML en JSON » (27), « Valeur de parcours SNMP » (28), « Conversion du parcours SNMP en JSON » (29) ou « Valeur SNMP get » (30)
error_handler_params	string	Paramètres du gestionnaire d'erreurs. Comportement de la propriété : - *required* si `error_handler` est défini sur « Définir une valeur personnalisée » ou « Définir un message d'erreur personnalisé »

Les paramètres et gestionnaires d’erreurs suivants sont pris en charge pour chaque type de prétraitement.

Type de prétraitement	Nom	Paramètre 1	Paramètre 2	Paramètre 3	Gestionnaires d’erreurs pris en charge
1	Multiplication personnalisé	nombre^1, 6^			0, 1, 2, 3
2	Suppression des espaces à droite	liste de caractères[2]			
3	Suppression des espaces à gauche	liste de caractères[2]			
4	Suppression des espaces	liste de caractères[2]			
5	Expression régulière	modèle[3]	sortie[2]		0, 1, 2, 3
6	Conversion de booléen en décimal				0, 1, 2, 3
7	Conversion d’octal en décimal				0, 1, 2, 3
8	Conversion d’hexadécimal en décimal				0, 1, 2, 3
9	Modification simple				0, 1, 2, 3
10	Modification par seconde				0, 1, 2, 3
11	XPath XML	chemin[4]			0, 1, 2, 3
12	JSONPath	chemin[4]			0, 1, 2, 3
13	Dans l’intervalle	min^1, 6^	max^1, 6^		0, 1, 2, 3
14	Correspond à l’expression régulière	modèle[3]			0, 1, 2, 3

Type de prétraitement	Nom	Paramètre 1	Paramètre 2	Paramètre 3	Gestionnaires d’erreurs pris en charge
15	Ne correspond pas à l’expression régulière	modèle[3]			0, 1, 2, 3
16	Vérification des erreurs dans JSON	chemin[4]			0, 1, 2, 3
17	Vérification des erreurs dans XML	chemin[4]			0, 1, 2, 3
18	Vérification des erreurs à l’aide d’une expression régulière	modèle[3]	sortie[2]		0, 1, 2, 3
19	Ignorer les valeurs inchangées				
20	Ignorer les valeurs inchangées avec battement de cœur	secondes^5, 6^			
21	JavaScript	script[2]			
22	Modèle Prometheus	modèle^6, 7^	`value`, `label`, `function`	sortie^8, 9^	0, 1, 2, 3
23	Conversion de Prometheus en JSON	modèle^6, 7^			0, 1, 2, 3
24	Conversion de CSV en JSON	caractère[2]	caractère[2]	0,1	0, 1, 2, 3
25	Remplacement	chaîne recherchée[2]	remplacement[2]		

Type de prétraitement	Nom	Paramètre 1	Paramètre 2	Paramètre 3	Gestionnaires d'erreurs pris en charge
26	Vérification des éléments non pris en charge	Type[1]	modèle^3, 6^		1, 2, 3
27	Conversion de XML en JSON				0, 1, 2, 3
28	Valeur de parcours SNMP	OID[2]	Format : 0 - Inchangé 1 - UTF-8 depuis Hex-STRING 2 - MAC depuis Hex-STRING 3 - Entier depuis BITS		0, 1, 2, 3
29	Conversion du parcours SNMP en JSON[10]	Nom du champ[2]	Préfixe OID[2]	Format : 0 - Inchangé 1 - UTF-8 depuis Hex-STRING 2 - MAC depuis Hex-STRING 3 - Entier depuis BITS	0, 1, 2, 3
30	Valeur SNMP get	Format : 1 - UTF-8 depuis Hex-STRING 2 - MAC depuis Hex-STRING 3 - Entier depuis BITS			0, 1, 2, 3

[1] nombre entier ou à virgule flottante
 [2] chaîne
 [3] expression régulière
 [4] JSONPath ou XPath XML
 [5] entier positif (avec prise en charge des suffixes temporels, par exemple 30s, 1m, 2h, 1d)
 [6] macro utilisateur, macro LLD
 [7] modèle Prometheus suivant la syntaxe : `<metric name>{<label name>="<label value>", ...} == <value>`. Chaque composant du modèle Prometheus (métrique, nom du libellé, valeur du libellé et valeur de la métrique) peut être une macro utilisateur ou une macro LLD.
 [8] sortie Prometheus suivant la syntaxe : `<label name>` (peut être une macro utilisateur ou une macro LLD) si `label` est sélectionné comme deuxième paramètre.
 [9] L'une des fonctions d'agrégation suivantes : `sum`, `min`, `max`, `avg`, `count` si `function` est sélectionné comme deuxième paramètre.
 [10] Prend en charge plusieurs enregistrements « Nom du champ,Préfixe OID,Format » délimités par un caractère de nouvelle ligne.

itemprototype.create

Description

`object itemprototype.create(object/array itemPrototypes)`

Cette méthode vous permet de créer de nouveaux prototypes d'élément.

Note:
Cette méthode est disponible uniquement pour les types d'utilisateur *Admin* et *Super admin*. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres des rôles utilisateur. Consultez Rôles utilisateur pour plus d'informations.

Paramètres

`(object/array)` Prototype d'élément à créer.

En plus des propriétés standard du prototype d'élément, la méthode accepte les paramètres suivants.

Paramètre	Type	Description
ruleid	ID	ID de la règle LLD à laquelle l'élément appartient. Comportement du paramètre: - *obligatoire*
preprocessing	array	Options de prétraitement du prototype d'élément.
tags	array	Balises du prototype d'élément.

Valeurs retournées

`(object)` Renvoie un objet contenant les IDs des prototypes d'éléments créés sous la propriété `itemids`. L'ordre des ID renvoyés correspond à l'ordre des prototypes d'éléments transmis.

Exemples

Création d'un prototype d'élément

Créez un prototype d'élément pour surveiller l'espace disque libre sur un système de fichiers découvert. Les éléments découverts doivent être des éléments numériques de l'agent Zabbix mis à jour toutes les 30 secondes.

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "itemprototype.create",
    "params": {
        "name": "Espace disque libre sur {#FSNAME}",
        "key_": "vfs.fs.size[{#FSNAME},free]",
        "hostid": "10197",
        "ruleid": "27665",
        "type": 0,
        "value_type": 3,
        "interfaceid": "112",
        "delay": "30s"
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "27666"
        ]
    },
    "id": 1
}
```

Création d'un prototype d'élément avec prétraitement

Créez un élément en utilisant la variation par seconde et un multiplicateur personnalisé comme deuxième étape.

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "itemprototype.create",
    "params": {
        "name": "Trafic réseau entrant sur {#IFNAME}",
        "key_": "net.if.in[{#IFNAME}]",
        "hostid": "10001",
        "ruleid": "27665",
        "type": 0,
        "value_type": 3,
        "delay": "60s",
        "units": "bps",
        "interfaceid": "1155",
        "preprocessing": [
            {
                "type": 10,
                "params": "",
                "error_handler": 0,
                "error_handler_params": ""
            },
            {
                "type": 1,
                "params": "8",
                "error_handler": 2,
                "error_handler_params": "10"
            }
        ]
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "44211"
        ]
    },
    "id": 1
}
```

Création d'un prototype d'élément dépendant

Créez un prototype d'élément dépendant pour le prototype d'élément maître avec l'ID 44211. Seules les dépendances sur le même hôte (modèle/règle de découverte) sont autorisées, par conséquent, l'élément maître et l'élément dépendant doivent avoir le même hostid et le même ruleid.

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "itemprototype.create",
    "params": {
      "hostid": "10001",
      "ruleid": "27665",
      "name": "Dependent test item prototype",
      "key_": "dependent.prototype",
      "type": 18,
      "master_itemid": "44211",
      "value_type": 3
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "44212"
        ]
    },
    "id": 1
}
```

Créer un prototype d'élément d'agent HTTP

Créez un prototype d'élément avec une URL utilisant une macro utilisateur, des champs de requête et des en-têtes personnalisés.

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "itemprototype.create",
    "params": {
        "type": "19",
        "hostid": "10254",
        "ruleid": "28256",
        "interfaceid": "2",
        "name": "exemple de prototype d'élément API",
        "key_": "api_http_item",
        "value_type": 3,
        "url": "{$URL_PROTOTYPE}",
        "query_fields": [
            {
                "name": "min",
                "value": "10"
            },
            {
                "name": "max",
                "value" "100"
            }
        ],
        "headers": [
            {
                "name": "X-Source",
                "value": "api"
            }
        ],
        "delay": "35"
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "28305"
        ]
    },
    "id": 1
}
```

Créer un prototype d'élément de script

Créez une collecte de données simple à l'aide d'un prototype d'élément de script.

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "itemprototype.create",
    "params": {
        "name": "Script example",
        "key_": "custom.script.itemprototype",
        "hostid": "12345",
        "type": 21,
        "value_type": 4,
        "params": "var request = new HttpRequest();\nreturn request.post(\"https://postman-echo.com/post\"
        "parameters": [
            {
                "name": "host",
                "value": "{HOST.CONN}"
            }
        ],
        "timeout": "6s",
        "delay": "30s"
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "23865"
        ]
    },
    "id": 1
}
```

Source

CItemPrototype::create() dans *ui/include/classes/api/services/CItemPrototype.php*.

itemprototype.delete

Description

`object itemprototype.delete(array itemPrototypeIds)`

Cette méthode vous permet de supprimer des prototypes d'élément.

> **Note:**
> Cette méthode est disponible uniquement pour les types d'utilisateur *Admin* et *Super admin*. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres des rôles utilisateur. Consultez Rôles utilisateur pour plus d'informations.

Paramètres

`(array)` ID des prototypes d'éléments à supprimer.

Valeurs retournées

`(object)` Renvoie un objet contenant les identifiants des prototypes d'éléments supprimés sous la propriété `prototypeids`.

Exemples

Suppression de plusieurs prototypes d'élément

Supprimez deux prototypes d'élément.
 Les prototypes d'élément dépendants sont supprimés automatiquement si l'élément maître ou le prototype d'élément est supprimé.

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "itemprototype.delete",
    "params": [
        "27352",
        "27356"
    ],
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "prototypeids": [
            "27352",
            "27356"
        ]
    },
    "id": 1
}
```

Source

CItemPrototype::delete() dans *ui/include/classes/api/services/CItemPrototype.php*.

itemprototype.get

Description

`integer/array itemprototype.get(object parameters)`

La méthode permet de récupérer les prototypes d'élément selon les paramètres fournis.

Note:
Cette méthode est disponible pour les utilisateurs de tout type. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres des rôles utilisateur. Voir Rôles utilisateur pour plus d'informations.

Paramètres

`(object)` Paramètres définissant la sortie souhaitée.

La méthode prend en charge les paramètres suivants.

Paramètre	Type	Description
discoveryids	ID/array	Retourner uniquement les prototypes d'élément qui appartiennent aux règles LLD données.
graphids	ID/array	Retourner uniquement les prototypes d'élément utilisés dans les prototypes de graphique donnés.
hostids	ID/array	Retourner uniquement les prototypes d'élément qui appartiennent aux hôtes donnés.
inherited	boolean	Si défini sur `true`, retourner uniquement les prototypes d'élément hérités d'un modèle.
itemids	ID/array	Retourner uniquement les prototypes d'élément avec les ID donnés.
monitored	boolean	Si défini sur `true`, retourner uniquement les prototypes d'élément activés qui appartiennent à des hôtes surveillés.
templated	boolean	Si défini sur `true`, retourner uniquement les prototypes d'élément qui appartiennent à des modèles.
templateids	ID/array	Retourner uniquement les prototypes d'élément qui appartiennent aux modèles donnés.
triggerids	ID/array	Retourner uniquement les prototypes d'élément utilisés dans les prototypes de déclencheur donnés.

Paramètre	Type	Description
selectDiscoveryData	query	Retourner une propriété `discoveryData` avec les données de l'objet de découverte du prototype d'élément. L'objet de découverte du prototype d'élément relie un prototype d'élément découvert à un prototype d'élément à partir duquel il a été découvert. Il possède les propriétés suivantes : `parent_itemid` - `(string)` ID du prototype d'élément à partir duquel le prototype d'élément a été créé ; `key_` - `(string)` clé du prototype d'élément ; `status` - `(int)` statut de découverte du prototype d'élément : 0 - *(par défaut)* le prototype d'élément est découvert, 1 - le prototype d'élément n'est plus découvert ; `ts_delete` - `(timestamp)` heure à laquelle un prototype d'élément qui n'est plus découvert sera supprimé ; `ts_disable` - `(timestamp)` heure à laquelle un prototype d'élément qui n'est plus découvert sera désactivé ; `disable_source` - `(int)` indicateur précisant si le prototype d'élément a été désactivé par une règle LLD ou manuellement : 0 - *(par défaut)* désactivé automatiquement, 1 - désactivé par une règle LLD.
selectDiscoveryRule	query	Retourner une propriété `discoveryRule` avec la règle de découverte de bas niveau à laquelle appartient le prototype d'élément.
selectDiscoveryRulePrototype	query	Retourner une propriété `discoveryRulePrototype` avec le prototype de règle LLD parent auquel appartient le prototype d'élément.
selectGraphs	query	Retourner une propriété `graphs` avec les prototypes de graphique dans lesquels le prototype d'élément est utilisé. Prend en charge `count`.
selectHosts	query	Retourner une propriété `hosts` avec un tableau des hôtes auxquels appartient le prototype d'élément.
selectInheritedTags	query	Retourner une propriété `inheritedTags` avec les tags présents sur le modèle/l'hôte/les modèles liés.
selectTags	query	Retourner les tags du prototype d'élément dans la propriété `tags`.
selectTriggers	query	Retourner une propriété `triggers` avec les prototypes de déclencheur dans lesquels le prototype d'élément est utilisé. Prend en charge `count`.
selectPreprocessing	query	Retourner une propriété `preprocessing` avec les options de prétraitement du prototype d'élément.
selectValueMap	query	Retourner une propriété `valuemap` avec la table de correspondance des valeurs du prototype d'élément.
filter	object	Retourner uniquement les résultats qui correspondent exactement au filtre donné. Accepte un objet où les clés sont des noms de propriété et les valeurs sont soit une valeur unique, soit un tableau de valeurs à comparer. Ne prend pas en charge les propriétés de type de données `text` data type. Prend en charge des propriétés supplémentaires : `host` - nom technique de l'hôte auquel appartient le prototype d'élément.
limitSelects	integer	Limite le nombre d'enregistrements renvoyés par les sous-sélections. S'applique aux sous-sélections suivantes : `selectGraphs` - les résultats seront triés par `name` ; `selectTriggers` - les résultats seront triés par `description`.

Paramètre	Type	Description
sortfield	string/array	Trier le résultat selon les propriétés données. Valeurs possibles : `itemid`, `name`, `key_`, `delay`, `type`, `status`, `history`, `trends`, `discovered`.
countOutput	boolean	Ces paramètres sont décrits dans le commentaire de référence.
editable	boolean	
excludeSearch	boolean	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Valeurs retournées

`(integer/array)` Retourne soit :

- un tableau d'objets ;
- le nombre d'objets récupérés, si le paramètre `countOutput` a été utilisé.

Exemples

Récupération des prototypes d'éléments depuis une règle LLD

Récupérez tous les prototypes d'éléments pour un ID de règle LLD spécifique.

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "itemprototype.get",
    "params": {
        "output": "extend",
        "discoveryids": "27426"
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "itemid": "23077",
            "type": "0",
            "snmp_oid": "",
            "hostid": "10079",
            "name": "Incoming network traffic on en0",
            "key_": "net.if.in[en0]",
            "delay": "1m",
            "history": "1w",
            "trends": "365d",
            "status": "0",
            "value_type": "3",
            "trapper_hosts": "",
            "units": "bps",
            "formula": "",
            "logtimefmt": "",
            "templateid": "0",
            "valuemapid": "0",
            "params": "",
```

```
        "ipmi_sensor": "",
        "authtype": "0",
        "username": "",
        "password": "",
        "publickey": "",
        "privatekey": "",
        "flags": "2",
        "interfaceid": "0",
        "description": "",
        "inventory_link": "0",
        "lifetime": "7d",
        "evaltype": "0",
        "jmx_endpoint": "",
        "master_itemid": "0",
        "timeout": "",
        "url": "",
        "query_fields": [],
        "posts": "",
        "status_codes": "200",
        "follow_redirects": "1",
        "post_type": "0",
        "http_proxy": "",
        "headers": [],
        "retrieve_mode": "0",
        "request_method": "0",
        "output_format": "0",
        "ssl_cert_file": "",
        "ssl_key_file": "",
        "ssl_key_password": "",
        "verify_peer": "0",
        "verify_host": "0",
        "allow_traps": "0",
        "discover": "0",
        "uuid": "",
        "lifetime_type": "0",
        "enabled_lifetime_type": "2",
        "enabled_lifetime": "0",
        "parameters": []
    },
    {
        "itemid": "10010",
        "type": "0",
        "snmp_oid": "",
        "hostid": "10001",
        "name": "Processor load (1 min average per core)",
        "key_": "system.cpu.load[percpu,avg1]",
        "delay": "1m",
        "history": "1w",
        "trends": "365d",
        "status": "0",
        "value_type": "0",
        "trapper_hosts": "",
        "units": "",
        "formula": "",
        "logtimefmt": "",
        "templateid": "0",
        "valuemapid": "0",
        "params": "",
        "ipmi_sensor": "",
        "authtype": "0",
        "username": "",
        "password": "",
```

```
            "publickey": "",
            "privatekey": "",
            "flags": "2",
            "interfaceid": "0",
            "description": "The processor load is calculated as system CPU load divided by number of CPU c
            "inventory_link": "0",
            "lifetime": "7d",
            "evaltype": "0",
            "jmx_endpoint": "",
            "master_itemid": "0",
            "timeout": "",
            "url": "",
            "query_fields": [],
            "posts": "",
            "status_codes": "200",
            "follow_redirects": "1",
            "post_type": "0",
            "http_proxy": "",
            "headers": [],
            "retrieve_mode": "0",
            "request_method": "0",
            "output_format": "0",
            "ssl_cert_file": "",
            "ssl_key_file": "",
            "ssl_key_password": "",
            "verify_peer": "0",
            "verify_host": "0",
            "allow_traps": "0",
            "discover": "0",
            "uuid": "",
            "lifetime_type": "0",
            "enabled_lifetime_type": "2",
            "enabled_lifetime": "0",
            "parameters": []
        }
    ],
    "id": 1
}
```


Recherche d'un élément dépendant

Trouver un élément dépendant pour un ID d'élément spécifique.

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "itemprototype.get",
    "params": {
        "output": "extend",
        "filter": {
            "type": 18,
            "master_itemid": "25545"
        },
        "limit": "1"
    },
    "id": 1
}
```

Réponse :


```
{
    "jsonrpc": "2.0",
    "result": [
        {
```

```
            "itemid": "25547",
            "type": "18",
            "snmp_oid": "",
            "hostid": "10116",
            "name": "Seconds",
            "key_": "apache.status.uptime.seconds",
            "delay": "0",
            "history": "90d",
            "trends": "365d",
            "status": "0",
            "value_type": "3",
            "trapper_hosts": "",
            "units": "",
            "formula": "",
            "logtimefmt": "",
            "templateid": "0",
            "valuemapid": "0",
            "params": "",
            "ipmi_sensor": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "flags": "0",
            "interfaceid": "0",
            "description": "",
            "inventory_link": "0",
            "lifetime": "7d",
            "evaltype": "0",
            "jmx_endpoint": "",
            "master_itemid": "25545",
            "timeout": "",
            "url": "",
            "query_fields": [],
            "posts": "",
            "status_codes": "200",
            "follow_redirects": "1",
            "post_type": "0",
            "http_proxy": "",
            "headers": [],
            "retrieve_mode": "0",
            "request_method": "0",
            "output_format": "0",
            "ssl_cert_file": "",
            "ssl_key_file": "",
            "ssl_key_password": "",
            "verify_peer": "0",
            "verify_host": "0",
            "allow_traps": "0",
            "discover": "0",
            "uuid": "",
            "lifetime_type": "0",
            "enabled_lifetime_type": "2",
            "enabled_lifetime": "0",
            "parameters": []
        }
    ],
    "id": 1
}
```


Rechercher un prototype d'élément d'agent HTTP

Recherchez un prototype d'élément d'agent HTTP avec la méthode de requête HEAD pour un ID d'hôte spécifique.

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "itemprototype.get",
    "params": {
        "hostids": "10254",
        "filter": {
            "type": 19,
            "request_method": 3
        }
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "itemid": "28257",
            "type": "19",
            "snmp_oid": "",
            "hostid": "10254",
            "name": "discovered",
            "key_": "item[{#INAME}]",
            "delay": "{#IUPDATE}",
            "history": "90d",
            "trends": "30d",
            "status": "0",
            "value_type": "3",
            "trapper_hosts": "",
            "units": "",
            "formula": "",
            "logtimefmt": "",
            "templateid": "28255",
            "valuemapid": "0",
            "params": "",
            "ipmi_sensor": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "flags": "2",
            "interfaceid": "2",
            "description": "",
            "inventory_link": "0",
            "lifetime": "7d",
            "evaltype": "0",
            "jmx_endpoint": "",
            "master_itemid": "0",
            "timeout": "",
            "url": "{#IURL}",
            "query_fields": [],
            "posts": "",
            "status_codes": "",
            "follow_redirects": "0",
            "post_type": "0",
            "http_proxy": "",
            "headers": [],
            "retrieve_mode": "0",
```

```
            "request_method": "3",
            "output_format": "0",
            "ssl_cert_file": "",
            "ssl_key_file": "",
            "ssl_key_password": "",
            "verify_peer": "0",
            "verify_host": "0",
            "allow_traps": "0",
            "discover": "0",
            "uuid": "",
            "lifetime_type": "0",
            "enabled_lifetime_type": "2",
            "enabled_lifetime": "0",
            "parameters": []
        }
    ],
    "id": 1
}
```


Voir aussi

- Hôte
- Prototype de graphique
- Prototype de déclencheur

Source

CItemPrototype::get() dans *ui/include/classes/api/services/CItemPrototype.php*.

itemprototype.update

Description

`object itemprototype.update(object/array itemPrototypes)`

Cette méthode vous permet de mettre à jour des prototypes d'élément existants.

Note:
Cette méthode est disponible uniquement pour les types d'utilisateur *Admin* et *Super admin*. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres des rôles utilisateur. Consultez Rôles utilisateur pour plus d'informations.

Paramètres

`(object/array)` Propriétés du prototype d'élément à mettre à jour.

La propriété `itemid` doit être définie pour chaque prototype d'élément, toutes les autres propriétés sont facultatives. Seules les propriétés transmises seront mises à jour, toutes les autres resteront inchangées.

En plus des propriétés standard du prototype d'élément, la méthode accepte les paramètres suivants.

Paramètre	Type	Description
preprocessing	array	Options de prétraitement du prototype d'élément pour remplacer les options de prétraitement actuelles. Comportement des paramètres: - *lecture seule* pour les objets hérités
tags	array	Tags du prototype d'élément.

Valeurs retournées

`(object)` Retourne un objet contenant les identifiants des prototypes d'élément mis à jour sous la propriété `itemids`.

Exemples

Modification de l'interface d'un prototype d'élément

Modifiez l'interface de l'hôte qui sera utilisée par les éléments découverts.

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "itemprototype.update",
    "params": {
        "itemid": "27428",
        "interfaceid": "132"
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "27428"
        ]
    },
    "id": 1
}
```

Mettre à jour un prototype d'élément dépendant

Mettez à jour le prototype d'élément dépendant avec un nouvel ID de prototype d'élément maître. Seules les dépendances sur le même hôte (modèle/règle de découverte) sont autorisées ; par conséquent, l'élément maître et l'élément dépendant doivent avoir le même hostid et le même ruleid.

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "itemprototype.update",
    "params": {
        "master_itemid": "25570",
        "itemid": "189030"
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "189030"
        ]
    },
    "id": 1
}
```

Mettre à jour un prototype d'élément d'agent HTTP

Modifier les champs de requête et supprimer tous les en-têtes personnalisés.

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "itemprototype.update",
    "params": {
        "itemid":"28305",
        "query_fields": [
            {
```

```
                "name": "random",
                "value": "qwertyuiopasdfghjklzxcvbnm"
            }
        ],
        "headers": []
    }
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "28305"
        ]
    },
    "id": 1
}
```

Mise à jour des options de prétraitement d'un élément

Mettez à jour un prototype d'élément avec la règle de prétraitement d'élément « Multiplicateur personnalisé ».

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "itemprototype.update",
    "params": {
        "itemid": "44211",
        "preprocessing": [
            {
                "type": 1,
                "params": "4",
                "error_handler": 2,
                "error_handler_params": "5"
            }
        ]
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "44211"
        ]
    },
    "id": 1
}
```

Mise à jour d'un prototype d'élément de script

Mettez à jour un prototype d'élément de script avec un script différent et supprimez les paramètres inutiles qui étaient utilisés par le script précédent.

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "itemprototype.update",
    "params": {
        "itemid": "23865",
```

```
        "parameters": [],
        "script": "Zabbix.log(3, 'Log test');\nreturn 1;"
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "23865"
        ]
    },
    "id": 1
}
```

Source

CItemPrototype::update() dans *ui/include/classes/api/services/CItemPrototype.php*.

Prototype de déclencheur

Cette classe est conçue pour fonctionner avec des prototypes de déclencheur.

Références d'objet :

- Prototype de déclencheur
- Tag de prototype de déclencheur

Méthodes disponibles :

- triggerprototype.create - créer de nouveaux prototypes de déclencheur
- triggerprototype.delete - supprimer des prototypes de déclencheur
- triggerprototype.get - récupérer des prototypes de déclencheur
- triggerprototype.update - mettre à jour des prototypes de déclencheur

Objet prototype de déclencheur

Les objets suivants sont directement liés à l'API `triggerprototype`.

Prototype de déclencheur

L'objet prototype de déclencheur possède les propriétés suivantes.

Propriété	Type	Description
triggerid	ID	ID du prototype de déclencheur. Comportement de la propriété : - *lecture seule* - *obligatoire* pour les opérations de mise à jour
description	string	Nom du prototype de déclencheur. Comportement de la propriété : - *obligatoire* pour les opérations de création
expression	string	Expression du déclencheur. Doit contenir au moins un prototype d'élément. Les prototypes d'élément avec `value_type` défini sur 5 (binaire) ou 6 (JSON) ne peuvent pas être utilisés dans les expressions de déclencheur. Comportement de la propriété : - *obligatoire* pour les opérations de création

Propriété	Type	Description
event_name	string	Nom de l'événement généré par le déclencheur.
opdata	string	Données opérationnelles.
comments	string	Commentaires supplémentaires sur le prototype de déclencheur.
priority	integer	Gravité du prototype de déclencheur. Valeurs possibles : 0 - *(par défaut)* non classé ; 1 - information ; 2 - avertissement ; 3 - moyen ; 4 - élevé ; 5 - désastre.
status	integer	Indique si le prototype de déclencheur est activé ou désactivé. Valeurs possibles : 0 - *(par défaut)* activé ; 1 - désactivé.
flags	integer	Origine du prototype de déclencheur. Valeurs possibles : 2 - un prototype de déclencheur ; 6 - un prototype de déclencheur découvert. Comportement de la propriété : - *lecture seule*
templateid	ID	ID du prototype de déclencheur du modèle parent. Comportement de la propriété : - *lecture seule*
type	integer	Indique si le prototype de déclencheur peut générer plusieurs événements de problème. Valeurs possibles : 0 - *(par défaut)* ne pas générer plusieurs événements ; 1 - générer plusieurs événements.
url	string	URL associée au prototype de déclencheur.
url_name	string	Libellé de l'URL associée au prototype de déclencheur.
recovery_mode	integer	Mode de génération de l'événement OK. Valeurs possibles : 0 - *(par défaut)* Expression ; 1 - Expression de récupération ; 2 - Aucun.
recovery_expression	string	Expression de récupération du déclencheur. Doit contenir au moins un prototype d'élément. Les prototypes d'élément avec `value_type` défini sur 5 (binaire) ou 6 (JSON) ne peuvent pas être utilisés dans les expressions de récupération du déclencheur.
correlation_mode	integer	Fermeture de l'événement OK. Valeurs possibles : 0 - *(par défaut)* Tous les problèmes ; 1 - Tous les problèmes si les valeurs des tags correspondent.
correlation_tag	string	Tag pour la correspondance.
manual_close	integer	Autoriser la fermeture manuelle. Valeurs possibles : 0 - *(par défaut)* Non ; 1 - Oui.

Propriété	Type	Description
discover	integer	Statut de découverte du prototype de déclencheur. Valeurs possibles : 0 - *(par défaut)* les nouveaux déclencheurs seront découverts ; 1 - les nouveaux déclencheurs ne seront pas découverts et les déclencheurs existants seront marqués comme perdus.
uuid	string	Identifiant unique universel, utilisé pour lier les prototypes de déclencheur importés à ceux qui existent déjà. Généré automatiquement s'il n'est pas fourni. Comportement de la propriété : - *pris en charge* si le prototype de déclencheur appartient à un modèle

Balise de prototype de déclencheur

L'objet de balise de prototype de déclencheur possède les propriétés suivantes.

Propriété	Type	Description
tag	string	Nom de la balise de prototype de déclencheur. Comportement de la propriété: - *obligatoire*
value	string	Valeur de la balise de prototype de déclencheur. Comportement de la propriété: - *lecture seule*.
object	integer	Type d'objet dont la balise a été héritée. Valeurs possibles : 0 - modèle; 1 - hôte; 3 - élément; 4 - prototype d'élément. Comportement de la propriété: - *lecture seule*.
objectid	ID	ID de l'objet dont la balise a été héritée. Comportement de la propriété: - *lecture seule*.

triggerprototype.create

Description

`object triggerprototype.create(object/array triggerPrototypes)`

Cette méthode vous permet de créer de nouveaux modèles de déclencheur.

> **Note:**
> Cette méthode est disponible uniquement pour les types d'utilisateur *Admin* et *Super admin*. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres des rôles utilisateur. Consultez Rôles utilisateur pour plus d'informations.

Paramètres

`(object/array)` Prototypes de déclencheur à créer.

En plus des propriétés standard des prototypes de déclencheur, la méthode accepte les paramètres suivants.

Paramètre	Type	Description
dependencies	array	Déclencheurs et prototypes de déclencheur dont dépend le prototype de déclencheur. Pour les déclencheurs, seule la propriété `triggerid` doit être définie.
tags	array	Tags de prototype de déclencheur.

Valeurs retournées

`(object)` Renvoie un objet contenant les IDs des prototypes de déclencheurs créés sous la propriété `itemids`. L'ordre des ID renvoyés correspond à l'ordre des prototypes de déclencheurs transmis.

Exemples

Création d'un prototype de déclencheur

Créez un prototype de déclencheur pour détecter lorsqu'un système de fichiers a moins de 20 % d'espace disque libre.

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "triggerprototype.create",
    "params": {
        "description": "L'espace disque libre est inférieur à 20 % sur le volume {#FSNAME}",
        "expression": "last(/Zabbix server/vfs.fs.size[{#FSNAME},pfree])<20",
        "tags": [
            {
                "tag": "volume",
                "value": "{#FSNAME}"
            },
            {
                "tag": "type",
                "value": "{#FSTYPE}"
            }
        ]
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "triggerids": [
            "17372"
        ]
    },
    "id": 1
}
```

Source

CTriggerPrototype::create() dans *ui/include/classes/api/services/CTriggerPrototype.php*.

triggerprototype.delete

Description

`object triggerprototype.delete(array triggerPrototypeIds)`

Cette méthode vous permet de supprimer des prototypes de déclencheur.

Note:
Cette méthode est disponible uniquement pour les types d'utilisateur *Admin* et *Super admin*. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres des rôles utilisateur. Consultez Rôles utilisateur pour plus d'informations.

Paramètres

`(array)` ID des prototypes de déclencheur à supprimer.

Valeurs de retour

`(object)` Retourne un objet contenant les identifiants des prototypes de déclencheur supprimés sous la propriété `triggerids`.

Exemples

Suppression de plusieurs prototypes de déclencheur

Supprimer deux prototypes de déclencheur.

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "triggerprototype.delete",
    "params": [
        "12002",
        "12003"
    ],
    "auth": "3a57200802b24cda67c4e4010b50c065",
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "triggerids": [
            "12002",
            "12003"
        ]
    },
    "id": 1
}
```

Source

CTriggerPrototype::delete() in *ui/include/classes/api/services/CTriggerPrototype.php*.

triggerprototype.get

Description

`integer/array triggerprototype.get(object parameters)`

Cette méthode permet de récupérer les prototypes de déclencheur selon les paramètres fournis.

Note:
Cette méthode est disponible pour les utilisateurs de tout type. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres du rôle utilisateur. Consultez Rôles utilisateur pour plus d'informations.

Paramètres

`(object)` Paramètres définissant la sortie souhaitée.

La méthode prend en charge les paramètres suivants.

Paramètre	Type	Description
active	flag	Retourner uniquement les prototypes de déclencheur activés qui appartiennent à des hôtes surveillés.
discoveryids	ID/array	Retourner uniquement les prototypes de déclencheur qui appartiennent aux règles LLD données.
functions	string/array	Retourner uniquement les déclencheurs qui utilisent les fonctions données. Consultez la page Fonctions prises en charge pour obtenir la liste des fonctions prises en charge.
group	string	Retourner uniquement les prototypes de déclencheur qui appartiennent à des hôtes ou des modèles des groupes d'hôtes ou groupes de modèles portant le nom donné.
groupids	ID/array	Retourner uniquement les prototypes de déclencheur qui appartiennent à des hôtes ou des modèles des groupes d'hôtes ou groupes de modèles donnés.
host	string	Retourner uniquement les prototypes de déclencheur qui appartiennent à des hôtes portant le nom donné.
hostids	ID/array	Retourner uniquement les prototypes de déclencheur qui appartiennent aux hôtes donnés.
inherited	boolean	Si défini sur `true`, retourner uniquement les prototypes de déclencheur hérités d'un modèle.
maintenance	boolean	Si défini sur `true`, retourner uniquement les prototypes de déclencheur activés qui appartiennent à des hôtes en maintenance.
min_severity	integer	Retourner uniquement les prototypes de déclencheur avec une sévérité supérieure ou égale à la sévérité donnée.
monitored	flag	Retourner uniquement les prototypes de déclencheur activés qui appartiennent à des hôtes surveillés et ne contiennent que des éléments activés.
templated	boolean	Si défini sur `true`, retourner uniquement les prototypes de déclencheur qui appartiennent à des modèles.
templateids	ID/array	Retourner uniquement les prototypes de déclencheur qui appartiennent aux modèles donnés.
triggerids	ID/array	Retourner uniquement les prototypes de déclencheur avec les ID donnés.
expandExpression	flag	Développer les fonctions et macros dans l'expression du déclencheur.
selectDependencies	query	Retourner les prototypes de déclencheur et les déclencheurs dont dépend le prototype de déclencheur dans la propriété `dependencies`.
selectDiscoveryData	query	Retourner une propriété `discoveryData` avec les données de l'objet de découverte du prototype de déclencheur. L'objet de découverte du prototype de déclencheur relie un prototype de déclencheur découvert à un prototype de déclencheur à partir duquel il a été découvert. Il possède les propriétés suivantes : `parent_triggerid` - `(ID)` ID du prototype de déclencheur à partir duquel le prototype de déclencheur a été créé ; `status` - `(int)` statut de découverte du prototype de déclencheur : 0 - *(par défaut)* le prototype de déclencheur est découvert, 1 - le prototype de déclencheur n'est plus découvert ; `ts_delete` - `(timestamp)` heure à laquelle un prototype de déclencheur qui n'est plus découvert sera supprimé ; `ts_disable` - `(timestamp)` heure à laquelle un prototype de déclencheur qui n'est plus découvert sera désactivé ; `disable_source` - `(int)` indicateur précisant si le prototype de déclencheur a été désactivé par une règle LLD ou manuellement : 0 - *(par défaut)* désactivé automatiquement, 1 - désactivé par une règle LLD.
selectDiscoveryRule	query	Retourner la règle LLD à laquelle appartient le prototype de déclencheur dans la propriété `discoveryRule`.
selectDiscoveryRulePrototype	query	Retourner une propriété `discoveryRulePrototype` avec le prototype de règle LLD parent auquel appartient le prototype de déclencheur.

Paramètre	Type	Description
selectFunctions	query	Retourner les fonctions utilisées dans le prototype de déclencheur dans la propriété `functions`. Les objets de fonction représentent les fonctions utilisées dans l'expression du déclencheur et possèdent les propriétés suivantes : `functionid` - `(ID)` ID de la fonction ; `itemid` - `(ID)` ID de l'élément utilisé dans la fonction ; `function` - `(string)` nom de la fonction ; `parameter` - `(string)` paramètre transmis à la fonction. Le paramètre de requête est remplacé par le symbole $ dans la chaîne retournée.
selectHostGroups	query	Retourner les groupes d'hôtes auxquels appartient le prototype de déclencheur dans la propriété `hostgroups`.
selectHosts	query	Retourner les hôtes auxquels appartient le prototype de déclencheur dans la propriété `hosts`.
selectInheritedTags	query	Retourner une propriété `inheritedTags` avec les tags hérités des modèles, des modèles liés et des hôtes, ainsi que des prototypes d'élément référencés dans les expressions de déclencheur ou de récupération.
selectItems	query	Retourner les éléments et prototypes d'élément utilisés par le prototype de déclencheur dans la propriété `items`.
selectTags	query	Retourner les tags du prototype de déclencheur dans la propriété `tags`.
selectTemplateGroups	query	Retourner les groupes de modèles auxquels appartient le prototype de déclencheur dans la propriété `templategroups`.
filter	object	Retourner uniquement les résultats qui correspondent exactement au filtre donné. Accepte un objet où les clés sont des noms de propriété et les valeurs sont soit une valeur unique, soit un tableau de valeurs à faire correspondre. Ne prend pas en charge les propriétés du type de données `text`. Prend en charge des propriétés supplémentaires : `host` - nom technique de l'hôte auquel appartient le prototype de déclencheur ; `hostid` - ID de l'hôte auquel appartient le prototype de déclencheur.
limitSelects	integer	Limite le nombre d'enregistrements retournés par les sous-sélections. S'applique aux sous-sélections suivantes : `selectHosts` - les résultats seront triés par `host`.
sortfield	string/array	Trier le résultat selon les propriétés données. Valeurs possibles : `triggerid`, `description`, `status`, `priority`, `discovered`.
countOutput	boolean	Ces paramètres sont décrits dans le commentaire de référence.
editable	boolean	
excludeSearch	boolean	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Valeurs de retour

`(integer/array)` Renvoie soit :

- un tableau d'objets ;
- le nombre d'objets récupérés, si le paramètre `countOutput` a été utilisé.

Exemples

Récupérer les prototypes de déclencheurs à partir d'une règle LLD

Récupérez tous les prototypes de déclencheurs et leurs fonctions à partir d'une règle LLD.

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "triggerprototype.get",
    "params": {
        "output": "extend",
        "selectFunctions": "extend",
        "discoveryids": "22450"
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "triggerid": "13272",
            "expression": "{12598}<20",
            "description": "Le nombre d'inodes libres est inférieur à 20 % sur le volume {#FSNAME}",
            "url": "",
            "status": "0",
            "value": "0",
            "priority": "2",
            "lastchange": "0",
            "comments": "",
            "error": "",
            "templateid": "0",
            "type": "0",
            "state": "0",
            "flags": "2",
            "recovery_mode": "0",
            "recovery_expression": "",
            "correlation_mode": "0",
            "correlation_tag": "",
            "manual_close": "0",
            "opdata": "",
            "discover": "0",
            "event_name": "",
            "uuid": "6ce467d05e8745409a177799bed34bb3",
            "url_name": "",
            "functions": [
                {
                    "functionid": "12598",
                    "itemid": "22454",
                    "triggerid": "13272",
                    "parameter": "$",
                    "function": "last"
                }
            ]
        },
        {
            "triggerid": "13266",
            "expression": "{13500}<20",
            "description": "L'espace disque libre est inférieur à 20 % sur le volume {#FSNAME}",
```

```
            "url": "",
            "status": "0",
            "value": "0",
            "priority": "2",
            "lastchange": "0",
            "comments": "",
            "error": "",
            "templateid": "0",
            "type": "0",
            "state": "0",
            "flags": "2",
            "recovery_mode": "0",
            "recovery_expression": "",
            "correlation_mode": "0",
            "correlation_tag": "",
            "manual_close": "0",
            "opdata": "",
            "discover": "0",
            "event_name": "",
            "uuid": "74a1fc62bfe24b7eabe4e244c70dc384",
            "url_name": "",
            "functions": [
                {
                    "functionid": "13500",
                    "itemid": "22686",
                    "triggerid": "13266",
                    "parameter": "$",
                    "function": "last"
                }
            ]
        }
    ],
    "id": 1
}
```


Récupération d'un prototype de déclencheur spécifique avec des tags

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "triggerprototype.get",
    "params": {
        "output": [
            "triggerid",
            "description"
        ],
        "selectTags": "extend",
        "triggerids": [
            "17373"
        ]
    },
    "id": 1
}
```

Réponse :


```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "triggerid": "17373",
            "description": "L'espace disque libre est inférieur à 20 % sur le volume {#FSNAME}",
            "tags": [
                {
```

```
                    "tag": "volume",
                    "value": "{#FSNAME}"
                },
                {
                    "tag": "type",
                    "value": "{#FSTYPE}"
                }
            ]
        }
    ],
    "id": 1
}
```

Voir aussi

- Règle de découverte
- Élément
- Hôte
- Groupe d'hôtes
- Groupe de modèles

Source

CTriggerPrototype::get() dans *ui/include/classes/api/services/CTriggerPrototype.php*.

triggerprototype.update

Description

`object triggerprototype.update(object/array triggerPrototypes)`

Cette méthode vous permet de mettre à jour des prototypes de déclencheur existants.

> **Note:**
> Cette méthode est disponible uniquement pour les types d'utilisateur *Admin* et *Super admin*. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres des rôles utilisateur. Consultez Rôles utilisateur pour plus d'informations.

Paramètres

`(object/array)` Propriétés du prototype de déclencheur à mettre à jour.

La propriété `triggerid` doit être définie pour chaque prototype de déclencheur, toutes les autres propriétés sont facultatives. Seules les propriétés transmises seront mises à jour, toutes les autres resteront inchangées.

En plus des propriétés standard du prototype de déclencheur, la méthode accepte les paramètres suivants.

Paramètre	Type	Description
dependencies	array	Déclencheurs et prototypes de déclencheur dont dépend le prototype de déclencheur. Pour les déclencheurs, seule la propriété `triggerid` doit être définie.
tags	array	Tags de prototype de déclencheur.

Valeurs retournées

`(object)` Retourne un objet contenant les identifiants des prototypes de déclencheur mis à jour sous la propriété `triggerids`.

Exemples

Activation d'un prototype de déclencheur

Activez un prototype de déclencheur, c'est-à-dire définissez son statut sur « 0 ».

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "triggerprototype.update",
    "params": {
        "triggerid": "13938",
        "status": 0
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "triggerids": [
            "13938"
        ]
    },
    "id": 1
}
```

Remplacement des tags de prototype de déclencheur

Remplace les tags d'un prototype de déclencheur.

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "triggerprototype.update",
    "params": {
        "triggerid": "17373",
        "tags": [
            {
                "tag": "volume",
                "value": "{#FSNAME}"
            },
            {
                "tag": "type",
                "value": "{#FSTYPE}"
            }
        ]
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "triggerids": [
            "17373"
        ]
    },
    "id": 1
}
```

Source

CTriggerPrototype::update() dans *ui/include/classes/api/services/CTriggerPrototype.php*.

Prototype de graphique

Cette classe est conçue pour fonctionner avec les prototypes de graphiques.

Références d'objet :

- Prototype de graphique

Méthodes disponibles :

- graphprototype.create - créer de nouveaux prototypes de graphiques
- graphprototype.delete - supprimer des prototypes de graphiques
- graphprototype.get - récupérer des prototypes de graphiques
- graphprototype.update - mettre à jour des prototypes de graphiques

Objet prototype de graphique

Les objets suivants sont directement liés à l'API `graphprototype`.

Prototype de graphique

L'objet prototype de graphique possède les propriétés suivantes.

Propriété	Type	Description
graphid	ID	ID du prototype de graphique. Comportement de la propriété: - *lecture seule* - *obligatoire* pour les opérations de mise à jour
height	integer	Hauteur du prototype de graphique en pixels. Comportement de la propriété: - *obligatoire* pour les opérations de création
name	string	Nom du prototype de graphique. Comportement de la propriété: - *obligatoire* pour les opérations de création
width	integer	Largeur du prototype de graphique en pixels. Comportement de la propriété: - *obligatoire* pour les opérations de création
flags	integer	Origine du prototype de graphique. Valeurs possibles : 2 - un prototype de graphique ; 6 - un prototype de graphique découvert Comportement de la propriété: - *lecture seule*
graphtype	integer	Type de disposition du prototype de graphique. Valeurs possibles : 0 - *(par défaut)* normal ; 1 - empilé ; 2 - camembert ; 3 - éclaté.
percent_left	float	Percentile gauche. Par défaut : 0.
percent_right	float	Percentile droit. Par défaut : 0.
show_3d	integer	Indique s'il faut afficher les graphiques découverts de type camembert et éclaté en 3D. Valeurs possibles : 0 - *(par défaut)* afficher en 2D ; 1 - afficher en 3D.

Propriété	Type	Description
show_legend	integer	Indique s'il faut afficher la légende sur le graphique découvert. Valeurs possibles : 0 - masquer ; 1 - *(par défaut)* afficher.
show_work_period	integer	Indique s'il faut afficher le temps de travail sur le graphique découvert. Valeurs possibles : 0 - masquer ; 1 - *(par défaut)* afficher.
templateid	ID	ID du prototype de graphique du modèle parent. Comportement de la propriété: - *lecture seule*
yaxismax	float	Valeur maximale fixe pour l'axe Y.
yaxismin	float	Valeur minimale fixe pour l'axe Y.
ymax_itemid	ID	ID de l'élément utilisé comme valeur maximale pour l'axe Y. Si un utilisateur n'a pas accès à l'élément spécifié, le graphique est affiché comme si `ymax_type` était défini sur « calculated ».
ymax_type	integer	Méthode de calcul de la valeur maximale pour l'axe Y. Valeurs possibles : 0 - *(par défaut)* calculated ; 1 - fixe ; 2 - élément.
ymin_itemid	ID	ID de l'élément utilisé comme valeur minimale pour l'axe Y. Si un utilisateur n'a pas accès à l'élément spécifié, le graphique est affiché comme si `ymin_type` était défini sur « calculated ».
ymin_type	integer	Méthode de calcul de la valeur minimale pour l'axe Y. Valeurs possibles : 0 - *(par défaut)* calculated ; 1 - fixe ; 2 - élément.
discover	integer	État de découverte du prototype de graphique. Valeurs possibles : 0 - *(par défaut)* les nouveaux graphiques seront découverts ; 1 - les nouveaux graphiques ne seront pas découverts et les graphiques existants seront marqués comme perdus.
uuid	string	Identifiant unique universel, utilisé pour lier les prototypes de graphique importés à ceux qui existent déjà. Généré automatiquement s'il n'est pas fourni. Comportement de la propriété: - *pris en charge* si le prototype de graphique appartient à un modèle

graphprototype.create

Description

`object graphprototype.create(object/array graphPrototypes)`

Cette méthode vous permet de créer de nouveaux prototypes de graphique.

Note:
Cette méthode est uniquement disponible pour les types d'utilisateur *Admin* et *Super admin*. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres du rôle utilisateur. Consultez Rôles d'utilisateur pour plus d'informations.

Paramètres

`(object/array)` Prototypes de graphiques à créer.

En plus des propriétés standard des prototypes de graphiques, la méthode accepte les paramètres suivants.

Paramètre	Type	Description
gitems	array	Éléments de graphique à créer pour les prototypes de graphiques. Les éléments de graphique peuvent référencer à la fois des éléments et des prototypes d'éléments, mais au moins un prototype d'élément doit être présent. Comportement des paramètres: *- obligatoire*

Valeurs retournées

`(object)` Renvoie un objet contenant les IDs des prototypes de graphe créés sous la propriété `graphids`. L'ordre des ID renvoyés correspond à l'ordre des prototypes de graphe transmis.

Exemples

Création d'un prototype de graphique

Créez un prototype de graphique avec deux éléments.

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "graphprototype.create",
    "params": {
        "name": "Disk space usage {#FSNAME}",
        "width": 900,
        "height": 200,
        "gitems": [
            {
                "itemid": "22828",
                "color": "00AA00"
            },
            {
                "itemid": "22829",
                "color": "3333FF"
            }
        ]
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "graphids": [
            "652"
        ]
    },
    "id": 1
}
```

Voir aussi

- Élément de graphique

Source

CGraphPrototype::create() dans *ui/include/classes/api/services/CGraphPrototype.php*.

graphprototype.delete

Description

`object graphprototype.delete(array graphPrototypeIds)`

Cette méthode vous permet de supprimer des prototypes de graphique.

> **Note:**
> Cette méthode est disponible uniquement pour les types d'utilisateur *Admin* et *Super admin*. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres du rôle utilisateur. Consultez Rôles d'utilisateur pour plus d'informations.

Paramètres

`(array)` ID des prototypes de graphiques à supprimer.

Valeurs de retour

`(object)` Renvoie un objet contenant les ID des prototypes de graphiques supprimés dans la propriété `graphids`.

Exemples

Suppression de plusieurs prototypes de graphiques

Supprimez deux prototypes de graphiques.

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "graphprototype.delete",
    "params": [
        "652",
        "653"
    ],
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "graphids": [
            "652",
            "653"
        ]
    },
    "id": 1
}
```

Source

CGraphPrototype::delete() dans *ui/include/classes/api/services/CGraphPrototype.php*.

graphprototype.get

Description

`integer/array graphprototype.get(object parameters)`

La méthode permet de récupérer des prototypes de graphiques selon les paramètres fournis.

> **Note:**
> Cette méthode est disponible pour les utilisateurs de tout type. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres des rôles utilisateur. Consultez Rôles utilisateur pour plus d'informations.

Paramètres

`(object)` Paramètres définissant la sortie souhaitée.

La méthode prend en charge les paramètres suivants.

Paramètre	Type	Description
discoveryids	ID/array	Retourner uniquement les prototypes de graphiques appartenant aux règles de découverte indiquées.
graphids	ID/array	Retourner uniquement les prototypes de graphiques ayant les ID indiqués.
groupids	ID/array	Retourner uniquement les prototypes de graphiques appartenant aux hôtes ou aux modèles des groupes d'hôtes ou groupes de modèles indiqués.
hostids	ID/array	Retourner uniquement les prototypes de graphiques appartenant aux hôtes indiqués.
inherited	boolean	Si défini sur `true`, retourner uniquement les prototypes de graphiques hérités d'un modèle.
itemids	ID/array	Retourner uniquement les prototypes de graphiques contenant les prototypes d'éléments indiqués.
templated	boolean	Si défini sur `true`, retourner uniquement les prototypes de graphiques appartenant aux modèles.
templateids	ID/array	Retourner uniquement les prototypes de graphiques appartenant aux modèles indiqués.
selectDiscoveryData	query	Retourner une propriété `discoveryData` avec les données de l'objet de découverte du prototype de graphique. L'objet de découverte du prototype de graphique relie un prototype de graphique découvert à un prototype de graphique à partir duquel il a été découvert. Il possède les propriétés suivantes : `parent_graphid` - `(ID)` ID du prototype de graphique à partir duquel le graphique a été créé ; `status` - `(int)` état de découverte du graphique : 0 - *(par défaut)* le prototype de graphique est découvert, 1 - le prototype de graphique n'est plus découvert ; `ts_delete` - `(timestamp)` heure à laquelle un prototype de graphique qui n'est plus découvert sera supprimé.
selectDiscoveryRule	query	Retourner une propriété `discoveryRule` avec la règle LLD à laquelle appartient le prototype de graphique.
selectDiscoveryRulePrototype	query	Retourner une propriété `discoveryRulePrototype` avec le prototype de règle LLD parent auquel appartient le prototype de graphique.
selectGraphItems	query	Retourner une propriété `gitems` avec les éléments de graphique utilisés dans le prototype de graphique.
selectHostGroups	query	Retourner une propriété `hostgroups` avec les groupes d'hôtes auxquels appartient le prototype de graphique.
selectHosts	query	Retourner une propriété `hosts` avec les hôtes auxquels appartient le prototype de graphique.
selectItems	query	Retourner une propriété `items` avec les éléments et les prototypes d'éléments utilisés dans le prototype de graphique.
selectTemplateGroups	query	Retourner une propriété `templategroups` avec les groupes de modèles auxquels appartient le prototype de graphique.
selectTemplates	query	Retourner une propriété `templates` avec les modèles auxquels appartient le prototype de graphique.

Paramètre	Type	Description
filter	object	Retourner uniquement les résultats correspondant exactement au filtre indiqué. Accepte un objet où les clés sont des noms de propriétés et les valeurs sont soit une valeur unique, soit un tableau de valeurs à comparer. Ne prend pas en charge les propriétés de type de données `text` data type. Prend en charge des propriétés supplémentaires : `host` - nom technique de l'hôte auquel appartient le prototype de graphique ; `hostid` - ID de l'hôte auquel appartient le prototype de graphique.
sortfield	string/array	Trier le résultat selon les propriétés indiquées. Valeurs possibles : `graphid`, `name`, `graphtype`, `discovered`.
countOutput	boolean	Ces paramètres sont décrits dans le commentaire de référence.
editable	boolean	
excludeSearch	boolean	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Valeurs retournées

`(integer/array)` Retourne soit :

- un tableau d'objets ;
- le nombre d'objets récupérés, si le paramètre `countOutput` a été utilisé.

Exemples

Récupération des prototypes de graphiques à partir d'une règle LLD

Récupérez tous les prototypes de graphiques à partir d'une règle LLD.

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "graphprototype.get",
    "params": {
        "output": "extend",
        "discoveryids": "27426"
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "graphid": "1017",
            "name": "Disk space usage {#FSNAME}",
            "width": "600",
            "height": "340",
            "yaxismin": "0.0000",
            "yaxismax": "0.0000",
```

```
            "templateid": "442",
            "show_work_period": "0",
            "show_triggers": "0",
            "graphtype": "2",
            "show_legend": "1",
            "show_3d": "1",
            "percent_left": "0.0000",
            "percent_right": "0.0000",
            "ymin_type": "0",
            "ymax_type": "0",
            "ymin_itemid": "0",
            "ymax_itemid": "0",
            "flags": "2"
            "discover": "0"
        }
    ],
    "id": 1
}
```


Voir aussi

- Règle de découverte
- Élément de graphique
- Élément
- Hôte
- Groupe d'hôtes
- Modèle
- Groupe de modèles

Source

CGraphPrototype::get() dans *ui/include/classes/api/services/CGraphPrototype.php*.

graphprototype.update

Description

`object graphprototype.update(object/array graphPrototypes)`

Cette méthode vous permet de mettre à jour des prototypes de graphique existants.

> **Note:**
> Cette méthode est disponible uniquement pour les types d'utilisateur *Admin* et *Super admin*. Les autorisations d'appel de cette méthode peuvent être révoquées dans les paramètres des rôles utilisateur. Consultez Rôles utilisateur pour plus d'informations.

Paramètres

`(object/array)` Propriétés des prototypes de graphiques à mettre à jour.

La propriété `graphid` doit être définie pour chaque prototype de graphique, toutes les autres propriétés sont facultatives. Seules les propriétés transmises seront mises à jour, toutes les autres resteront inchangées.

En plus des propriétés standard des prototypes de graphiques, la méthode accepte les paramètres suivants.

Paramètre	Type	Description
gitems	array	Éléments de graphique pour remplacer les éléments de graphique existants. Si un élément de graphique a la propriété `gitemid` définie, il sera mis à jour ; sinon, un nouvel élément de graphique sera créé.

Valeurs retournées

`(object)` Retourne un objet contenant les identifiants des prototypes de graphique mis à jour sous la propriété `graphids`.

Exemples

Modification de la taille d'un prototype de graphique

Modifiez la taille d'un prototype de graphique à 1100 sur 400 pixels.

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "graphprototype.update",
    "params": {
        "graphid": "439",
        "width": 1100,
        "height": 400
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "graphids": [
            "439"
        ]
    },
    "id": 1
}
```

Source

CGraphPrototype::update() dans *ui/include/classes/api/services/CGraphPrototype.php*.

Prototype de règle LLD

Cette classe est conçue pour fonctionner avec les prototypes de règles de découverte de bas niveau.

Références d'objet :

- Prototype de règle LLD
  - En-tête HTTP
  - Champ de requête HTTP
- Filtre de prototype de règle LLD
  - Condition de filtre de prototype de règle LLD
- Chemin de macro de prototype LLD
- Prétraitement de prototype de règle LLD
- Surcharges de prototype de règle LLD
  - Filtre de surcharge de prototype de règle LLD
    - Condition de filtre de surcharge de prototype de règle LLD
  - Opération de surcharge de prototype de règle LLD
    - Statut d'opération de surcharge de prototype de règle LLD
    - Découverte d'opération de surcharge de prototype de règle LLD
    - Période d'opération de surcharge de prototype de règle LLD
    - Historique d'opération de surcharge de prototype de règle LLD
    - Tendances d'opération de surcharge de prototype de règle LLD
    - Sévérité d'opération de surcharge de prototype de règle LLD
    - Tag d'opération de surcharge de prototype de règle LLD
    - Modèle d'opération de surcharge de prototype de règle LLD
    - Inventaire d'opération de surcharge de prototype de règle LLD

Méthodes disponibles :

- discoveryruleprototype.create - créer de nouveaux prototypes de règles LLD
- discoveryruleprototype.delete - supprimer des prototypes de règles LLD
- discoveryruleprototype.get - récupérer des prototypes de règles LLD
- discoveryruleprototype.update - mettre à jour des prototypes de règles LLD

Objet prototype de règle LLD

Les objets suivants sont directement liés à l'API `discoveryruleprototype`.

Prototype de règle LLD

L'objet prototype de règle de découverte de bas niveau possède les propriétés suivantes.

Propriété	Type	Description
itemid	ID	ID du prototype de règle LLD. Comportement de la propriété : - *en lecture seule* - *obligatoire* pour les opérations de mise à jour
ruleid	ID	ID de la règle LLD parente ou du prototype de règle LLD parent. Comportement de la propriété : - *en lecture seule* - *obligatoire* pour les opérations de création
delay	string	Intervalle de mise à jour du prototype de règle LLD. Accepte un nombre de secondes (par exemple, 1, 60, 300), une valeur avec un suffixe temporel (par exemple, 30s, 1m, 2h, 1d) ou, facultativement, un ou plusieurs intervalles personnalisés, tous séparés par des points-virgules. Les intervalles personnalisés peuvent combiner des intervalles flexibles et des intervalles de planification. Accepte les macros utilisateur. Si elle est utilisée, la valeur doit être une macro unique. Les macros multiples ou les macros combinées à du texte ne sont pas prises en charge. Les intervalles flexibles peuvent être écrits sous la forme de deux macros séparées par une barre oblique (par exemple, `{$FLEX_INTERVAL}/{$FLEX_PERIOD}`). Exemple : `1h;wd1-5h9-18;{$Macro1}/1-7,00:00-24:00;0/6-7,12:00-24:00;{$Macr` Comportement de la propriété : - *obligatoire* si `type` est défini sur « Zabbix agent » (0), « Simple check » (3), « Zabbix internal » (5), « External check » (10), « Database monitor » (11), « IPMI agent » (12), « SSH agent » (13), « TELNET agent » (14), « JMX agent » (16), « HTTP agent » (19), « SNMP agent » (20), « Script » (21), « Browser » (22), ou si `type` est défini sur « Zabbix agent (active) » (7) et que `key_` ne contient pas « mqtt.get »
hostid	ID	ID de l'hôte auquel appartient le prototype de règle LLD. Comportement de la propriété : - *constant* - *obligatoire* pour les opérations de création
flags	integer	Origine du prototype de règle LLD. Valeurs possibles : 3 - prototype de règle de découverte de bas niveau ; 7 - prototype de règle de découverte de bas niveau découvert. Comportement de la propriété : - *en lecture seule*

Propriété	Type	Description
interfaceid	ID	ID de l'interface de l'hôte du prototype de règle LLD. Comportement de la propriété : - *obligatoire* si le prototype de règle LLD appartient à un hôte et que `type` est défini sur « Zabbix agent », « IPMI agent », « JMX agent » ou « SNMP agent » - *pris en charge* si le prototype de règle LLD appartient à un hôte et que `type` est défini sur « Simple check », « External check », « SSH agent », « TELNET agent » ou « HTTP agent »
key_	string	Clé du prototype de règle LLD. Au moins une macro LLD est requise. Comportement de la propriété : - *obligatoire* pour les opérations de création - *en lecture seule* pour les objets hérités
name	string	Nom du prototype de règle LLD. Comportement de la propriété : - *obligatoire* pour les opérations de création - *en lecture seule* pour les objets hérités
type	integer	Type du prototype de règle LLD. Valeurs possibles : 0 - Zabbix agent ; 2 - Zabbix trapper ; 3 - Simple check ; 5 - Zabbix internal ; 7 - Zabbix agent (active) ; 10 - External check ; 11 - Database monitor ; 12 - IPMI agent ; 13 - SSH agent ; 14 - TELNET agent ; 16 - JMX agent ; 18 - Élément dépendant ; 19 - HTTP agent ; 20 - SNMP agent ; 21 - Script ; 22 - Browser ; 23 - Imbriqué. Comportement de la propriété : - *obligatoire* pour les opérations de création - *en lecture seule* pour les objets hérités
url	string	Chaîne d'URL. Prend en charge les macros utilisateur, {HOST.IP}, {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.NAME}, {HOST.PORT}, {ITEM.ID}, {ITEM.KEY}. Comportement de la propriété : - *obligatoire* si `type` est défini sur « HTTP agent » - *en lecture seule* pour les objets hérités
allow_traps	integer	Autoriser le remplissage de la valeur de manière similaire à l'élément trapper. Valeurs possibles : 0 - *(par défaut)* Ne pas accepter les données entrantes ; 1 - Accepter les données entrantes. Comportement de la propriété : - *pris en charge* si `type` est défini sur « HTTP agent »

Propriété	Type	Description
authtype	integer	Méthode d'authentification. Valeurs possibles si `type` est défini sur « SSH agent » : 0 - *(par défaut)* mot de passe ; 1 - clé publique. Valeurs possibles si `type` est défini sur « HTTP agent » : 0 - *(par défaut)* aucune ; 1 - basic ; 2 - NTLM. Comportement de la propriété : - *pris en charge* si `type` est défini sur « SSH agent » ou « HTTP agent » - *en lecture seule* pour les objets hérités (si `type` est défini sur « HTTP agent »)
description	string	Description du prototype de règle LLD.
error	string	Texte d'erreur en cas de problème lors de la mise à jour de la valeur du prototype de règle LLD. Comportement de la propriété : - *en lecture seule*
follow_redirects	integer	Suivre les redirections de la réponse lors de l'interrogation des données. Valeurs possibles : 0 - Ne pas suivre les redirections ; 1 - *(par défaut)* Suivre les redirections. Comportement de la propriété : - *pris en charge* si `type` est défini sur « HTTP agent » - *en lecture seule* pour les objets hérités
headers	array	Tableau d'en-têtes qui seront envoyés lors de l'exécution d'une requête HTTP. Comportement de la propriété : - *pris en charge* si `type` est défini sur « HTTP agent » - *en lecture seule* pour les objets hérités
http_proxy	string	Chaîne de connexion du proxy HTTP(S). Comportement de la propriété : - *pris en charge* si `type` est défini sur « HTTP agent » - *en lecture seule* pour les objets hérités
ipmi_sensor	string	Capteur IPMI. Comportement de la propriété : - *obligatoire* si `type` est défini sur « IPMI agent » et que `key_` n'est pas défini sur « ipmi.get » - *pris en charge* si `type` est défini sur « IPMI agent » - *en lecture seule* pour les objets hérités
jmx_endpoint	string	Chaîne de connexion personnalisée de l'agent JMX. Valeur par défaut : service:jmx:rmi:///jndi/rmi://{HOST.CONN}:{HOST.PORT}/jmxrmi Comportement de la propriété : - *pris en charge* si `type` est défini sur « JMX agent »
lifetime	string	Période après laquelle les éléments qui ne sont plus découverts seront supprimés. Accepte un nombre de secondes (par exemple, `1`, `60`, `300`), une valeur avec un suffixe temporel (par exemple, `30s`, `1m`, `2h`, `1d`) ou une macro utilisateur. Valeur par défaut : `7d`.

Propriété	Type	Description
lifetime_type	integer	Scénario de suppression des ressources LLD perdues. Valeurs possibles : 0 - *(par défaut)* Supprimer une fois le seuil de durée de vie atteint ; 1 - Ne pas supprimer ; 2 - Supprimer immédiatement.
enabled_lifetime	string	Période après laquelle les éléments qui ne sont plus découverts seront désactivés. Accepte un nombre de secondes (par exemple, `1`, `60`, `300`), une valeur avec un suffixe temporel (par exemple, `30s`, `1m`, `2h`, `1d`) ou une macro utilisateur. Valeur par défaut : `0`.
enabled_lifetime_type	integer	Scénario de désactivation des ressources LLD perdues. Valeurs possibles : 0 - Désactiver une fois le seuil de durée de vie atteint ; 1 - Ne pas désactiver ; 2 - *(par défaut)* Désactiver immédiatement.
master_itemid	ID	ID de l'élément maître. Une règle de découverte ne peut pas être l'élément maître d'une autre règle de découverte. Comportement de la propriété : - *obligatoire* si `type` est défini sur « Élément dépendant » - *en lecture seule* pour les objets hérités
output_format	integer	Indique si la réponse doit être convertie en JSON. Valeurs possibles : 0 - *(par défaut)* Stocker les données brutes ; 1 - Convertir en JSON. Comportement de la propriété : - *pris en charge* si `type` est défini sur « HTTP agent » - *en lecture seule* pour les objets hérités
params	string	Paramètres supplémentaires selon le type du prototype de règle LLD : - script exécuté pour les règles LLD SSH et Telnet ; - requête SQL pour les règles LLD de supervision de bases de données ; - formule pour les règles LLD calculées ; - script pour les règles LLD de type script et navigateur. Comportement de la propriété : - *obligatoire* si `type` est défini sur « Database monitor », « SSH agent », « TELNET agent », « Script » ou « Browser » - *en lecture seule* pour les objets hérités (si `type` est défini sur « Script » ou « Browser »)
parameters	object/array	Paramètres supplémentaires si `type` est défini sur « Script » ou « Browser ». Tableau d'objets possédant les propriétés `name` et `value`, où `name` doit être unique. Comportement de la propriété : - *pris en charge* si `type` est défini sur « Script » ou « Browser » - *en lecture seule* pour les objets hérités
password	string	Mot de passe d'authentification. Comportement de la propriété : - *obligatoire* si `type` est défini sur « JMX agent » et que `username` est défini - *pris en charge* si `type` est défini sur « Simple check », « Database monitor », « SSH agent », « TELNET agent » ou « HTTP agent » - *en lecture seule* pour les objets hérités (si `type` est défini sur « HTTP agent »)

Propriété	Type	Description
post_type	integer	Type du corps des données POST stocké dans la propriété `posts`. Valeurs possibles : 0 - *(par défaut)* Données brutes ; 2 - Données JSON ; 3 - Données XML. Comportement de la propriété : - *pris en charge* si `type` est défini sur « HTTP agent » - *en lecture seule* pour les objets hérités
posts	string	Données du corps de la requête HTTP(S). Comportement de la propriété : - *obligatoire* si `type` est défini sur « HTTP agent » et que `post_type` est défini sur « Données JSON » ou « Données XML » - *pris en charge* si `type` est défini sur « HTTP agent » et que `post_type` est défini sur « Données brutes » - *en lecture seule* pour les objets hérités
privatekey	string	Nom du fichier de clé privée. Comportement de la propriété : - *obligatoire* si `type` est défini sur « SSH agent » et que `authtype` est défini sur « clé publique »
publickey	string	Nom du fichier de clé publique. Comportement de la propriété : - *obligatoire* si `type` est défini sur « SSH agent » et que `authtype` est défini sur « clé publique »
query_fields	array	Tableau de champs de requête qui seront envoyés lors de l'exécution d'une requête HTTP. Comportement de la propriété : - *pris en charge* si `type` est défini sur « HTTP agent » - *en lecture seule* pour les objets hérités
request_method	integer	Type de méthode de requête. Valeurs possibles : 0 - *(par défaut)* GET ; 1 - POST ; 2 - PUT ; 3 - HEAD. Comportement de la propriété : - *pris en charge* si `type` est défini sur « HTTP agent » - *en lecture seule* pour les objets hérités
retrieve_mode	integer	Partie de la réponse à stocker. Valeurs possibles si `request_method` est défini sur « GET », « POST » ou « PUT » : 0 - *(par défaut)* Corps ; 1 - En-têtes ; 2 - Le corps et les en-têtes seront stockés. Valeurs possibles si `request_method` est défini sur « HEAD » : 1 - En-têtes. Comportement de la propriété : - *pris en charge* si `type` est défini sur « HTTP agent » - *en lecture seule* pour les objets hérités

Propriété	Type	Description
snmp_oid	string	OID SNMP. Comportement de la propriété : - *obligatoire* si `type` est défini sur « SNMP agent » - *en lecture seule* pour les objets hérités
ssl_cert_file	string	Chemin du fichier de clé SSL publique. Comportement de la propriété : - *pris en charge* si `type` est défini sur « HTTP agent » - *en lecture seule* pour les objets hérités
ssl_key_file	string	Chemin du fichier de clé SSL privée. Comportement de la propriété : - *pris en charge* si `type` est défini sur « HTTP agent » - *en lecture seule* pour les objets hérités
ssl_key_password	string	Mot de passe du fichier de clé SSL. Comportement de la propriété : - *pris en charge* si `type` est défini sur « HTTP agent » - *en lecture seule* pour les objets hérités
state	integer	État du prototype de règle LLD. Valeurs possibles : 0 - *(par défaut)* normal ; 1 - non pris en charge. Comportement de la propriété : - *en lecture seule*
status	integer	Statut du prototype de règle LLD. Valeurs possibles : 0 - *(par défaut)* prototype de règle LLD activé ; 1 - prototype de règle LLD désactivé.
status_codes	string	Plages de codes d'état HTTP requis, séparées par des virgules. Prend également en charge les macros utilisateur dans une liste séparée par des virgules. Exemple : 200,200-{$M},{$M},200-400 Comportement de la propriété : - *pris en charge* si `type` est défini sur « HTTP agent » - *en lecture seule* pour les objets hérités
templateid	ID	ID du prototype de règle LLD parent du modèle. Comportement de la propriété : - *en lecture seule*

Propriété	Type	Description
timeout	string	Délai d'expiration de la requête d'interrogation des données de l'élément. Accepte un nombre de secondes (par exemple, 1, 60, 300) ou une valeur avec un suffixe temporel (par exemple, 30s, 1m). Accepte également les macros utilisateur. Plage de valeurs possibles : 1-600s. Valeur par défaut : "" - utiliser les paramètres du proxy ou les paramètres globaux. Comportement de la propriété : - *pris en charge* si type est défini sur « Zabbix agent » (0), « Simple check » (3) et que key_ ne commence pas par « vmware. » ni « icmpping », « Zabbix agent (active) » (7), « External check » (10), « Database monitor » (11), « SSH agent » (13), « TELNET agent » (14), « HTTP agent » (19), « SNMP agent » (20) et que snmp_oid commence par « walk[» ou « get[», « Script » (21), « Browser » (22) - *en lecture seule* pour les objets hérités
trapper_hosts	string	Hôtes autorisés. Comportement de la propriété : - *pris en charge* si type est défini sur « Zabbix trapper », ou si type est défini sur « HTTP agent » et que allow_traps est défini sur « Autoriser les données entrantes ». Si la propriété est omise, elle est automatiquement définie sur la macro utilisateur {$TRAPPER.ALLOWED_HOSTS} (valeur par défaut). Par défaut, {$TRAPPER.ALLOWED_HOSTS} vaut 127.0.0.1,::1. Si la propriété est définie sur une chaîne vide, les connexions entrantes seront refusées pour tous les hôtes.
username	string	Nom d'utilisateur pour l'authentification. Comportement de la propriété : - *obligatoire* si type est défini sur « SSH agent », « TELNET agent », ou si type est défini sur « JMX agent » et que password est défini - *pris en charge* si type est défini sur « Simple check », « Database monitor » ou « HTTP agent » - *en lecture seule* pour les objets hérités (si type est défini sur « HTTP agent »)
uuid	string	Identifiant unique universel utilisé pour associer les prototypes de règles LLD importés à ceux qui existent déjà. Généré automatiquement s'il n'est pas fourni. Comportement de la propriété : - *pris en charge* si le prototype de règle LLD appartient à un modèle
verify_host	integer	Indique s'il faut vérifier que le nom d'hôte de la connexion correspond à celui indiqué dans le certificat de l'hôte. Valeurs possibles : 0 - *(par défaut)* Ne pas vérifier ; 1 - Vérifier. Comportement de la propriété : - *pris en charge* si type est défini sur « HTTP agent » - *en lecture seule* pour les objets hérités

Propriété	Type	Description
verify_peer	integer	Indique s'il faut vérifier que le certificat de l'hôte est authentique. Valeurs possibles : 0 - *(par défaut)* Ne pas vérifier ; 1 - Vérifier. Comportement de la propriété : - *pris en charge* si `type` est défini sur « HTTP agent » - *en lecture seule* pour les objets hérités

En-tête HTTP

L'objet d'en-tête possède les propriétés suivantes :

Propriété	Type	Description
name	string	Nom de l'en-tête HTTP. Comportement de la propriété : - *obligatoire*
value	string	Valeur de l'en-tête. Comportement de la propriété : - *obligatoire*

Champ de requête HTTP

L'objet de champ de requête définit un nom et une valeur utilisés pour spécifier un paramètre d'URL. Il possède les propriétés suivantes :

Propriété	Type	Description
name	string	Nom du paramètre. Comportement de la propriété : - *obligatoire*
value	string	Valeur du paramètre. Comportement de la propriété : - *obligatoire*

Filtre de prototype de règle LLD

L'objet de filtre de prototype de règle LLD définit un ensemble de conditions pouvant être utilisées pour filtrer les objets découverts. Il possède les propriétés suivantes :

Propriété	Type	Description
conditions	object/array	Ensemble de conditions de filtre à utiliser pour filtrer les résultats. Les conditions seront triées dans l'ordre de leur placement dans la formule. Comportement de la propriété: - *obligatoire*

Propriété	Type	Description
evaltype	integer	Méthode d'évaluation des conditions de filtre. Valeurs possibles : 0 - Et/Ou ; 1 - Et ; 2 - Ou ; 3 - Expression personnalisée. Comportement de la propriété: - *obligatoire*
eval_formula	string	Expression générée qui sera utilisée pour évaluer les conditions de filtre. L'expression contient des ID qui référencent des conditions de filtre spécifiques par leur `formulaid`. La valeur de `eval_formula` est égale à la valeur de `formula` pour les filtres avec une expression personnalisée. Comportement de la propriété: - *lecture seule*
formula	string	Expression définie par l'utilisateur à utiliser pour évaluer les conditions des filtres avec une expression personnalisée. L'expression doit contenir des ID qui référencent des conditions de filtre spécifiques par leur `formulaid`. Les ID utilisés dans l'expression doivent correspondre exactement à ceux définis dans les conditions de filtre : aucune condition ne peut rester inutilisée ou omise. Comportement de la propriété: - *obligatoire* si `evaltype` est défini sur "expression personnalisée"

Condition de filtre du prototype de règle LLD

L'objet de condition de filtre du prototype de règle LLD définit une vérification distincte à effectuer sur la valeur d'une macro LLD. Il possède les propriétés suivantes :

Propriété	Type	Description
macro	string	Macro LLD sur laquelle effectuer la vérification. Comportement de la propriété : - *obligatoire*
value	string	Valeur à comparer. Comportement de la propriété : - *obligatoire* si `operator` est défini sur « correspond à l'expression régulière » ou « ne correspond pas à l'expression régulière »
formulaid	string	ID unique arbitraire utilisé pour référencer la condition depuis une expression personnalisée. Ne peut contenir que des lettres majuscules. L'ID doit être défini par l'utilisateur lors de la modification des conditions de filtre, mais sera généré à nouveau lors de leur demande ultérieure. Comportement de la propriété : - *obligatoire* si `evaltype` de l'objet de filtre du prototype de règle LLD est défini sur « expression personnalisée »
operator	integer	Opérateur de condition. Valeurs possibles : 8 - *(par défaut)* correspond à l'expression régulière ; 9 - ne correspond pas à l'expression régulière ; 12 - existe ; 13 - n'existe pas.

Note:
Pour mieux comprendre comment utiliser les filtres avec différents types d'expressions, consultez les exemples dans les pages des méthodes discoveryruleprototype.get et discoveryruleprototype.create.

Chemin de macro LLD

Le chemin de macro LLD possède les propriétés suivantes :

Propriété	Type	Description
lld_macro	string	Macro LLD. Comportement de la propriété : - *obligatoire*
path	string	Sélecteur de la valeur qui sera attribuée à la macro correspondante. Comportement de la propriété : - *obligatoire*

Prétraitement du prototype de règle LLD

L'objet de prétraitement du prototype de règle LLD possède les propriétés suivantes.

Propriété	Type	Description
type	integer	Type d'option de prétraitement. Valeurs possibles : 5 - Expression régulière ; 11 - XPath XML ; 12 - JSONPath ; 14 - Correspond à une expression régulière ; 15 - Ne correspond pas à une expression régulière ; 16 - Vérifier la présence d'une erreur dans JSON ; 17 - Vérifier la présence d'une erreur dans XML ; 20 - Ignorer les valeurs inchangées avec intervalle de contrôle ; 21 - JavaScript ; 23 - Prometheus vers JSON ; 24 - CSV vers JSON ; 25 - Remplacer ; 27 - XML vers JSON ; 28 - Valeur de parcours SNMP ; 29 - Parcours SNMP vers JSON ; 30 - Valeur SNMP get. Comportement de la propriété : - *obligatoire*
params	string	Paramètres supplémentaires utilisés par l'option de prétraitement. Plusieurs paramètres sont séparés par le caractère de nouvelle ligne (\n). Comportement de la propriété : - *obligatoire* si `type` est défini sur « Expression régulière » (5), « XPath XML » (11), « JSONPath » (12), « Correspond à une expression régulière » (14), « Ne correspond pas à une expression régulière » (15), « Vérifier la présence d'une erreur dans JSON » (16), « Vérifier la présence d'une erreur dans XML » (17), « Ignorer les valeurs inchangées avec intervalle de contrôle » (20), « JavaScript » (21), « Prometheus vers JSON » (23), « CSV vers JSON » (24), « Remplacer » (25), « Valeur de parcours SNMP » (28), « Parcours SNMP vers JSON » (29) ou « Valeur SNMP get » (30)

Propriété	Type	Description
error_handler	integer	Type d'action utilisé en cas d'échec de l'étape de prétraitement. Valeurs possibles : 0 - Le message d'erreur est défini par le serveur Zabbix ; 1 - Ignorer la valeur ; 2 - Définir une valeur personnalisée ; 3 - Définir un message d'erreur personnalisé. Comportement de la propriété : - *obligatoire* si `type` est défini sur « Expression régulière » (5), « XPath XML » (11), « JSONPath » (12), « Correspond à une expression régulière » (14), « Ne correspond pas à une expression régulière » (15), « Vérifier la présence d'une erreur dans JSON » (16), « Vérifier la présence d'une erreur dans XML » (17), « Prometheus vers JSON » (23), « CSV vers JSON » (24), « XML vers JSON » (27), « Valeur de parcours SNMP » (28), « Parcours SNMP vers JSON » (29) ou « Valeur SNMP get » (30)
error_handler_params	string	Paramètres du gestionnaire d'erreurs. Comportement de la propriété : - *obligatoire* si `error_handler` est défini sur « Définir une valeur personnalisée » ou « Définir un message d'erreur personnalisé »

Les paramètres et gestionnaires d'erreurs suivants sont pris en charge pour chaque type de prétraitement.

Type de prétraitement	Nom	Paramètre 1	Paramètre 2	Paramètre 3	Gestionnaires d'erreurs pris en charge
5	Expression régulière	pattern[1]	output[2]		0, 1, 2, 3
11	XPath XML	path[3]			0, 1, 2, 3
12	JSONPath	path[3]			0, 1, 2, 3
14	Correspond à une expression régulière	pattern[1]			0, 1, 2, 3
15	Ne correspond pas à une expression régulière	pattern[1]			0, 1, 2, 3
16	Vérifier la présence d'une erreur dans JSON	path[3]			0, 1, 2, 3

Type de prétraitement	Nom	Paramètre 1	Paramètre 2	Paramètre 3	Gestionnaires d'erreurs pris en charge
17	Vérifier la présence d'une erreur dans XML	path[3]			0, 1, 2, 3
20	Ignorer les valeurs inchangées avec intervalle de contrôle	seconds^4, 5^			
21	JavaScript	script[2]			
23	Prometheus vers JSON	pattern^5, 6^			0, 1, 2, 3
24	CSV vers JSON	character[2]	character[2]	0,1	0, 1, 2, 3
25	Remplacer	search string[2]	replacement[2]		
27	XML vers JSON				0, 1, 2, 3
28	Valeur de parcours SNMP	OID[2]	Format : 0 - Inchangé 1 - UTF-8 depuis Hex-STRING 2 - MAC depuis Hex-STRING 3 - Entier depuis BITS		0, 1, 2, 3
29	Parcours SNMP vers JSON[7]	Field name[2]	OID prefix[2]	Format : 0 - Inchangé 1 - UTF-8 depuis Hex-STRING 2 - MAC depuis Hex-STRING 3 - Entier depuis BITS	0, 1, 2, 3
30	Valeur SNMP get	Format : 1 - UTF-8 depuis Hex-STRING 2 - MAC depuis Hex-STRING 3 - Entier depuis BITS			0, 1, 2, 3

[1] expression régulière
 [2] chaîne
 [3] JSONPath ou XPath XML
 [4] entier positif (avec prise en charge des suffixes de temps, par exemple 30s, 1m, 2h, 1d)
 [5] macro utilisateur
 [6] modèle Prometheus suivant la syntaxe : `<metric name>{<label name>="<label value>", ...} == <value>`. Chaque composant du modèle Prometheus (métrique, nom du libellé, valeur

du libellé et valeur de la métrique) peut être une macro utilisateur.
 [7] Prend en charge plusieurs enregistrements « Field name,OID prefix,Format » délimités par un caractère de nouvelle ligne.

Remplacements de prototypes de règles LLD

L'objet de remplacements de prototypes de règles LLD définit un ensemble de règles (filtres, conditions et opérations) utilisées pour remplacer les propriétés de différents objets prototypes. Il possède les propriétés suivantes :

Propriété	Type	Description
name	string	Nom de remplacement unique. Comportement de la propriété : - *obligatoire*
step	integer	Numéro d'ordre unique du remplacement. Comportement de la propriété : - *obligatoire*
stop	integer	Arrêter le traitement des remplacements suivants en cas de correspondance. Valeurs possibles : 0 - *(par défaut)* ne pas arrêter le traitement des remplacements ; 1 - arrêter le traitement des remplacements si le filtre correspond.
filter	object	Filtre de remplacement.
operations	object/array	Opérations de remplacement.

Filtre de substitution du prototype de règle LLD

L'objet de filtre de substitution du prototype de règle LLD définit un ensemble de conditions qui, si elles correspondent à l'objet découvert, appliquent la substitution. Il possède les propriétés suivantes :

Propriété	Type	Description
conditions	object/array	Ensemble de conditions de filtre de substitution à utiliser pour faire correspondre les objets découverts. Les conditions seront triées dans l'ordre de leur placement dans la formule. Comportement de la propriété : - *obligatoire*
evaltype	integer	Méthode d'évaluation des conditions du filtre de substitution. Valeurs possibles : 0 - Et/Ou ; 1 - Et ; 2 - Ou ; 3 - Expression personnalisée. Comportement de la propriété : - *obligatoire*
eval_formula	string	Expression générée qui sera utilisée pour évaluer les conditions du filtre de substitution. L'expression contient des ID qui référencent des conditions spécifiques du filtre de substitution par leur `formulaid`. La valeur de `eval_formula` est égale à la valeur de `formula` pour les filtres avec une expression personnalisée. Comportement de la propriété : - *lecture seule*

Propriété	Type	Description
formula	string	Expression définie par l'utilisateur à utiliser pour évaluer les conditions des filtres de substitution avec une expression personnalisée. L'expression doit contenir des ID qui référencent des conditions spécifiques du filtre de substitution par leur `formulaid`. Les ID utilisés dans l'expression doivent correspondre exactement à ceux définis dans les conditions du filtre de substitution : aucune condition ne peut rester inutilisée ou omise. Comportement de la propriété : - *obligatoire* si `evaltype` est défini sur "expression personnalisée"

Condition de filtre de substitution de prototype de règle LLD

L'objet de condition de filtre de substitution de prototype de règle LLD définit une vérification distincte à effectuer sur la valeur d'une macro LLD. Il possède les propriétés suivantes :

Propriété	Type	Description
macro	string	Macro LLD sur laquelle effectuer la vérification. Comportement de la propriété : - *obligatoire*
value	string	Valeur à comparer. Comportement de la propriété : - *obligatoire* si `operator` est défini sur "correspond à l'expression régulière" ou "ne correspond pas à l'expression régulière"
formulaid	string	ID unique arbitraire utilisé pour référencer la condition depuis une expression personnalisée. Ne peut contenir que des lettres majuscules. L'ID doit être défini par l'utilisateur lors de la modification des conditions de filtre, mais sera généré à nouveau lors de leur demande ultérieure. Comportement de la propriété : - *obligatoire* si `evaltype` de l'objet de filtre de substitution de prototype de règle LLD est défini sur "expression personnalisée"
operator	integer	Opérateur de condition. Valeurs possibles : 8 - *(par défaut)* correspond à l'expression régulière ; 9 - ne correspond pas à l'expression régulière ; 12 - existe ; 13 - n'existe pas.

Opération de surcharge de prototype de règle LLD

L'opération de surcharge de prototype de règle LLD est une combinaison de conditions et d'actions à effectuer sur l'objet prototype. Elle possède les propriétés suivantes :

Propriété	Type	Description
operationobject	integer	Type d'objet découvert sur lequel effectuer l'action. Valeurs possibles : 0 - Prototype d'élément ; 1 - Prototype de déclencheur ; 2 - Prototype de graphique ; 3 - Prototype d'hôte. Comportement de la propriété : - *obligatoire*

Propriété	Type	Description
operator	integer	Opérateur de condition de surcharge. Valeurs possibles : 0 - *(par défaut)* égal à ; 1 - différent de ; 2 - contient ; 3 - ne contient pas ; 8 - correspond ; 9 - ne correspond pas.
value	string	Motif à faire correspondre au nom du prototype d'élément, de déclencheur, de graphique ou d'hôte selon l'objet sélectionné.
opstatus	object	Objet d'état d'opération de surcharge pour les objets prototype d'élément, de déclencheur et d'hôte.
opdiscover	object	Objet d'état de découverte d'opération de surcharge (tous les types d'objet).
opperiod	object	Objet de période d'opération de surcharge (intervalle de mise à jour) pour l'objet prototype d'élément.
ophistory	object	Objet d'historique d'opération de surcharge pour l'objet prototype d'élément.
optrends	object	Objet de tendances d'opération de surcharge pour l'objet prototype d'élément.
opseverity	object	Objet de gravité d'opération de surcharge pour l'objet prototype de déclencheur.
optag	object/array	Objet d'étiquette d'opération de surcharge pour les objets prototype de déclencheur et d'hôte.
optemplate	object/array	Objet de modèle d'opération de surcharge pour l'objet prototype d'hôte.
opinventory	object	Objet d'inventaire d'opération de surcharge pour l'objet prototype d'hôte.

Statut de l'opération de remplacement du prototype de règle LLD

Statut de l'opération de remplacement du prototype de règle LLD défini pour l'objet découvert. Il possède les propriétés suivantes :

Propriété	Type	Description
status	integer	Remplace le statut de l'objet sélectionné. Valeurs possibles : 0 - Créer activé ; 1 - Créer désactivé. Comportement de la propriété : - *obligatoire*

Découverte de l'opération de substitution du prototype de règle LLD

Statut de découverte de l'opération de substitution du prototype de règle LLD défini sur l'objet découvert. Il possède les propriétés suivantes :

Propriété	Type	Description
discover	integer	Remplace le statut de découverte pour l'objet sélectionné. Valeurs possibles : 0 - Oui, continuer à découvrir les objets ; 1 - Non, les nouveaux objets ne seront pas découverts et les objets existants seront marqués comme perdus. Comportement de la propriété : - *obligatoire*

Période de fonctionnement de la substitution du prototype de règle LLD

La période de fonctionnement de la substitution du prototype de règle LLD est une valeur d'intervalle de mise à jour définie pour l'élément découvert. Elle possède les propriétés suivantes :

Property	Type	Description
delay	string	Remplace l'intervalle de mise à jour du prototype d'élément. Accepte un nombre de secondes (par exemple, 1, 60, 300), une valeur avec un suffixe temporel (par exemple, 30s, 1m, 2h, 1d) ou, éventuellement, un ou plusieurs intervalles personnalisés, tous séparés par des points-virgules. Les intervalles personnalisés peuvent être une combinaison d'intervalles flexibles et d'intervalles de planification. Accepte les macros utilisateur ou les macros LLD. Si elle est utilisée, la valeur doit être une macro unique. Plusieurs macros ou des macros combinées à du texte ne sont pas pris en charge. Les intervalles flexibles peuvent être écrits sous la forme de deux macros séparées par une barre oblique (par exemple, `{$FLEX_INTERVAL}/{$FLEX_PERIOD}`). Exemple : `1h;wd1-5h9-18;{$Macro1}/1-7,00:00-24:00;0/6-7,12:00-24:00;{$Macr` Comportement de la propriété : - *required*

Historique de l'opération de substitution du prototype de règle LLD

Valeur de l'historique de l'opération de substitution du prototype de règle LLD définie pour l'élément découvert. Elle possède les propriétés suivantes :

Propriété	Type	Description
history	string	Remplace l'historique du prototype d'élément, qui est une unité de temps définissant la durée de conservation des données d'historique. Accepte également une macro utilisateur et une macro LLD. Comportement de la propriété : - *obligatoire*

Tendances de l'opération de substitution du prototype de règle LLD

Valeur des tendances de l'opération de substitution du prototype de règle LLD définie pour l'élément découvert. Elle possède les propriétés suivantes :

Propriété	Type	Description
trends	string	Remplace les tendances du prototype d'élément, c'est-à-dire une unité de temps indiquant combien de temps les données de tendances doivent être conservées. Accepte également une macro utilisateur et une macro LLD. Comportement de la propriété : - *obligatoire*

Gravité de l'opération de substitution du prototype de règle LLD

Valeur de gravité de l'opération de substitution du prototype de règle LLD définie pour le déclencheur découvert. Elle possède les propriétés suivantes :

Propriété	Type	Description
severity	integer	Remplace la gravité du prototype de déclencheur. Valeurs possibles : 0 - *(par défaut)* non classé ; 1 - information ; 2 - avertissement ; 3 - moyenne ; 4 - élevée ; 5 - sinistre. Comportement de la propriété : - *obligatoire*

Balise d'opération de substitution de prototype de règle LLD

L'objet de balise d'opération de substitution de prototype de règle LLD contient le nom et la valeur de la balise qui sont définis sur l'objet découvert. Il possède les propriétés suivantes :

Property	Type	Description
tag	string	Nouveau nom de balise. Comportement de la propriété : - *obligatoire*
value	string	Nouvelle valeur de balise.

Modèle d'opération de substitution de prototype de règle LLD

Objet de modèle d'opération de substitution de prototype de règle LLD lié à l'hôte découvert. Il possède les propriétés suivantes :

Propriété	Type	Description
templateid	ID	Remplace le modèle des modèles liés du prototype d'hôte. Comportement de la propriété : - *obligatoire*

Valeur du mode d'inventaire de l'opération de remplacement du prototype de règle LLD

Valeur du mode d'inventaire de l'opération de remplacement du prototype de règle LLD définie sur l'hôte découvert. Elle possède les propriétés suivantes :

Propriété	Type	Description
inventory_mode	integer	Remplace le mode d'inventaire du prototype d'hôte. Valeurs possibles : -1 - désactivé ; 0 - *(par défaut)* manuel ; 1 - automatique. Comportement des propriétés : - *obligatoire*

discoveryruleprototype.create

Description

`object discoveryruleprototype.create(object/array lldRules)`

Cette méthode vous permet de créer de nouveaux modèles de règle LLD.

Note:
Cette méthode est disponible uniquement pour les types d'utilisateur *Admin* et *Super admin*. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres des rôles utilisateur. Consultez Rôles utilisateur pour plus d'informations.

Paramètres

`(object/array)` Prototypes de règles LLD à créer.

En plus des propriétés standard des prototypes de règles LLD, la méthode accepte les paramètres suivants.

Paramètre	Type	Description
filter	object	Filtre de prototype de règle LLD pour la règle LLD.
preprocessing	object/array	Options de prétraitement de prototype de règle LLD.
lld_macro_paths	object/array	Options de lld_macro_path du prototype de règle LLD.
overrides	object/array	Options de surcharges de prototype de règle LLD.

Valeurs de retour

`(object)` Renvoie un objet contenant les ID des prototypes de règle LLD créés dans la propriété `itemids`. L'ordre des ID renvoyés correspond à l'ordre des prototypes de règle LLD transmis.

Exemples

Création d'un prototype de règle LLD

Créez un prototype de règle LLD (Type : imbriqué) pour découvrir les tablespaces dans l'instance de base de données.

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "discoveryruleprototype.create",
    "params": {
        "name": "Discover tablespaces for {#DB}",
        "key_": "db.tablespace.discovery[{#DB}]",
        "hostid": "10084",
        "ruleid": "47251",
        "type": 23
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "47252"
        ]
    },
    "id": 1
}
```

Voir aussi

- Filtre de prototype de règle LLD
- Chemins de macro LLD
- Prétraitement de prototype de règle LLD

Source

CDiscoveryRulePrototype::create() dans *ui/include/classes/api/services/CDiscoveryRulePrototype.php*.

discoveryruleprototype.delete

Description

`object discoveryruleprototype.delete(array lldRuleIds)`

Cette méthode vous permet de supprimer des prototypes de règle LLD.

> **Note:**
> Cette méthode est disponible uniquement pour les types d'utilisateur *Admin* et *Super admin*. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres du rôle utilisateur. Consultez Rôles d'utilisateur pour plus d'informations.

Paramètres

`(array)` ID des prototypes de règle LLD à supprimer.

Valeurs de retour

`(object)` Renvoie un objet contenant les ID des prototypes de règle LLD supprimés dans la propriété `ruleids`.

Exemples

Suppression de plusieurs prototypes de règle LLD

Supprimez deux prototypes de règle LLD.

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "discoveryruleprototype.delete",
    "params": [
        "47252",
        "47253"
    ],
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "ruleids": [
            "47252",
            "47253"
        ]
    },
```

```
    "id": 1
}
```

Source

CDiscoveryRulePrototype::delete() dans *ui/include/classes/api/services/CDiscoveryRulePrototype.php*.

discoveryruleprototype.get

Description

`integer/array discoveryruleprototype.get(object parameters)`

La méthode permet de récupérer les prototypes de règle LLD selon les paramètres fournis.

> **Note:**
> Cette méthode est disponible pour les utilisateurs de tout type. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres des rôles utilisateur. Consultez Rôles utilisateur pour plus d'informations.

Paramètres

`(object)` Paramètres définissant la sortie souhaitée.

La méthode prend en charge les paramètres suivants.

Paramètre	Type	Description
discoveryids	ID/array	Retourner uniquement les prototypes de règle LLD qui appartiennent aux règles LLD ou aux prototypes de règle LLD indiqués.
itemids	ID/array	Retourner uniquement les prototypes de règle LLD avec les ID indiqués.
groupids	ID/array	Retourner uniquement les prototypes de règle LLD qui appartiennent aux hôtes des groupes indiqués.
hostids	ID/array	Retourner uniquement les prototypes de règle LLD qui appartiennent aux hôtes indiqués.
inherited	boolean	Si défini sur `true`, retourner uniquement les prototypes de règle LLD hérités d'un modèle.
interfaceids	ID/array	Retourner uniquement les prototypes de règle LLD qui utilisent les interfaces d'hôte indiquées.
monitored	boolean	Si défini sur `true`, retourner uniquement les prototypes de règle LLD activés qui appartiennent à des hôtes supervisés.
templated	boolean	Si défini sur `true`, retourner uniquement les prototypes de règle LLD qui appartiennent à des modèles.
templateids	ID/array	Retourner uniquement les prototypes de règle LLD qui appartiennent aux modèles indiqués.
selectDiscoveryRule	query	Retourner une propriété `discoveryRule` avec la règle LLD parente du prototype de règle LLD. Une réponse vide sera renvoyée si le parent est un prototype de règle de découverte.
selectDiscoveryRulePrototype	query	Retourner une propriété `discoveryRulePrototype` avec le prototype de règle LLD parent du prototype de règle LLD. Une réponse vide sera renvoyée si le parent est une règle de découverte (et non un prototype).
selectDiscoveryRulePrototypes	query	Retourner une propriété `discoveryRulePrototypes` avec les prototypes de règle LLD enfants qui appartiennent au prototype de règle LLD. Prend en charge `count`.
selectFilter	query	Retourner une propriété `filter` avec les données du filtre utilisé par le prototype de règle LLD.
selectGraphs	query	Retourne une propriété `graphs` avec les prototypes de graphique qui appartiennent au prototype de règle LLD. Prend en charge `count`.

Paramètre	Type	Description
selectHostPrototypes	query	Retourner une propriété `hostPrototypes` avec les prototypes d'hôte qui appartiennent au prototype de règle LLD. Prend en charge `count`.
selectHosts	query	Retourner une propriété `hosts` avec un tableau d'hôtes auxquels appartient le prototype de règle LLD.
selectItems	query	Retourner une propriété `items` avec les prototypes d'élément qui appartiennent au prototype de règle LLD. Prend en charge `count`.
selectTriggers	query	Retourner une propriété `triggers` avec les prototypes de déclencheur qui appartiennent au prototype de règle LLD. Prend en charge `count`.
selectLLDMacroPaths	query	Retourner une propriété `lld_macro_paths` avec une liste de macros LLD et les chemins vers les valeurs attribuées à chaque macro correspondante.
selectPreprocessing	query	Retourner une propriété `preprocessing` avec les options de prétraitement du prototype de règle LLD.
selectOverrides	query	Retourner une propriété `lld_rule_overrides` avec une liste de filtres de substitution, de conditions et d'opérations exécutées sur les objets prototypes.
filter	object	Retourner uniquement les résultats qui correspondent exactement au filtre indiqué. Accepte un objet où les clés sont des noms de propriété et les valeurs sont soit une valeur unique, soit un tableau de valeurs à faire correspondre. Ne prend pas en charge les propriétés de type de données `text` data type. Prend en charge des propriétés supplémentaires : `host` - nom technique de l'hôte auquel appartient le prototype de règle LLD.
limitSelects	integer	Limite le nombre d'enregistrements renvoyés par les sous-sélections. S'applique aux sous-sélections suivantes : `selectItems`, `selectGraphs`, `selectTriggers`.
sortfield	string/array	Trier le résultat selon les propriétés indiquées. Valeurs possibles : `itemid`, `name`, `key_`, `delay`, `type`, `status`.
countOutput	boolean	Ces paramètres sont décrits dans le commentaire de référence.
editable	boolean	
excludeSearch	boolean	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Valeurs de retour

`(integer/array)` renvoie soit :

- un tableau d'objets ;
- le nombre d'objets récupérés, si le paramètre `countOutput` a été utilisé.

Exemples

Récupération des prototypes de règles de découverte à partir d'un hôte

Récupérez tous les prototypes de règles de découverte pour un ID d'hôte spécifique.

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryruleprototype.get",
    "params": {
        "hostids": "10084"
    },
    "id": 1
}
```

Réponse :


```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "itemid": "47253",
            "type": "23",
            "snmp_oid": "",
            "hostid": "10084",
            "name": "Découvrir les tablespaces pour {#DB}",
            "key_": "db.tablespace.discovery[{#DB}]",
            "delay": "0",
            "history": "31d",
            "trends": "365d",
            "status": "0",
            "value_type": "4",
            "trapper_hosts": "",
            "units": "",
            "logtimefmt": "",
            "templateid": "0",
            "valuemapid": "0",
            "params": "",
            "ipmi_sensor": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "flags": "3",
            "interfaceid": "0",
            "description": "",
            "inventory_link": "0",
            "lifetime": "7d",
            "jmx_endpoint": "",
            "master_itemid": "0",
            "timeout": "",
            "url": "",
            "query_fields": [],
            "posts": "",
            "status_codes": "200",
            "follow_redirects": "1",
            "post_type": "0",
            "http_proxy": "",
            "headers": [],
            "retrieve_mode": "0",
            "request_method": "0",
            "output_format": "0",
            "ssl_cert_file": "",
            "ssl_key_file": "",
```

```
            "ssl_key_password": "",
            "verify_peer": "0",
            "verify_host": "0",
            "allow_traps": "0",
            "discover": "0",
            "uuid": "",
            "lifetime_type": "0",
            "enabled_lifetime_type": "2",
            "enabled_lifetime": "0",
            "parameters": []
        }
    ],
    "id": 1
}
```


Voir aussi

- Prototype de graphique
- Hôte
- Prototype d'élément
- Filtre de prototype de règle LLD
- Prototype de déclencheur

Source

CDiscoveryRulePrototype::get() dans *ui/include/classes/api/services/CDiscoveryRulePrototype.php*.

discoveryruleprototype.update

Description

`object discoveryruleprototype.update(object/array lldRules)`

Cette méthode vous permet de mettre à jour des prototypes de règle LLD existants.

Notez que la mise à jour des prototypes déjà découverts est limitée.

> **Note:**
> Cette méthode est disponible uniquement pour les types d'utilisateur *Admin* et *Super admin*. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres du rôle utilisateur. Consultez Rôles d'utilisateur pour plus d'informations.

Paramètres

`(object/array)` Propriétés des prototypes de règle LLD à mettre à jour.

La propriété `itemid` doit être définie pour chaque prototype de règle LLD, toutes les autres propriétés sont facultatives. Seules les propriétés transmises seront mises à jour, toutes les autres resteront inchangées.

En plus des propriétés standard des prototypes de règle LLD, la méthode accepte les paramètres suivants.

Paramètre	Type	Description
filter	object	Filtre de prototype de règle LLD pour remplacer le filtre existant.
preprocessing	object/array	Options de prétraitement de prototype de règle LLD pour remplacer les options de prétraitement existantes. Comportement du paramètre: - *lecture seule* pour les objets hérités
lld_macro_paths	object/array	Options lld_macro_path du prototype de règle LLD pour remplacer les options lld_macro_path existantes. Comportement du paramètre: - *lecture seule* pour les objets hérités

Paramètre	Type	Description
overrides	object/array	Options de surcharge de prototype de règle LLD pour remplacer les options de surcharge existantes. Comportement du paramètre: - *lecture seule* pour les objets hérités

Valeurs de retour

`(object)` Renvoie un objet contenant les ID des prototypes de règles LLD mis à jour dans la propriété `itemids`.

Exemples

Mise à jour des options de prétraitement d'un prototype de règle LLD

Mettez à jour un prototype de règle LLD avec une règle de prétraitement JSONPath. Requête :

```
{
    "jsonrpc": "2.0",
    "method": "discoveryruleprototype.update",
    "params": {
        "itemid": "47253",
        "preprocessing": [
            {
                "type": 12,
                "params": "$.tablespaces",
                "error_handler": 1
            }
        ]
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "47253"
        ]
    },
    "id": 1
}
```

Source

CDiscoveryRulePrototype::update() dans *ui/include/classes/api/services/CDiscoveryRulePrototype.php*.

Proxy

Cette classe est conçue pour fonctionner avec des proxys.

Références d'objet :

- Proxy

Méthodes disponibles :

- proxy.create - créer de nouveaux proxys
- proxy.delete - supprimer des proxys
- proxy.get - récupérer des proxys
- proxy.update - mettre à jour des proxys

Objet proxy

Les objets suivants sont directement liés à l'API `proxy`.

Proxy

L'objet proxy possède les propriétés suivantes.

Property	Type	Description
proxyid	ID	ID du proxy. Comportement de la propriété : - *en lecture seule* - *obligatoire* pour les opérations de mise à jour
name	string	Nom du proxy. Comportement de la propriété : - *obligatoire* pour les opérations de création
proxy_groupid	ID	ID du groupe de proxys. 0 - le proxy n'est affecté à aucun groupe de proxys.
local_address	string	Adresse pour les agents actifs. Adresse IP ou nom DNS auquel se connecter. Comportement de la propriété : - *obligatoire* si `proxy_groupid` n'est pas égal à 0
local_port	string	Numéro de port local du proxy auquel se connecter. Les macros utilisateur sont prises en charge. Valeur par défaut : 10051. Comportement de la propriété : - *pris en charge* si `proxy_groupid` n'est pas égal à 0
operating_mode	integer	Type de proxy. Valeurs possibles : 0 - proxy actif ; 1 - proxy passif. Comportement de la propriété : - *obligatoire* pour les opérations de création
description	text	Description du proxy.
lastaccess	timestamp	Heure à laquelle le proxy s'est connecté pour la dernière fois au serveur. Comportement de la propriété : - *en lecture seule*
address	string	Adresse IP ou nom DNS auquel se connecter. Les macros utilisateur sont prises en charge. Comportement de la propriété : - *obligatoire* si le mode de fonctionnement du proxy Zabbix est passif
port	string	Numéro de port auquel se connecter. Les macros utilisateur sont prises en charge. Valeur par défaut : 10051. Comportement de la propriété : - *pris en charge* si le mode de fonctionnement du proxy Zabbix est passif
allowed_addresses	string	Adresses IP ou noms DNS, séparés par des virgules, du proxy Zabbix actif.

Property	Type	Description
tls_connect	integer	Connexions au proxy. Valeurs possibles : 1 - *(par défaut)* aucune chiffrement ; 2 - PSK ; 4 - certificat.
tls_accept	integer	Connexions depuis le proxy. Valeurs bitmap possibles : 1 - *(par défaut)* aucune chiffrement ; 2 - PSK ; 4 - certificat. Il s'agit d'un champ de masque de bits ; toute somme de valeurs bitmap possibles est acceptée (par exemple, 6 pour PSK et certificat).
tls_issuer	string	Émetteur du certificat.
tls_subject	string	Sujet du certificat.
tls_psk_identity	string	Identité PSK ; doit être associée à une seule PSK (parmi l'enregistrement automatique, les hôtes et les proxys). N'incluez pas d'informations sensibles dans l'identité PSK, car celle-ci est envoyée sans chiffrement sur le réseau afin d'indiquer au destinataire quelle PSK utiliser. Comportement de la propriété : - *en écriture seule* - *obligatoire* si `tls_connect` est défini sur « PSK » ou si `tls_accept` contient le bit « PSK »
tls_psk	string	Clé prépartagée (PSK) ; doit comporter au moins 32 chiffres hexadécimaux. Comportement de la propriété : - *en écriture seule* - *obligatoire* si `tls_connect` est défini sur « PSK » ou si `tls_accept` contient le bit « PSK »
custom_timeouts	integer	Indique s'il faut remplacer les délais d'expiration globaux des éléments au niveau du proxy. Valeurs possibles : 0 - *(par défaut)* utiliser les paramètres globaux ; 1 - remplacer les délais d'expiration.
timeout_zabbix_agent	string	Ne pas consacrer plus de `timeout_*` secondes au traitement. Accepte un nombre de secondes (par exemple, 1, 60, 300) ou une valeur avec un suffixe de temps (par exemple, `30s`, `1m`). Les macros utilisateur sont également acceptées. Utilisé uniquement si `custom_timeouts` est défini sur « remplacer les délais d'expiration » ; sinon, la valeur est ignorée. Plage de valeurs possibles : 1-600 s. Comportement de la propriété : - *obligatoire* si `custom_timeouts` est défini sur « remplacer les délais d'expiration »
timeout_simple_check		
timeout_snmp_agent		
timeout_external_check		
timeout_db_monitor		
timeout_http_agent		
timeout_ssh_agent		
timeout_telnet_agent		
timeout_script		

Property	Type	Description
timeout_browser		
timeout_telemetry_query		
version	integer	Version du proxy. Numéro de version Zabbix en trois parties, où deux chiffres décimaux sont utilisés pour chaque partie, par exemple 60401 pour la version 6.4.1, 70002 pour la version 7.0.2, etc. 0 - Version du proxy inconnue. Comportement de la propriété : - *en lecture seule*
compatibility	integer	Version du proxy comparée à celle du serveur Zabbix. Valeurs possibles : 0 - Non définie ; 1 - Version actuelle (le proxy et le serveur ont la même version majeure) ; 2 - Version obsolète (la version du proxy est antérieure à celle du serveur, mais elle est partiellement prise en charge) ; 3 - Version non prise en charge (la version du proxy est antérieure à la version de la précédente version LTS du serveur ou la version majeure du serveur est antérieure à celle du proxy). Comportement de la propriété : - *en lecture seule*
state	integer	État du proxy. Valeurs possibles : 0 - Inconnu ; 1 - Hors ligne ; 2 - En ligne. Comportement de la propriété : - *en lecture seule*

Supervision des performances des applications (APM)

L'objet APM possède les propriétés suivantes.

Property	Type	Description
data_collection_status	integer	Indique si la collecte des données de télémétrie est activée sur le proxy. Valeurs possibles : 0 - *(par défaut)* Désactivée ; 1 - Activée.
max_messages_per_second	integer	Nombre maximal de messages par seconde acceptés par le proxy. Plage de valeurs possibles : 0 (illimité) – 2147483647. Valeur par défaut : 0. Comportement de la propriété : - *supported* si `data_collection_status` est défini sur « Enabled »

Property	Type	Description
additional_resource_attributes	array	Tableau d'attributs de ressource à ajouter aux données de télémétrie entrantes. Si les données entrantes contiennent déjà un attribut de ressource portant le même nom, l'attribut entrant est conservé à la place de celui spécifié ici. Comportement de la propriété : - *supported* si `data_collection_status` est défini sur « Enabled »

Attributs de ressource

L'objet des attributs de ressource possède les propriétés suivantes.

Property	Type	Description
key	string	Nom de l'attribut de ressource. Doit être unique pour le `signal_type` spécifié. Comportement de la propriété : - *requis*
value	string	Valeur de l'attribut de ressource. Peut être vide.
signal_type	integer	Type de données de télémétrie auquel l'attribut s'applique. Valeurs possibles : 0 - Traces ; 1 - Métriques ; 2 - Journaux. Comportement de la propriété : - *requis*

proxy.create

Description

`object proxy.create(object/array proxies)`

Cette méthode vous permet de créer de nouveaux proxy.

> **Note:**
> Cette méthode est disponible uniquement pour le type d'utilisateur *Super admin*. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres du rôle utilisateur. Consultez Rôles d'utilisateur pour plus d'informations.

Paramètres

`(object/array)` Proxies à créer.

En plus des propriétés standard des proxies, la méthode accepte les paramètres suivants.

Paramètre	Type	Description
hosts	array	Hôtes à surveiller par le proxy. Si un hôte est déjà surveillé par un autre proxy, il sera réaffecté au proxy actuel. Les hôtes doivent uniquement avoir la propriété `hostid` définie.
apm	object	Paramètres de surveillance des performances des applications (APM) du proxy.

Valeurs de retour

(`object`) Renvoie un objet contenant les ID des proxies créés dans la propriété `proxyids`. L'ordre des ID renvoyés correspond à l'ordre des proxies transmis.

Exemples

Créer un proxy actif

Créez un proxy d'action « Active proxy » et assignez-lui un hôte à surveiller.

Request:

```
{
    "jsonrpc": "2.0",
    "method": "proxy.create",
    "params": {
        "name": "Active proxy",
        "operating_mode": "0",
        "hosts": [
            {
                "hostid": "10279"
            }
        ]
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "proxyids": [
            "10280"
        ]
    },
    "id": 1
}
```

Créer un proxy passif

Créez un proxy passif « Passive proxy » et assignez-lui deux hôtes à surveiller.

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "proxy.create",
    "params": {
        "name": "Passive proxy",
        "operating_mode": "1",
        "address": "127.0.0.1",
        "port": "10051",
        "hosts": [
            {
                "hostid": "10192"
            },
            {
                "hostid": "10139"
            }
        ]
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
```

```
        "proxyids": [
            "10284"
        ]
    },
    "id": 1
}
```

Création d'un proxy et ajout à un groupe de proxys

Créez un proxy actif « Active proxy » et ajoutez-le au groupe de proxys avec l'ID « 1 ».

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "proxy.create",
    "params": {
        "name": "Active proxy",
        "proxy_groupid": "1",
        "operating_mode": "0"
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "proxyids": [
            "5"
        ]
    },
    "id": 1
}
```

Création d'un proxy avec chiffrement basé sur un certificat

Créez un proxy actif avec un chiffrement basé sur un certificat encryption. Le proxy est configuré pour surveiller l'hôte "10785".

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "proxy.create",
    "params": {
        "name": "Active proxy",
        "operating_mode": "0",
        "hosts": [
            {
                "hostid": "10785"
            }
        ],
        "tls_accept": "4",
        "tls_issuer": "CN=Signing CA,OU=Development group,O=Zabbix SIA,DC=zabbix,DC=com",
        "tls_subject": "CN=Zabbix server,OU=Development group,O=Zabbix SIA,DC=zabbix,DC=com"
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "proxyids": [
            "2"
        ]
```

```
    },
    "id": 1
}
```

Création d'un proxy avec APM activé

Créez un proxy actif « Active proxy » avec APM activé. Le proxy accepte jusqu'à 1000 messages par seconde, ajoute l'attribut de ressource `deployment.environment` aux traces et ajoute l'attribut de ressource `cloud.region` aux métriques.

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "proxy.create",
    "params": {
        "name": "Active proxy",
        "operating_mode": "0",
        "apm": {
            "data_collection_status": "1",
            "max_messages_per_second": "1000",
            "additional_resource_attributes": [
                {
                    "key": "deployment.environment",
                    "value": "production",
                    "signal_type": "0"
                },
                {
                    "key": "cloud.region",
                    "value": "eu-central-1",
                    "signal_type": "1"
                }
            ]
        }
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "proxyids": [
            "3"
        ]
    },
    "id": 1
}
```

Voir aussi

- Hôte
- Groupe de proxy

Source

CProxy::create() dans *ui/include/classes/api/services/CProxy.php*.

proxy.delete

Description

`object proxy.delete(array proxies)`

Cette méthode permet de supprimer des proxy.

Note:
Cette méthode est disponible uniquement pour le type d'utilisateur *Super admin*. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres du rôle utilisateur. Consultez User roles pour plus d'informations.

Paramètres

`(array)` Identifiants des proxys à supprimer.

Valeurs de retour

`(object)` Renvoie un objet contenant les ID des proxies supprimés dans la propriété `proxyids`.

Exemples

Supprimer plusieurs proxys

Supprimez deux proxys.

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "proxy.delete",
    "params": [
        "10286",
        "10285"
    ],
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "proxyids": [
            "10286",
            "10285"
        ]
    },
    "id": 1
}
```

Source

CProxy::delete() dans *ui/include/classes/api/services/CProxy.php*.

proxy.get

Description

`integer/array proxy.get(object parameters)`

Cette méthode permet de récupérer des proxys selon les paramètres indiqués.

Note:
Cette méthode est disponible pour les utilisateurs de tout type. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres du rôle utilisateur. Consultez la page Rôles utilisateur pour plus d'informations.
 Seuls les proxys autorisés par la *Liste d'accès aux proxys* des groupes d'utilisateurs de l'utilisateur sont renvoyés.

Paramètres

`(object)` Paramètres définissant le résultat souhaité.

La méthode prend en charge les paramètres suivants.

Paramètre	Type	Description
proxyids	ID/array	Ne retourner que les proxy correspondant aux ID indiqués.

Paramètre	Type	Description
proxy_groupids	ID/array	Ne retourner que les proxy appartenant aux groupes de proxy indiqués.
selectApm	query	Retourner une propriété `apm` contenant les paramètres de surveillance des performances des applications (APM) du proxy.
selectAssignedHosts	query	Retourner une propriété `assignedHosts` contenant les hôtes affectés au proxy. Prend en charge `count`.
selectHosts	query	Retourner une propriété `hosts` contenant les hôtes surveillés par le proxy. Prend en charge `count`.
selectProxyGroup	query	Retourner une propriété `proxyGroup` contenant l'objet du groupe de proxy.
sortfield	string/array	Trier le résultat selon les propriétés indiquées. Valeurs possibles : `proxyid`, `name`, `operating_mode`.
countOutput	boolean	Ces paramètres sont décrits dans le commentaire de référence.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Valeurs de retour

`(integer/array)` Renvoie soit :

- un tableau d'objets ;
- le nombre d'objets récupérés, si le paramètre `countOutput` a été utilisé.

Exemples

Récupérer tous les proxys

Récupérez tous les proxys configurés et leurs interfaces.

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "proxy.get",
    "params": {
        "output": "extend"
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "proxyid": "11",
            "name": "Active proxy",
            "proxy_groupid": "0",
            "local_address": "",
            "local_port": "10051",
            "operating_mode": "0",
```

```
            "description": "",
            "allowed_addresses": "",
            "address": "127.0.0.1",
            "port": "10051",
            "tls_connect": "1",
            "tls_accept": "1",
            "tls_issuer": "",
            "tls_subject": "",
            "custom_timeouts": "0",
            "timeout_zabbix_agent": "",
            "timeout_simple_check": "",
            "timeout_snmp_agent": "",
            "timeout_external_check": "",
            "timeout_db_monitor": "",
            "timeout_http_agent": "",
            "timeout_ssh_agent": "",
            "timeout_telnet_agent": "",
            "timeout_script": "",
            "last_access": "1693391880",
            "version": "70000",
            "compatibility": "1",
            "state": "1"
        },
        {
            "proxyid": "12",
            "name": "Passive proxy",
            "proxy_groupid": "1",
            "local_address": "127.0.0.1",
            "local_port": "10051",
            "operating_mode": "1",
            "description": "",
            "allowed_addresses": "",
            "address": "127.0.0.1",
            "port": "10051",
            "tls_connect": "1",
            "tls_accept": "1",
            "tls_issuer": "",
            "tls_subject": "",
            "custom_timeouts": "1",
            "timeout_zabbix_agent": "5s",
            "timeout_simple_check": "5s",
            "timeout_snmp_agent": "5s",
            "timeout_external_check": "5s",
            "timeout_db_monitor": "5s",
            "timeout_http_agent": "5s",
            "timeout_ssh_agent": "5s",
            "timeout_telnet_agent": "5s",
            "timeout_script": "5s",
            "lastaccess": "1693391875",
            "version": "60400",
            "compatibility": "2",
            "state": "2"
        }
    ],
    "id": 1
}
```


Voir aussi

- Hôte
- Groupe de proxies

Source

CProxy::get() dans *ui/include/classes/api/services/CProxy.php*.

proxy.update

Description

`object proxy.update(object/array proxies)`

Cette méthode vous permet de mettre à jour des proxy existants.

> **Note:**
> Cette méthode est disponible uniquement pour le type d'utilisateur *Super admin*. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres du rôle utilisateur. Consultez Rôles utilisateur pour plus d'informations.

Paramètres

`(object/array)` Propriétés du proxy à mettre à jour.

La propriété `proxyid` doit être définie pour chaque proxy, toutes les autres propriétés sont facultatives. Seules les propriétés transmises seront mises à jour, toutes les autres resteront inchangées.

En plus des propriétés standard du proxy, la méthode accepte les paramètres suivants.

Parameter	Type	Description
hosts	array	Hôtes à surveiller par le proxy. Si un hôte est déjà surveillé par un autre proxy, il sera réaffecté au proxy actuel. Les hôtes doivent avoir uniquement la propriété `hostid` définie.
apm	object	Paramètres de surveillance des performances des applications (APM) du proxy.

Valeurs de retour

`(object)` Renvoie un objet contenant les ID des proxys mis à jour dans la propriété `proxyids`.

Exemples

Modifier les hôtes surveillés par un proxy

Mettez à jour le proxy afin qu'il surveille les deux hôtes indiqués.

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "proxy.update",
    "params": {
        "proxyid": "10293",
        "hosts": [
            {
                "hostid": "10294"
            },
            {
                "hostid": "10295"
            }
        ]
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "proxyids": [
```

```
            "10293"
        ]
    },
    "id": 1
}
```

Modifier le statut du proxy

Modifiez le proxy pour en faire un proxy actif et renommez-le en « Active proxy ».

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "proxy.update",
    "params": {
        "proxyid": "10293",
        "name": "Active proxy",
        "operating_mode": "0"
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "proxyids": [
            "10293"
        ]
    },
    "id": 1
}
```

Ajouter un proxy à un groupe de proxys

Mettez à jour le proxy avec l'ID « 5 » et ajoutez-le au groupe de proxys avec l'ID « 1 ».

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "proxy.create",
    "params": {
        "proxyid": "5",
        "proxy_groupid": "1",
        "local_address": "127.0.0.1"
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "proxyids": [
            "5"
        ]
    },
    "id": 1
}
```

Voir aussi

- Hôte
- Groupe de proxy

Source

CProxy::update() dans *ui/include/classes/api/services/CProxy.php*.

Rapport

Cette classe est conçue pour fonctionner avec les rapports planifiés.

Références d'objet :

- Report
- Users
- User groups

Méthodes disponibles :

- report.create - créer de nouveaux rapports planifiés
- report.delete - supprimer des rapports planifiés
- report.get - récupérer des rapports planifiés
- report.update - mettre à jour des rapports planifiés

Objet rapport

Les objets suivants sont directement liés à l'API `report`.

Rapport

L'objet rapport possède les propriétés suivantes :

Property	Type	Description
reportid	ID	ID du rapport. Comportement de la propriété : - *en lecture seule* - *obligatoire* pour les opérations de mise à jour
userid	ID	ID de l'utilisateur ayant créé le rapport. Comportement de la propriété : - *obligatoire* pour les opérations de création
name	string	Nom unique du rapport. Comportement de la propriété : - *obligatoire* pour les opérations de création
dashboardid	ID	ID du tableau de bord sur lequel le rapport est basé. Comportement de la propriété : - *obligatoire* pour les opérations de création
period	integer	Période pour laquelle le rapport sera préparé. Valeurs possibles : 0 - *(par défaut)* jour précédent ; 1 - semaine précédente ; 2 - mois précédent ; 3 - année précédente.
cycle	integer	Périodicité de répétition. Valeurs possibles : 0 - *(par défaut)* quotidiennement ; 1 - hebdomadairement ; 2 - mensuellement ; 3 - annuellement.
start_time	integer	Heure de la journée, en secondes, à laquelle le rapport sera préparé pour être envoyé. Valeur par défaut : 0.

Property	Type	Description
weekdays	integer	Jours de la semaine pendant lesquels envoyer le rapport. Valeurs bitmap possibles : 1 - lundi ; 2 - mardi ; 4 - mercredi ; 8 - jeudi ; 16 - vendredi ; 32 - samedi ; 64 - dimanche. Il s'agit d'un champ de masque de bits ; toute somme de valeurs bitmap possibles est acceptée (par exemple, 21 pour lundi, mercredi et vendredi). Valeur par défaut : 0. Comportement de la propriété : - *obligatoire* si `cycle` est défini sur « weekly ».
active_since	string	Date à laquelle commencer. Valeurs possibles : chaîne vide - *(par défaut)* non spécifiée (stockée sous la valeur 0) ; date spécifique au format AAAA-MM-JJ (stockée sous la forme d'un horodatage correspondant au début de la journée (00:00:00)).
active_till	string	Date à laquelle terminer. Valeurs possibles : chaîne vide - *(par défaut)* non spécifiée (stockée sous la valeur 0) ; date spécifique au format AAAA-MM-JJ (stockée sous la forme d'un horodatage correspondant à la fin de la journée (23:59:59)).
subject	string	Objet du message du rapport.
message	string	Texte du message du rapport.
status	integer	Indique si le rapport est activé ou désactivé. Valeurs possibles : 0 - *(par défaut)* activé ; 1 - désactivé.
description	text	Description du rapport.
state	integer	État du rapport. Valeurs possibles : 0 - *(par défaut)* le rapport n'a pas encore été traité ; 1 - le rapport a été généré et envoyé avec succès à tous les destinataires ; 2 - la génération du rapport a échoué ; « info » contient les informations relatives à l'erreur ; 3 - le rapport a été généré, mais son envoi à certains destinataires (ou à tous) a échoué ; « info » contient les informations relatives à l'erreur. Comportement de la propriété : - *en lecture seule*
lastsent	timestamp	Horodatage Unix du dernier rapport envoyé avec succès. Comportement de la propriété : - *en lecture seule*
info	string	Description de l'erreur ou informations supplémentaires. Comportement de la propriété : - *en lecture seule*

Utilisateurs

L'objet users possède les propriétés suivantes :

Propriété	Type	Description
userid	ID	ID de l'utilisateur auquel envoyer le rapport. Comportement de la propriété : - *obligatoire*
access_userid	ID	ID de l'utilisateur au nom duquel le rapport sera généré. 0 - *(par défaut)* Générer le rapport par le destinataire.
exclude	integer	Indique s'il faut exclure l'utilisateur de la liste de diffusion. Valeurs possibles : 0 - *(par défaut)* Inclure ; 1 - Exclure.

Groupes d'utilisateurs

L'objet des groupes d'utilisateurs possède les propriétés suivantes :

Property	Type	Description
usrgrpid	ID	ID du groupe d'utilisateurs auquel envoyer le rapport. Comportement de la propriété : - *obligatoire*
access_userid	ID	ID de l'utilisateur au nom duquel le rapport sera généré. 0 - *(par défaut)* Générer le rapport par le destinataire.

report.create

Description

`object report.create(object/array reports)`

Cette méthode permet de créer de nouveaux rapports planifiés.

Note:
Cette méthode est disponible uniquement pour les types d'utilisateur *Admin* et *Super admin*. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres des rôles utilisateur. Voir Rôles utilisateur pour plus d'informations.

Paramètres

`(object/array)` Rapports planifiés à créer.

En plus des propriétés standard des rapports planifiés, la méthode accepte les paramètres suivants.

Paramètre	Type	Description
users	object/array	Utilisateurs auxquels envoyer le rapport. Comportement du paramètre : - *obligatoire* si `user_groups` n'est pas défini
user_groups	object/array	Groupes d'utilisateurs auxquels envoyer le rapport. Comportement du paramètre : - *obligatoire* si `users` n'est pas défini

Valeurs retournées

(`object`) Renvoie un objet contenant les ID des rapports planifiés créés sous la propriété `reportids`. L'ordre des ID renvoyés correspond à l'ordre des rapports planifiés transmis.

Exemples

Création d'un rapport planifié

Créez un rapport hebdomadaire qui sera préparé pour la semaine précédente chaque lundi-vendredi à 12:00 du 2021-04-01 au 2021-08-31.

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "report.create",
    "params": {
        "userid": "1",
        "name": "Weekly report",
        "dashboardid": "1",
        "period": "1",
        "cycle": "1",
        "start_time": "43200",
        "weekdays": "31",
        "active_since": "2021-04-01",
        "active_till": "2021-08-31",
        "subject": "Weekly report",
        "message": "Report accompanying text",
        "status": "1",
        "description": "Report description",
        "users": [
            {
                "userid": "1",
                "access_userid": "1",
                "exclude": "0"
            },
            {
                "userid": "2",
                "access_userid": "0",
                "exclude": "1"
            }
        ],
        "user_groups": [
            {
                "usrgrpid": "7",
                "access_userid": "0"
            }
        ]
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "reportids": [
            "1"
        ]
    },
    "id": 1
}
```

Voir aussi

- Utilisateurs
- Groupes d'utilisateurs

Source

CReport::create() dans *ui/include/classes/api/services/CReport.php*.

report.delete

Description

`object report.delete(array reportids)`

Cette méthode vous permet de supprimer des rapports planifiés.

> **Note:**
> Cette méthode est disponible uniquement pour les utilisateurs de type *Admin* et *Super admin*. Les autorisations d'appel de cette méthode peuvent être révoquées dans les paramètres du rôle utilisateur. Consultez Rôles utilisateur pour plus d'informations.

Paramètres

`(array)` ID des rapports planifiés à supprimer.

Valeurs de retour

`(object)` Renvoie un objet contenant les ID des rapports planifiés supprimés dans la propriété `reportids`.

Exemples

Suppression de plusieurs rapports planifiés

Supprimez deux rapports planifiés.

Request:

```
{
    "jsonrpc": "2.0",
    "method": "report.delete",
    "params": [
        "1",
        "2"
    ],
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "reportids": [
            "1",
            "2"
        ]
    },
    "id": 1
}
```

Source

CReport::delete() dans *ui/include/classes/api/services/CReport.php*.

report.get

Description

`integer/array report.get(object parameters)`

Cette méthode permet de récupérer les rapports planifiés selon les paramètres fournis.

Note:
Cette méthode est disponible pour les utilisateurs de tout type. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres du rôle utilisateur. Consultez Rôles utilisateur pour plus d'informations.

Paramètres

`(object)` Paramètres définissant la sortie souhaitée.

La méthode prend en charge les paramètres suivants.

Paramètre	Type	Description
reportids	ID/array	Retourner uniquement les rapports planifiés avec les ID de rapport indiqués.
expired	boolean	Si défini sur `true`, retourner uniquement les rapports planifiés expirés.
selectUsers	query	Retourner une propriété `users` contenant les utilisateurs auxquels le rapport est configuré pour être envoyé.
selectUserGroups	query	Retourner une propriété `user_groups` contenant les groupes d'utilisateurs auxquels le rapport est configuré pour être envoyé.
sortfield	string/array	Trier le résultat selon les propriétés indiquées. Valeurs possibles : `reportid`, `name`, `status`.
countOutput	boolean	Ces paramètres sont décrits dans le commentaire de référence.
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Valeurs retournées

`(integer/array)` Retourne soit :

- un tableau d'objets ;
- le nombre d'objets récupérés, si le paramètre `countOutput` a été utilisé.

Exemples

Récupération des données des rapports

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "report.get",
    "params": {
        "output": "extend",
        "selectUsers": "extend",
        "selectUserGroups": "extend",
        "reportids": ["1", "2"]
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "reportid": "1",
            "userid": "1",
```

```
            "name": "Rapport hebdomadaire",
            "dashboardid": "1",
            "period": "1",
            "cycle": "1",
            "start_time": "43200",
            "weekdays": "31",
            "active_since": "2021-04-01",
            "active_till": "2021-08-31",
            "subject": "Rapport hebdomadaire",
            "message": "Texte accompagnant le rapport",
            "status": "1",
            "description": "Description du rapport",
            "state": "1",
            "lastsent": "1613563219",
            "info": "",
            "users": [
                {
                    "userid": "1",
                    "access_userid": "1",
                    "exclude": "0"
                },
                {
                    "userid": "2",
                    "access_userid": "0",
                    "exclude": "1"
                }
            ],
            "user_groups": [
                {
                    "usrgrpid": "7",
                    "access_userid": "0"
                }
            ]
        },
        {
            "reportid": "2",
            "userid": "1",
            "name": "Rapport mensuel",
            "dashboardid": "2",
            "period": "2",
            "cycle": "2",
            "start_time": "0",
            "weekdays": "0",
            "active_since": "2021-05-01",
            "active_till": "",
            "subject": "Rapport mensuel",
            "message": "Texte accompagnant le rapport",
            "status": "1",
            "description": "",
            "state": "0",
            "lastsent": "0",
            "info": "",
            "users": [
                {
                    "userid": "1",
                    "access_userid": "1",
                    "exclude": "0"
                }
            ],
            "user_groups": []
        }
    ],
```

```
    "id": 1
}
```

Voir aussi

- Utilisateurs
- Groupes d'utilisateurs

Source

CReport::get() dans *ui/include/classes/api/services/CReport.php*.

report.update

Description

`object report.update(object/array reports)`

Cette méthode permet de mettre à jour des rapports planifiés existants.

> **Note:**
> Cette méthode est disponible uniquement pour les types d'utilisateur *Admin* et *Super admin*. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres du rôle utilisateur. Voir Rôles utilisateur pour plus d'informations.

Paramètres

`(object/array)` Propriétés du rapport planifié à mettre à jour.

La propriété `reportid` doit être définie pour chaque rapport planifié, toutes les autres propriétés sont facultatives. Seules les propriétés transmises seront mises à jour, toutes les autres resteront inchangées.

En plus des propriétés standard du rapport planifié, la méthode accepte les paramètres suivants.

Paramètre	Type	Description
users	object/array	Utilisateurs remplaçant les utilisateurs actuellement attribués au rapport planifié. Comportement des paramètres : - *obligatoire* si `user_groups` n'est pas défini
user_groups	object/array	Groupes d'utilisateurs remplaçant les groupes d'utilisateurs actuellement attribués au rapport planifié. Comportement des paramètres : - *obligatoire* si `users` n'est pas défini

Valeurs retournées

`(object)` Retourne un objet contenant les identifiants des rapports planifiés mis à jour sous la propriété `reportids`.

Exemples

Désactivation d'un rapport planifié

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "report.update",
    "params": {
        "reportid": "1",
        "status": "0"
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "reportids": [
            "1"
        ]
    },
    "id": 1
}
```

Voir aussi

- Utilisateurs
- Groupes d'utilisateurs

Source

CReport::update() dans *ui/include/classes/api/services/CReport.php*.

Règle de découverte

Cette classe est conçue pour fonctionner avec les règles de découverte réseau.

> **Note:**
> Cette API est destinée à fonctionner avec les règles de découverte réseau. Pour les règles de découverte de bas niveau, voir API des règles LLD.

Références d'objet :

- Règle de découverte

Méthodes disponibles :

- drule.create - créer de nouvelles règles de découverte réseau
- drule.delete - supprimer des règles de découverte réseau
- drule.get - récupérer des règles de découverte réseau
- drule.update - mettre à jour des règles de découverte réseau

Objet de règle de découverte

Les objets suivants sont directement liés à l'API `drule`.

Règle de découverte

L'objet de règle de découverte définit une règle de découverte réseau. Il possède les propriétés suivantes.

Property	Type	Description
druleid	ID	ID de la règle de découverte. Comportement de la propriété : - *lecture seule* - *obligatoire* pour les opérations de mise à jour
iprange	string	Une ou plusieurs plages d'adresses IP à vérifier, séparées par des virgules. Consultez la section configuration de la découverte réseau pour plus d'informations sur les formats de plages d'adresses IP pris en charge. Comportement de la propriété : - *obligatoire* pour les opérations de création
name	string	Nom de la règle de découverte. Comportement de la propriété : - *obligatoire* pour les opérations de création

Property	Type	Description
delay	string	Intervalle d'exécution de la règle de découverte. Accepte un nombre de secondes (par ex. 1, 60, 300), une valeur avec un suffixe temporel (par ex. 30s, 1m, 2h, 1d) ou une macro utilisateur. Valeur par défaut : 1h.
proxyid	ID	ID du proxy utilisé pour la découverte.
status	integer	Indique si la règle de découverte est activée. Valeurs possibles : 0 - *(par défaut)* activée ; 1 - désactivée.
concurrency_max	integer	Nombre maximal de vérifications simultanées par règle de découverte. Valeurs possibles : 0 - *(par défaut)* nombre illimité de vérifications ; 1 - une vérification ; 2-999 - nombre personnalisé de vérifications.
error	string	Texte de l'erreur si des problèmes sont survenus lors de l'exécution de la règle de découverte. Comportement de la propriété : - *lecture seule*

drule.create

Description

`object drule.create(object/array discoveryRules)`

Cette méthode permet de créer de nouvelles règles de découverte.

> **Note:**
> Cette méthode est disponible uniquement pour les types d'utilisateur *Admin* et *Super admin*. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres des rôles utilisateur. Consultez Rôles utilisateur pour plus d'informations.

Paramètres

`(object/array)` Règles de découverte à créer.

En plus des propriétés standard des règles de découverte, la méthode accepte les paramètres suivants.

Paramètre	Type	Description
dchecks	array	Contrôles de découverte à créer pour la règle de découverte. Comportement des paramètres: - *obligatoire*

Valeurs retournées

`(object)` Retourne un objet contenant les identifiants des règles de découverte créées sous la propriété `druleids`. L'ordre des ID retournés correspond à l'ordre des règles de découverte passées.

Exemples

Créer une règle de découverte

Créez une règle de découverte pour trouver les machines exécutant l'agent Zabbix sur le réseau local. La règle doit utiliser une seule vérification de l'agent Zabbix sur le port 10050.

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "drule.create",
    "params": {
        "name": "Zabbix agent discovery",
        "iprange": "192.168.1.1-255",
        "concurrency_max": "10",
        "dchecks": [
            {
                "type": "9",
                "key_": "system.uname",
                "ports": "10050",
                "uniq": "0"
            }
        ]
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "druleids": [
            "6"
        ]
    },
    "id": 1
}
```

Voir aussi

- Vérification de découverte

Source

CDRule::create() dans *ui/include/classes/api/services/CDRule.php*.

drule.delete

Description

`object drule.delete(array discoveryRuleIds)`

Cette méthode vous permet de supprimer des règles de découverte.

> **Note:**
> Cette méthode est disponible uniquement pour les types d'utilisateur *Admin* et *Super admin*. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres des rôles utilisateur. Consultez Rôles utilisateur pour plus d'informations.

Paramètres

`(array)` IDs des règles de découverte à supprimer.

Valeurs de retour

`(object)` Renvoie un objet contenant les ID des règles de découverte supprimées dans la propriété `druleids`.

Exemples

Supprimer plusieurs règles de découverte

Supprimez deux règles de découverte.

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "drule.delete",
    "params": [
        "4",
        "6"
    ],
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "druleids": [
            "4",
            "6"
        ]
    },
    "id": 1
}
```

Source

CDRule::delete() dans *ui/include/classes/api/services/CDRule.php*.

drule.get

Description

`integer/array drule.get(object parameters)`

La méthode permet de récupérer les règles de découverte selon les paramètres fournis.

> **Note:**
> Cette méthode est disponible pour les utilisateurs de tout type. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres des rôles utilisateur. Consultez Rôles utilisateur pour plus d'informations.

Paramètres

`(object)` Paramètres définissant la sortie souhaitée.

La méthode prend en charge les paramètres suivants.

Paramètre	Type	Description
dhostids	ID/array	Retourner uniquement les règles de découverte qui ont créé les hôtes découverts indiqués.
druleids	ID/array	Retourner uniquement les règles de découverte avec les ID indiqués.
dserviceids	ID/array	Retourner uniquement les règles de découverte qui ont créé les services découverts indiqués.
selectDChecks	query	Retourner une propriété `dchecks` avec les vérifications de découverte utilisées par la règle de découverte. Prend en charge `count`.
selectDHosts	query	Retourner une propriété `dhosts` avec les hôtes découverts créés par la règle de découverte. Prend en charge `count`.
limitSelects	integer	Limite le nombre d'enregistrements renvoyés par les sous-sélections. S'applique aux sous-sélections suivantes : `selectDChecks` - les résultats seront triés par `dcheckid` ; `selectDHosts` - les résultats seront triés par `dhostsid`.

Paramètre	Type	Description
sortfield	string/array	Trier le résultat selon les propriétés indiquées. Valeurs possibles : `druleid`, `name`.
countOutput	boolean	Ces paramètres sont décrits dans le commentaire de référence.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Valeurs retournées

`(entier/tableau)` Renvoie soit :

- un tableau d'objets ;
- le nombre d'objets récupérés, si le paramètre `countOutput` a été utilisé.

Exemples

Récupérer toutes les règles de découverte

Récupérez toutes les règles de découverte configurées ainsi que les contrôles de découverte qu'elles utilisent.

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "drule.get",
    "params": {
        "output": "extend",
        "selectDChecks": "extend"
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "druleid": "2",
            "proxyid": "0",
            "name": "Local network",
            "iprange": "192.168.3.1-255",
            "delay": "5s",
            "status": "0",
            "concurrency_max": "0",
            "error": "",
            "dchecks": [
                {
                    "dcheckid": "7",
                    "druleid": "2",
                    "type": "3",
                    "key_": "",
                    "snmp_community": "",
                    "ports": "21",
                    "snmpv3_securityname": "",
                    "snmpv3_securitylevel": "0",
```

```
                "snmpv3_authpassphrase": "",
                "snmpv3_privpassphrase": "",
                "uniq": "0",
                "snmpv3_authprotocol": "0",
                "snmpv3_privprotocol": "0",
                "snmpv3_contextname": "",
                "host_source": "1",
                "name_source": "0",
                "allow_redirect": "0"
            },
            {
                "dcheckid": "8",
                "druleid": "2",
                "type": "4",
                "key_": "",
                "snmp_community": "",
                "ports": "80",
                "snmpv3_securityname": "",
                "snmpv3_securitylevel": "0",
                "snmpv3_authpassphrase": "",
                "snmpv3_privpassphrase": "",
                "uniq": "0",
                "snmpv3_authprotocol": "0",
                "snmpv3_privprotocol": "0",
                "snmpv3_contextname": "",
                "host_source": "1",
                "name_source": "0",
                "allow_redirect": "0"
            }
        ]
    },
    {
        "druleid": "6",
        "proxyid": "0",
        "name": "Zabbix agent discovery",
        "iprange": "192.168.1.1-255",
        "delay": "1h",
        "status": "0",
        "concurrency_max": "10",
        "error": "",
        "dchecks": [
            {
                "dcheckid": "10",
                "druleid": "6",
                "type": "9",
                "key_": "system.uname",
                "snmp_community": "",
                "ports": "10050",
                "snmpv3_securityname": "",
                "snmpv3_securitylevel": "0",
                "snmpv3_authpassphrase": "",
                "snmpv3_privpassphrase": "",
                "uniq": "0",
                "snmpv3_authprotocol": "0",
                "snmpv3_privprotocol": "0",
                "snmpv3_contextname": "",
                "host_source": "2",
                "name_source": "3",
                "allow_redirect": "0"
            }
        ]
    }
```

```
    ],
    "id": 1
}
```

Voir aussi

- Hôte découvert
- Vérification de découverte

Source

CDRule::get() dans *ui/include/classes/api/services/CDRule.php*.

drule.update

Description

`object drule.update(object/array discoveryRules)`

Cette méthode permet de mettre à jour des règles de découverte existantes.

> **Note:**
> Cette méthode est disponible uniquement pour les types d'utilisateur *Admin* et *Super admin*. Les autorisations d'appel de cette méthode peuvent être révoquées dans les paramètres des rôles utilisateur. Voir Rôles utilisateur pour plus d'informations.

Paramètres

`(object/array)` Propriétés des règles de découverte à mettre à jour.

La propriété `druleid` doit être définie pour chaque règle de découverte, toutes les autres propriétés sont facultatives. Seules les propriétés transmises seront mises à jour, toutes les autres resteront inchangées.

En plus des propriétés standard des règles de découverte, la méthode accepte les paramètres suivants.

Paramètre	Type	Description
dchecks	array	Vérifications de découverte pour remplacer les vérifications existantes.

Valeurs retournées

`(object)` Retourne un objet contenant les identifiants des règles de découverte mises à jour sous la propriété `druleids`.

Exemples

Modifier la plage d'adresses IP d'une règle de découverte

Modifiez la plage d'adresses IP d'une règle de découverte en "192.168.2.1-255".

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "drule.update",
    "params": {
        "druleid": "6",
        "iprange": "192.168.2.1-255"
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "druleids": [
            "6"
        ]
```

```
    },
    "id": 1
}
```

Voir aussi

- Vérification de découverte

Source

CDRule::update() dans *ui/include/classes/api/services/CDRule.php*.

Règle LLD

Cette classe est conçue pour fonctionner avec les règles de découverte de bas niveau.

Références d'objet :

- Règle LLD
  - En-tête HTTP
  - Champ de requête HTTP
- Filtre de règle LLD
  - Condition de filtre de règle LLD
- Chemin de macro LLD
- Prétraitement de règle LLD
- Remplacements de règle LLD
  - Filtre de remplacement de règle LLD
    - Condition de filtre de remplacement de règle LLD
  - Opération de remplacement de règle LLD
    - Statut d'opération de remplacement de règle LLD
    - Découverte d'opération de remplacement de règle LLD
    - Période d'opération de remplacement de règle LLD
    - Historique d'opération de remplacement de règle LLD
    - Tendances d'opération de remplacement de règle LLD
    - Sévérité d'opération de remplacement de règle LLD
    - Tag d'opération de remplacement de règle LLD
    - Modèle d'opération de remplacement de règle LLD
    - Inventaire d'opération de remplacement de règle LLD

Méthodes disponibles :

- discoveryrule.create - créer de nouvelles règles LLD
- discoveryrule.delete - supprimer des règles LLD
- discoveryrule.get - récupérer des règles LLD
- discoveryrule.update - mettre à jour des règles LLD

Objet de règle LLD

Les objets suivants sont directement liés à l'API `discoveryrule`.

Règle LLD

L'objet de règle de découverte de bas niveau possède les propriétés suivantes.

Propriété	Type	Description
itemid	ID	ID de la règle LLD. Comportement de la propriété : - *en lecture seule* - *obligatoire* pour les opérations de mise à jour

Propriété	Type	Description
delay	string	Intervalle de mise à jour de la règle LLD. Accepte un nombre de secondes (par exemple, 1, 60, 300), une valeur avec un suffixe temporel (par exemple, 30s, 1m, 2h, 1d) ou, facultativement, un ou plusieurs intervalles personnalisés, séparés par des points-virgules. Les intervalles personnalisés peuvent combiner des intervalles flexibles et des intervalles de planification. Accepte les macros utilisateur. Si elle est utilisée, la valeur doit être une macro unique. Les macros multiples ou les macros combinées à du texte ne sont pas prises en charge. Les intervalles flexibles peuvent être écrits sous la forme de deux macros séparées par une barre oblique (par exemple, {$FLEX_INTERVAL}/{$FLEX_PERIOD}). Exemple : 1h;wd1-5h9-18;{$Macro1}/1-7,00:00-24:00;0/6-7,12:00-24:00;{$Macr Comportement de la propriété : - *obligatoire* si type est défini sur « Zabbix agent » (0), « Simple check » (3), « Zabbix internal » (5), « External check » (10), « Database monitor » (11), « IPMI agent » (12), « SSH agent » (13), « TELNET agent » (14), « JMX agent » (16), « HTTP agent » (19), « SNMP agent » (20), « Script » (21), « Browser » (22), ou si type est défini sur « Zabbix agent (active) » (7) et que key_ ne contient pas « mqtt.get »
hostid	ID	ID de l'hôte auquel appartient la règle LLD. Comportement de la propriété : - *constante* - *obligatoire* pour les opérations de création
flags	integer	Origine de la règle de découverte. Valeurs possibles : 1 - règle de découverte de bas niveau ; 5 - règle de découverte de bas niveau convertie à partir d'un prototype. Comportement de la propriété : - *en lecture seule*
interfaceid	ID	ID de l'interface de l'hôte de la règle LLD. Comportement de la propriété : - *obligatoire* si la règle LLD appartient à un hôte et que type est défini sur « Zabbix agent », « IPMI agent », « JMX agent » ou « SNMP agent » - *pris en charge* si la règle LLD appartient à un hôte et que type est défini sur « Simple check », « External check », « SSH agent », « TELNET agent » ou « HTTP agent »
key_	string	Clé de la règle LLD. Comportement de la propriété : - *obligatoire* pour les opérations de création - *en lecture seule* pour les objets hérités
name	string	Nom de la règle LLD. Comportement de la propriété : - *obligatoire* pour les opérations de création - *en lecture seule* pour les objets hérités

Propriété	Type	Description
type	integer	Type de la règle LLD. Valeurs possibles : 0 - Zabbix agent ; 2 - Zabbix trapper ; 3 - Simple check ; 5 - Zabbix internal ; 7 - Zabbix agent (active) ; 10 - External check ; 11 - Database monitor ; 12 - IPMI agent ; 13 - SSH agent ; 14 - TELNET agent ; 16 - JMX agent ; 18 - Dependent item ; 19 - HTTP agent ; 20 - SNMP agent ; 21 - Script ; 22 - Browser ; 23 - Nested. Ce type est autorisé uniquement si le parent de la règle est un modèle (supposé être lié à un prototype d'hôte lors de la découverte) ou un hôte découvert (qui possède une règle LLD à sa racine). Comportement de la propriété : - *obligatoire* pour les opérations de création - *en lecture seule* pour les objets hérités
url	string	Chaîne d'URL. Prend en charge les macros utilisateur, {HOST.IP}, {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.NAME}, {HOST.PORT}, {ITEM.ID}, {ITEM.KEY}. Comportement de la propriété : - *obligatoire* si `type` est défini sur « HTTP agent » - *en lecture seule* pour les objets hérités
allow_traps	integer	Autorise le remplissage de la valeur de la même manière que pour l'élément trapper. Valeurs possibles : 0 - *(par défaut)* Ne pas accepter les données entrantes ; 1 - Accepter les données entrantes. Comportement de la propriété : - *pris en charge* si `type` est défini sur « HTTP agent »
authtype	integer	Méthode d'authentification. Valeurs possibles si `type` est défini sur « SSH agent » : 0 - *(par défaut)* mot de passe ; 1 - clé publique. Valeurs possibles si `type` est défini sur « HTTP agent » : 0 - *(par défaut)* aucune ; 1 - basic ; 2 - NTLM ; 3 - Kerberos ; 4 - Digest. Comportement de la propriété : - *pris en charge* si `type` est défini sur « SSH agent » ou « HTTP agent » - *en lecture seule* pour les objets hérités (si `type` est défini sur « HTTP agent »)

Propriété	Type	Description
description	string	Description de la règle LLD.
error	string	Texte d'erreur en cas de problème lors de la mise à jour de la valeur de la règle LLD. Comportement de la propriété : - *en lecture seule*
follow_redirects	integer	Suit les redirections de la réponse lors de l'interrogation des données. Valeurs possibles : 0 - Ne pas suivre les redirections ; 1 - *(par défaut)* Suivre les redirections. Comportement de la propriété : - *pris en charge* si `type` est défini sur « HTTP agent » - *en lecture seule* pour les objets hérités
headers	array	Tableau d'en-têtes envoyés lors de l'exécution d'une requête HTTP. Comportement de la propriété : - *pris en charge* si `type` est défini sur « HTTP agent » - *en lecture seule* pour les objets hérités
http_proxy	string	Chaîne de connexion du proxy HTTP(S). Comportement de la propriété : - *pris en charge* si `type` est défini sur « HTTP agent » - *en lecture seule* pour les objets hérités
ipmi_sensor	string	Capteur IPMI. Comportement de la propriété : - *obligatoire* si `type` est défini sur « IPMI agent » et que `key_` n'est pas défini sur « ipmi.get » - *pris en charge* si `type` est défini sur « IPMI agent » - *en lecture seule* pour les objets hérités
jmx_endpoint	string	Chaîne de connexion personnalisée de l'agent JMX. Valeur par défaut : service:jmx:rmi:///jndi/rmi://{HOST.CONN}:{HOST.PORT}/jmxrmi Comportement de la propriété : - *pris en charge* si `type` est défini sur « JMX agent »
lifetime	string	Période après laquelle les éléments qui ne sont plus découverts seront supprimés. Accepte un nombre de secondes (par exemple, 3600), une valeur avec un suffixe temporel (par exemple, `3600s`, `180m`, `24h`, `1d`) ou une macro utilisateur. Valeur par défaut : `7d`.
lifetime_type	integer	Scénario de suppression des ressources LLD perdues. Valeurs possibles : 0 - *(par défaut)* Supprimer une fois le seuil de durée de vie atteint ; 1 - Ne pas supprimer ; 2 - Supprimer immédiatement.
enabled_lifetime	string	Période après laquelle les éléments qui ne sont plus découverts seront désactivés. Accepte un nombre de secondes (par exemple, `1`, `60`, `300`), une valeur avec un suffixe temporel (par exemple, `30s`, `1m`, `2h`, `1d`) ou une macro utilisateur. Valeur par défaut : `0`.

Propriété	Type	Description
enabled_lifetime_type	integer	Scénario de désactivation des ressources LLD perdues. Valeurs possibles : 0 - Désactiver une fois le seuil de durée de vie atteint ; 1 - Ne pas désactiver ; 2 - *(par défaut)* Désactiver immédiatement.
master_itemid	ID	ID de l'élément maître. Une règle de découverte ne peut pas être l'élément maître d'une autre règle de découverte. Comportement de la propriété : - *obligatoire* si `type` est défini sur « Dependent item » - *en lecture seule* pour les objets hérités
output_format	integer	Indique si la réponse doit être convertie en JSON. Valeurs possibles : 0 - *(par défaut)* Stocker les données brutes ; 1 - Convertir en JSON. Comportement de la propriété : - *pris en charge* si `type` est défini sur « HTTP agent » - *en lecture seule* pour les objets hérités
params	string	Paramètres supplémentaires selon le type de la règle LLD : - script exécuté pour les règles LLD SSH et Telnet ; - requête SQL pour les règles LLD de surveillance de bases de données ; - formule pour les règles LLD calculées ; - script pour les règles LLD de type script et navigateur. Comportement de la propriété : - *obligatoire* si `type` est défini sur « Database monitor », « SSH agent », « TELNET agent », « Script » ou « Browser » - *en lecture seule* pour les objets hérités (si `type` est défini sur « Script » ou « Browser »)
parameters	object/array	Paramètres supplémentaires si `type` est défini sur « Script » ou « Browser ». Tableau d'objets possédant les propriétés `name` et `value`, où `name` doit être unique. Comportement de la propriété : - *pris en charge* si `type` est défini sur « Script » ou « Browser » - *en lecture seule* pour les objets hérités
password	string	Mot de passe d'authentification. Comportement de la propriété : - *obligatoire* si `type` est défini sur « JMX agent » et que `username` est défini - *pris en charge* si `type` est défini sur « Simple check », « Database monitor », « SSH agent », « TELNET agent » ou « HTTP agent » - *en lecture seule* pour les objets hérités (si `type` est défini sur « HTTP agent »)
post_type	integer	Type de corps des données POST stocké dans la propriété `posts`. Valeurs possibles : 0 - *(par défaut)* Données brutes ; 2 - Données JSON ; 3 - Données XML. Comportement de la propriété : - *pris en charge* si `type` est défini sur « HTTP agent » - *en lecture seule* pour les objets hérités

Propriété	Type	Description
posts	string	Données du corps de la requête HTTP(S). Comportement de la propriété : - *obligatoire* si `type` est défini sur « HTTP agent » et que `post_type` est défini sur « Données JSON » ou « Données XML » - *pris en charge* si `type` est défini sur « HTTP agent » et que `post_type` est défini sur « Données brutes » - *en lecture seule* pour les objets hérités
privatekey	string	Nom du fichier de clé privée. Comportement de la propriété : - *obligatoire* si `type` est défini sur « SSH agent » et que `authtype` est défini sur « clé publique »
publickey	string	Nom du fichier de clé publique. Comportement de la propriété : - *obligatoire* si `type` est défini sur « SSH agent » et que `authtype` est défini sur « clé publique »
query_fields	array	Tableau de champs de requête envoyés lors de l'exécution d'une requête HTTP. Comportement de la propriété : - *pris en charge* si `type` est défini sur « HTTP agent » - *en lecture seule* pour les objets hérités
request_method	integer	Type de méthode de requête. Valeurs possibles : 0 - *(par défaut)* GET ; 1 - POST ; 2 - PUT ; 3 - HEAD. Comportement de la propriété : - *pris en charge* si `type` est défini sur « HTTP agent » - *en lecture seule* pour les objets hérités
retrieve_mode	integer	Partie de la réponse à stocker. Valeurs possibles si `request_method` est défini sur « GET », « POST » ou « PUT » : 0 - *(par défaut)* Corps ; 1 - En-têtes ; 2 - Le corps et les en-têtes seront stockés. Valeurs possibles si `request_method` est défini sur « HEAD » : 1 - En-têtes. Comportement de la propriété : - *pris en charge* si `type` est défini sur « HTTP agent » - *en lecture seule* pour les objets hérités
snmp_oid	string	OID SNMP. Comportement de la propriété : - *obligatoire* si `type` est défini sur « SNMP agent » - *en lecture seule* pour les objets hérités
ssl_cert_file	string	Chemin du fichier de clé publique SSL. Comportement de la propriété : - *pris en charge* si `type` est défini sur « HTTP agent » - *en lecture seule* pour les objets hérités

Propriété	Type	Description
ssl_key_file	string	Chemin du fichier de clé privée SSL. Comportement de la propriété : - *pris en charge* si `type` est défini sur « HTTP agent » - *en lecture seule* pour les objets hérités
ssl_key_password	string	Mot de passe du fichier de clé SSL. Comportement de la propriété : - *pris en charge* si `type` est défini sur « HTTP agent » - *en lecture seule* pour les objets hérités
state	integer	État de la règle LLD. Valeurs possibles : 0 - *(par défaut)* normale ; 1 - non prise en charge. Comportement de la propriété : - *en lecture seule*
status	integer	Statut de la règle LLD. Valeurs possibles : 0 - *(par défaut)* règle LLD activée ; 1 - règle LLD désactivée.
status_codes	string	Plages de codes d'état HTTP requis, séparées par des virgules. Prend également en charge les macros utilisateur dans une liste séparée par des virgules. Exemple : 200,200-{$M},{$M},200-400 Comportement de la propriété : - *pris en charge* si `type` est défini sur « HTTP agent » - *en lecture seule* pour les objets hérités
templateid	ID	ID de la règle LLD du modèle parent. Comportement de la propriété : - *en lecture seule*
timeout	string	Délai d'expiration de la requête d'interrogation des données de l'élément. Accepte un nombre de secondes (par exemple, 1, 60, 300) ou une valeur avec un suffixe temporel (par exemple, 30s, 1m). Accepte également les macros utilisateur. Plage de valeurs possibles : 1-600s. Valeur par défaut : "" - utiliser les paramètres du proxy ou les paramètres globaux. Comportement de la propriété : - *pris en charge* si `type` est défini sur « Zabbix agent » (0), « Simple check » (3) et que `key_` ne commence pas par « `vmware.` » ni « `icmpping` », « Zabbix agent (active) » (7), « External check » (10), « Database monitor » (11), « SSH agent » (13), « TELNET agent » (14), « HTTP agent » (19), « SNMP agent » (20) et que `snmp_oid` commence par « `walk[` » ou « `get[` », « Script » (21), « Browser » (22) - *en lecture seule* pour les objets hérités, - si la valeur par défaut est appliquée et que l'élément appartient à un hôte surveillé par un proxy ou un groupe de proxies inaccessible, le délai d'expiration global est utilisé.

Propriété	Type	Description
trapper_hosts	string	Hôtes autorisés. Comportement de la propriété : - *pris en charge* si `type` est défini sur « Zabbix trapper », ou si `type` est défini sur « HTTP agent » et que `allow_traps` est défini sur « Allow incoming data ». Si la propriété est omise, elle est automatiquement définie sur la macro utilisateur `{$TRAPPER.ALLOWED_HOSTS}` (valeur par défaut). Par défaut, `{$TRAPPER.ALLOWED_HOSTS}` vaut `127.0.0.1,::1`. Si la propriété est définie sur une chaîne vide, les connexions entrantes seront refusées pour tous les hôtes.
username	string	Nom d'utilisateur pour l'authentification. Comportement de la propriété : - *obligatoire* si `type` est défini sur « SSH agent », « TELNET agent », ou si `type` est défini sur « JMX agent » et que `password` est défini - *pris en charge* si `type` est défini sur « Simple check », « Database monitor » ou « HTTP agent » - *en lecture seule* pour les objets hérités (si `type` est défini sur « HTTP agent »)
uuid	string	Identifiant universel unique utilisé pour lier les règles LLD importées à celles qui existent déjà. Généré automatiquement s'il n'est pas fourni. Comportement de la propriété : - *pris en charge* si la règle LLD appartient à un modèle
verify_host	integer	Indique s'il faut vérifier que le nom d'hôte de la connexion correspond à celui indiqué dans le certificat de l'hôte. Valeurs possibles : 0 - *(par défaut)* Ne pas vérifier ; 1 - Vérifier. Comportement de la propriété : - *pris en charge* si `type` est défini sur « HTTP agent » - *en lecture seule* pour les objets hérités
verify_peer	integer	Indique s'il faut vérifier que le certificat de l'hôte est authentique. Valeurs possibles : 0 - *(par défaut)* Ne pas vérifier ; 1 - Vérifier. Comportement de la propriété : - *pris en charge* si `type` est défini sur « HTTP agent » - *en lecture seule* pour les objets hérités

En-tête HTTP

L'objet d'en-tête possède les propriétés suivantes :

Propriété	Type	Description
name	string	Nom de l'en-tête HTTP. Comportement de la propriété : - *requis*

Propriété	Type	Description
value	string	Valeur de l'en-tête. Comportement de la propriété : - *requis*

Champ de requête HTTP

L'objet de champ de requête définit un nom et une valeur utilisés pour spécifier un paramètre d'URL. Il possède les propriétés suivantes :

Propriété	Type	Description
name	string	Nom du paramètre. Comportement de la propriété : - *obligatoire*
value	string	Valeur du paramètre. Comportement de la propriété : - *obligatoire*

Filtre de règle LLD

L'objet de filtre de règle LLD définit un ensemble de conditions pouvant être utilisées pour filtrer les objets découverts. Il possède les propriétés suivantes :

Propriété	Type	Description
conditions	object/array	Ensemble de conditions de filtre à utiliser pour filtrer les résultats. Les conditions seront triées dans l'ordre de leur placement dans la formule. Comportement de la propriété : - *obligatoire*
evaltype	integer	Méthode d'évaluation des conditions de filtre. Valeurs possibles : 0 - Et/Ou ; 1 - Et ; 2 - Ou ; 3 - Expression personnalisée. Comportement de la propriété : - *obligatoire*
eval_formula	string	Expression générée qui sera utilisée pour évaluer les conditions de filtre. L'expression contient des ID qui font référence à des conditions de filtre spécifiques via leur `formulaid`. La valeur de `eval_formula` est égale à la valeur de `formula` pour les filtres avec une expression personnalisée. Comportement de la propriété : - *lecture seule*

Propriété	Type	Description
formula	string	Expression définie par l'utilisateur à utiliser pour évaluer les conditions des filtres avec une expression personnalisée. L'expression doit contenir des ID qui font référence à des conditions de filtre spécifiques via leur `formulaid`. Les ID utilisés dans l'expression doivent correspondre exactement à ceux définis dans les conditions de filtre : aucune condition ne peut rester inutilisée ou omise. Comportement de la propriété : - *obligatoire* si `evaltype` est défini sur "expression personnalisée"

Condition de filtre de règle LLD

L'objet de condition de filtre de règle LLD définit une vérification distincte à effectuer sur la valeur d'une macro LLD. Il possède les propriétés suivantes :

Propriété	Type	Description
macro	string	Macro LLD sur laquelle effectuer la vérification. Comportement de la propriété : - *obligatoire*
value	string	Valeur à comparer. Comportement de la propriété : - *obligatoire* si `operator` est défini sur « correspond à l'expression régulière » ou « ne correspond pas à l'expression régulière »
formulaid	string	ID unique arbitraire utilisé pour référencer la condition depuis une expression personnalisée. Ne peut contenir que des lettres majuscules. L'ID doit être défini par l'utilisateur lors de la modification des conditions de filtre, mais sera généré à nouveau lors de leur demande ultérieure. Comportement de la propriété : - *obligatoire* si `evaltype` de l'objet de filtre de règle LLD est défini sur « expression personnalisée »
operator	integer	Opérateur de condition. Valeurs possibles : 8 - *(par défaut)* correspond à l'expression régulière ; 9 - ne correspond pas à l'expression régulière ; 12 - existe ; 13 - n'existe pas.

Note:
Pour mieux comprendre comment utiliser les filtres avec différents types d'expressions, consultez les exemples sur les pages des méthodes discoveryrule.get et discoveryrule.create.

Chemin de macro LLD

Le chemin de macro LLD possède les propriétés suivantes :

Propriété	Type	Description
lld_macro	string	Macro LLD. Comportement de la propriété : - *obligatoire*

Propriété	Type	Description
path	string	Sélecteur de la valeur qui sera attribuée à la macro correspondante. Comportement de la propriété : - *obligatoire*

Prétraitement de la règle LLD

L'objet de prétraitement de la règle LLD possède les propriétés suivantes.

Propriété	Type	Description
type	integer	Type d'option de prétraitement. Valeurs possibles : 5 - Expression régulière ; 11 - XPath XML ; 12 - JSONPath ; 14 - Correspond à une expression régulière ; 15 - Ne correspond pas à une expression régulière ; 16 - Vérifier la présence d'une erreur dans JSON ; 17 - Vérifier la présence d'une erreur dans XML ; 20 - Ignorer les valeurs inchangées avec heartbeat ; 21 - JavaScript ; 23 - Prometheus vers JSON ; 24 - CSV vers JSON ; 25 - Remplacer ; 27 - XML vers JSON ; 28 - Valeur de parcours SNMP ; 29 - Parcours SNMP vers JSON ; 30 - Valeur obtenue via SNMP. Comportement de la propriété : - *obligatoire*
params	string	Paramètres supplémentaires utilisés par l'option de prétraitement. Plusieurs paramètres sont séparés par le caractère de nouvelle ligne (\n). Comportement de la propriété : - *obligatoire* si `type` est défini sur « Expression régulière » (5), « XPath XML » (11), « JSONPath » (12), « Correspond à une expression régulière » (14), « Ne correspond pas à une expression régulière » (15), « Vérifier la présence d'une erreur dans JSON » (16), « Vérifier la présence d'une erreur dans XML » (17), « Ignorer les valeurs inchangées avec heartbeat » (20), « JavaScript » (21), « Prometheus vers JSON » (23), « CSV vers JSON » (24), « Remplacer » (25), « Valeur de parcours SNMP » (28), « Parcours SNMP vers JSON » (29) ou « Valeur obtenue via SNMP » (30)

Propriété	Type	Description
error_handler	integer	Type d'action utilisé en cas d'échec de l'étape de prétraitement. Valeurs possibles : 0 - Le message d'erreur est défini par le serveur Zabbix ; 1 - Ignorer la valeur ; 2 - Définir une valeur personnalisée ; 3 - Définir un message d'erreur personnalisé. Comportement de la propriété : - *obligatoire* si `type` est défini sur « Expression régulière » (5), « XPath XML » (11), « JSONPath » (12), « Correspond à une expression régulière » (14), « Ne correspond pas à une expression régulière » (15), « Vérifier la présence d'une erreur dans JSON » (16), « Vérifier la présence d'une erreur dans XML » (17), « Prometheus vers JSON » (23), « CSV vers JSON » (24), « XML vers JSON » (27), « Valeur de parcours SNMP » (28), « Parcours SNMP vers JSON » (29) ou « Valeur obtenue via SNMP » (30)
error_handler_params	string	Paramètres du gestionnaire d'erreurs. Comportement de la propriété : - *obligatoire* si `error_handler` est défini sur « Définir une valeur personnalisée » ou « Définir un message d'erreur personnalisé »

Les paramètres et gestionnaires d'erreurs suivants sont pris en charge pour chaque type de prétraitement.

Type de prétraitement	Nom	Paramètre 1	Paramètre 2	Paramètre 3	Gestionnaires d'erreurs pris en charge
5	Expression régulière	pattern[1]	output[2]		0, 1, 2, 3
11	XPath XML	path[3]			0, 1, 2, 3
12	JSONPath	path[3]			0, 1, 2, 3
14	Correspond à une expression régulière	pattern[1]			0, 1, 2, 3
15	Ne correspond pas à une expression régulière	pattern[1]			0, 1, 2, 3
16	Vérifier la présence d'une erreur dans JSON	path[3]			0, 1, 2, 3

Type de prétraitement	Nom	Paramètre 1	Paramètre 2	Paramètre 3	Gestionnaires d'erreurs pris en charge
17	Vérifier la présence d'une erreur dans XML	path[3]			0, 1, 2, 3
20	Ignorer les valeurs inchangées avec heartbeat	seconds[4], 5^			
21	JavaScript	script[2]			
23	Prometheus vers JSON	pattern[6]			0, 1, 2, 3
24	CSV vers JSON	character[2]	character[2]	0,1	0, 1, 2, 3
25	Remplacer	search string[2]	replacement[2]		
27	XML vers JSON				0, 1, 2, 3
28	Valeur de parcours SNMP	OID[2]	Format : 0 - Inchangé 1 - UTF-8 depuis Hex-STRING 2 - MAC depuis Hex-STRING 3 - Entier depuis BITS		0, 1, 2, 3
29	Parcours SNMP vers JSON[7]	Field name[2]	OID prefix[2]	Format : 0 - Inchangé 1 - UTF-8 depuis Hex-STRING 2 - MAC depuis Hex-STRING 3 - Entier depuis BITS	0, 1, 2, 3
30	Valeur obtenue via SNMP	Format : 1 - UTF-8 depuis Hex-STRING 2 - MAC depuis Hex-STRING 3 - Entier depuis BITS			0, 1, 2, 3

[1] expression régulière
 [2] chaîne
 [3] JSONPath ou XPath XML
 [4] entier positif (avec prise en charge des suffixes temporels, par exemple 30s, 1m, 2h, 1d)
 [5] macro utilisateur
 [6] modèle Prometheus suivant la syntaxe : `<metric name>{<label name>="<label value>", ...} == <value>`. Chaque composant du modèle Prometheus (métrique, nom de libellé, valeur de libellé et valeur de métrique) peut être une macro utilisateur.
 [7] Prend en charge plusieurs enregistrements « Field name,OID prefix,Format » délimités par un caractère de nouvelle ligne.

Remplacements de règle LLD

L'objet de remplacements de règle LLD définit un ensemble de règles (filtres, conditions et opérations) utilisées pour remplacer les propriétés de différents objets prototypes. Il possède les propriétés suivantes :

Propriété	Type	Description
name	string	Nom de remplacement unique. Comportement de la propriété : - *obligatoire*
step	integer	Numéro d'ordre unique du remplacement. Comportement de la propriété : - *obligatoire*
stop	integer	Arrêter le traitement des remplacements suivants en cas de correspondance. Valeurs possibles : 0 - *(par défaut)* ne pas arrêter le traitement des remplacements ; 1 - arrêter le traitement des remplacements si le filtre correspond.
filter	object	Filtre de remplacement.
operations	object/array	Opérations de remplacement.

Filtre de remplacement de règle LLD

L'objet de filtre de remplacement de règle LLD définit un ensemble de conditions qui, si elles correspondent à l'objet découvert, appliquent le remplacement. Il possède les propriétés suivantes :

Propriété	Type	Description
conditions	object/array	Ensemble de conditions de filtre de remplacement à utiliser pour la correspondance avec les objets découverts. Les conditions seront triées dans l'ordre de leur placement dans la formule. Comportement de la propriété : - *obligatoire*
evaltype	integer	Méthode d'évaluation des conditions du filtre de remplacement. Valeurs possibles : 0 - Et/Ou ; 1 - Et ; 2 - Ou ; 3 - Expression personnalisée. Comportement de la propriété : - *obligatoire*
eval_formula	string	Expression générée qui sera utilisée pour évaluer les conditions du filtre de remplacement. L'expression contient des ID qui référencent des conditions spécifiques du filtre de remplacement par leur `formulaid`. La valeur de `eval_formula` est égale à la valeur de `formula` pour les filtres avec une expression personnalisée. Comportement de la propriété : - *lecture seule*

Propriété	Type	Description
formula	string	Expression définie par l'utilisateur à utiliser pour évaluer les conditions des filtres de remplacement avec une expression personnalisée. L'expression doit contenir des ID qui référencent des conditions spécifiques du filtre de remplacement par leur `formulaid`. Les ID utilisés dans l'expression doivent correspondre exactement à ceux définis dans les conditions du filtre de remplacement : aucune condition ne peut rester inutilisée ou omise. Comportement de la propriété : - *obligatoire* si `evaltype` est défini sur "expression personnalisée"

Condition de filtre de substitution de règle LLD

L'objet de condition de filtre de substitution de règle LLD définit une vérification distincte à effectuer sur la valeur d'une macro LLD. Il possède les propriétés suivantes :

Propriété	Type	Description
macro	string	Macro LLD sur laquelle effectuer la vérification. Comportement de la propriété : - *obligatoire*
value	string	Valeur à comparer. Comportement de la propriété : - *obligatoire* si `operator` est défini sur « correspond à l'expression régulière » ou « ne correspond pas à l'expression régulière »
formulaid	string	ID unique arbitraire utilisé pour référencer la condition depuis une expression personnalisée. Ne peut contenir que des lettres majuscules. L'ID doit être défini par l'utilisateur lors de la modification des conditions de filtre, mais sera généré à nouveau lors de leur demande ultérieure. Comportement de la propriété : - *obligatoire* si `evaltype` de l'objet de filtre de substitution de règle LLD est défini sur « expression personnalisée »
operator	integer	Opérateur de condition. Valeurs possibles : 8 - *(par défaut)* correspond à l'expression régulière ; 9 - ne correspond pas à l'expression régulière ; 12 - existe ; 13 - n'existe pas.

Opération de remplacement de règle LLD

L'opération de remplacement de règle LLD est une combinaison de conditions et d'actions à effectuer sur l'objet prototype. Elle possède les propriétés suivantes :

Propriété	Type	Description
operationobject	integer	Type d'objet découvert sur lequel effectuer l'action. Valeurs possibles : 0 - Prototype d'élément ; 1 - Prototype de déclencheur ; 2 - Prototype de graphique ; 3 - Prototype d'hôte ; 4 - Prototype de découverte. Comportement de la propriété : - *obligatoire*
operator	integer	Opérateur de condition de remplacement. Valeurs possibles : 0 - *(par défaut)* égal à ; 1 - différent de ; 2 - contient ; 3 - ne contient pas ; 8 - correspond ; 9 - ne correspond pas.
value	string	Motif à faire correspondre au nom du prototype d'élément, de déclencheur, de graphique ou d'hôte selon l'objet sélectionné.
opstatus	object	Objet d'état d'opération de remplacement pour les objets prototype d'élément, de déclencheur et d'hôte.
opdiscover	object	Objet d'état de découverte d'opération de remplacement (tous les types d'objet).
opperiod	object	Objet de période d'opération de remplacement (intervalle de mise à jour) pour l'objet prototype d'élément.
ophistory	object	Objet d'historique d'opération de remplacement pour l'objet prototype d'élément.
optrends	object	Objet de tendances d'opération de remplacement pour l'objet prototype d'élément.
opseverity	object	Objet de sévérité d'opération de remplacement pour l'objet prototype de déclencheur.
optag	object/array	Objet de balise d'opération de remplacement pour les objets prototype de déclencheur et d'hôte.
optemplate	object/array	Objet de modèle d'opération de remplacement pour l'objet prototype d'hôte.
opinventory	object	Objet d'inventaire d'opération de remplacement pour l'objet prototype d'hôte.

Statut de l'opération de remplacement de règle LLD

Statut de l'opération de remplacement de règle LLD défini sur l'objet découvert. Il possède les propriétés suivantes :

Propriété	Type	Description
status	integer	Remplace le statut de l'objet sélectionné. Valeurs possibles : 0 - Créer activé ; 1 - Créer désactivé. Comportement de la propriété : - *obligatoire*

Découverte de l'opération de remplacement de règle LLD

Statut de découverte de l'opération de remplacement de règle LLD défini pour l'objet découvert. Elle possède les propriétés suivantes :

Propriété	Type	Description
discover	integer	Remplace le statut de découverte pour l'objet sélectionné. Valeurs possibles : 0 - Oui, continuer à découvrir les objets ; 1 - Non, les nouveaux objets ne seront pas découverts et les objets existants seront marqués comme perdus. Comportement de la propriété : - *obligatoire*

Période de l'opération de substitution de la règle LLD

La période de l'opération de substitution de la règle LLD est une valeur d'intervalle de mise à jour définie pour l'élément découvert. Elle possède les propriétés suivantes :

Property	Type	Description
delay	string	Remplace l'intervalle de mise à jour du prototype d'élément. Accepte un nombre de secondes (par exemple, `1`, `60`, `300`), une valeur avec un suffixe temporel (par exemple, `30s`, `1m`, `2h`, `1d`) ou, éventuellement, un ou plusieurs intervalles personnalisés, tous séparés par des points-virgules. Les intervalles personnalisés peuvent être une combinaison d'intervalles flexibles et d'intervalles de planification. Accepte les macros utilisateur ou les macros LLD. Si elle est utilisée, la valeur doit être une macro unique. Plusieurs macros ou des macros combinées à du texte ne sont pas pris en charge. Les intervalles flexibles peuvent être écrits sous la forme de deux macros séparées par une barre oblique (par exemple, `{$FLEX_INTERVAL}/{$FLEX_PERIOD}`). Exemple : `1h;wd1-5h9-18;{$Macro1}/1-7,00:00-24:00;0/6-7,12:00-24:00;{$Macr` Comportement de la propriété : - *required*

Historique des opérations de substitution de règle LLD

Valeur de l'historique des opérations de substitution de règle LLD définie pour l'élément découvert. Elle possède les propriétés suivantes :

Propriété	Type	Description
history	string	Remplace l'historique du prototype d'élément, qui est une unité de temps indiquant combien de temps les données d'historique doivent être conservées. Accepte également une macro utilisateur et une macro LLD. Comportement de la propriété : - *obligatoire*

Tendances de l'opération de substitution de règle LLD

Valeur des tendances de l'opération de substitution de règle LLD définie pour l'élément découvert. Elle possède les propriétés suivantes :

Propriété	Type	Description
trends	string	Remplace les tendances du prototype d'élément, c'est-à-dire l'unité de temps pendant laquelle les données de tendances doivent être conservées. Accepte également une macro utilisateur et une macro LLD. Comportement de la propriété : - *obligatoire*

Gravité de l'opération de substitution de règle LLD

Valeur de gravité de l'opération de substitution de règle LLD définie pour le déclencheur découvert. Elle possède les propriétés suivantes :

Propriété	Type	Description
severity	integer	Remplace la gravité du prototype de déclencheur. Valeurs possibles : 0 - *(par défaut)* non classé ; 1 - information ; 2 - avertissement ; 3 - moyenne ; 4 - élevée ; 5 - sinistre. Comportement de la propriété : - *obligatoire*

Balise d'opération de remplacement de règle LLD

L'objet balise d'opération de remplacement de règle LLD contient le nom et la valeur de la balise qui sont définis pour l'objet découvert. Il possède les propriétés suivantes :

Property	Type	Description
tag	string	Nouveau nom de balise. Accepte également une macro utilisateur et une macro LLD. Comportement de la propriété: - *required*
value	string	Nouvelle valeur de balise. Accepte également une macro utilisateur et une macro LLD.

Modèle d'opération de substitution de règle LLD

Objet de modèle d'opération de substitution de règle LLD lié à l'hôte découvert.
Il possède les propriétés suivantes :

Propriété	Type	Description
templateid	ID	Remplace le modèle des modèles liés du prototype d'hôte. Comportement de la propriété: - *obligatoire*

Valeur du mode d'inventaire de l'opération de surcharge de règle LLD

Valeur du mode d'inventaire de l'opération de surcharge de règle LLD définie sur l'hôte découvert. Elle possède les propriétés suivantes :

Propriété	Type	Description
inventory_mode	integer	Surcharge le mode d'inventaire du prototype d'hôte. Valeurs possibles : -1 - désactivé ; 0 - *(par défaut)* manuel ; 1 - automatique. Comportement de la propriété : - *obligatoire*

discoveryrule.create

Description

`object discoveryrule.create(object/array lldRules)`

Cette méthode permet de créer de nouvelles règles LLD.

Note:
Cette méthode est disponible uniquement pour les types d'utilisateur *Admin* et *Super admin*. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres des rôles utilisateur. Consultez Rôles utilisateur pour plus d'informations.

Paramètres

`(object/array)` Règles LLD à créer.

En plus des propriétés standard des règles LLD, la méthode accepte les paramètres suivants.

Paramètre	Type	Description
filter	object	Filtre de règle LLD pour la règle LLD.
preprocessing	object/array	Options de prétraitement de règle LLD.
lld_macro_paths	object/array	Options de lld_macro_path de la règle LLD.
overrides	object/array	Options de surcharges de règle LLD.

Valeurs retournées

`(object)` Retourne un objet contenant les identifiants des règles de découverte bas niveau créées sous la propriété `itemids`. L'ordre des ID retournés correspond à l'ordre des règles de découverte bas niveau transmises.

Exemples

Création d'une règle LLD

Créez une règle LLD d'agent Zabbix pour découvrir les systèmes de fichiers montés. Les éléments découverts seront mis à jour toutes les 30 secondes.

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "discoveryrule.create",
    "params": {
        "name": "Mounted filesystem discovery",
        "key_": "vfs.fs.discovery",
        "hostid": "10197",
        "type": 0,
        "interfaceid": "112",
        "delay": "30s"
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "27665"
        ]
    },
    "id": 1
}
```

Utilisation d'un filtre

Créez une règle LLD avec un ensemble de conditions permettant de filtrer les résultats. Les conditions seront regroupées à l'aide de l'opérateur logique « and ».

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "discoveryrule.create",
    "params": {
        "name": "Filtered LLD rule",
        "key_": "lld",
        "hostid": "10116",
        "type": 0,
        "interfaceid": "13",
        "delay": "30s",
        "filter": {
            "evaltype": 1,
            "conditions": [
                {
                    "macro": "{#MACRO1}",
                    "value": "@regex1"
                },
                {
                    "macro": "{#MACRO2}",
                    "value": "@regex2",
                    "operator": "9"
                },
                {
                    "macro": "{#MACRO3}",
                    "value": "",
                    "operator": "12"
                },
                {
                    "macro": "{#MACRO4}",
                    "value": "",
                    "operator": "13"
                }
            ]
        }
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "27665"
        ]
    },
    "id": 1
```

```
}
```

Création d'une règle LLD avec des chemins de macro

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryrule.create",
    "params": {
        "name": "Règle LLD avec chemins de macro LLD",
        "key_": "lld",
        "hostid": "10116",
        "type": 0,
        "interfaceid": "13",
        "delay": "30s",
        "lld_macro_paths": [
            {
                "lld_macro": "{#MACRO1}",
                "path": "$.path.1"
            },
            {
                "lld_macro": "{#MACRO2}",
                "path": "$.path.2"
            }
        ]
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "27665"
        ]
    },
    "id": 1
}
```

Utilisation d'un filtre d'expression personnalisé

Créez une règle LLD avec un filtre qui utilisera une expression personnalisée pour évaluer les conditions. La règle LLD ne doit découvrir que les objets dont la valeur de macro "{#MACRO1}" correspond à la fois aux expressions régulières "regex1" et "regex2", et dont la valeur de "{#MACRO2}" correspond soit à "regex3", soit à "regex4". Les identifiants de formule "A", "B", "C" et "D" ont été choisis arbitrairement.

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryrule.create",
    "params": {
        "name": "Filtered LLD rule",
        "key_": "lld",
        "hostid": "10116",
        "type": 0,
        "interfaceid": "13",
        "delay": "30s",
        "filter": {
            "evaltype": 3,
            "formula": "(A and B) and (C or D)",
            "conditions": [
                {
```

```
                    "macro": "{#MACRO1}",
                    "value": "@regex1",
                    "formulaid": "A"
                },
                {
                    "macro": "{#MACRO1}",
                    "value": "@regex2",
                    "formulaid": "B"
                },
                {
                    "macro": "{#MACRO2}",
                    "value": "@regex3",
                    "formulaid": "C"
                },
                {
                    "macro": "{#MACRO2}",
                    "value": "@regex4",
                    "formulaid": "D"
                }
            ]
        }
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "27665"
        ]
    },
    "id": 1
}
```

Utilisation de champs de requête et d'en-têtes personnalisés

Créez une règle LLD avec des champs de requête et des en-têtes personnalisés.

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryrule.create",
    "params": {
        "hostid": "10257",
        "interfaceid": "5",
        "type": 19,
        "name": "API HTTP agent",
        "key_": "api_discovery_rule",
        "delay": "5s",
        "url": "http://127.0.0.1?discoverer.php",
        "query_fields": [
            {
                "name": "mode",
                "value": "json"
            },
            {
                "name": "elements",
                "value": "2"
            }
        ],
        "headers": [
```

```
            {
                "name": "X-Type",
                "value": "api"
            },
            {
                "name": "Authorization",
                "value": "Bearer mF_A.B5f-2.1JcM"
            }
        ],
        "allow_traps": 1,
        "trapper_hosts": "127.0.0.1"
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "28336"
        ]
    },
    "id": 1
}
```

Création d'une règle LLD avec prétraitement

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryrule.create",
    "params": {
        "name": "Règle de découverte avec prétraitement",
        "key_": "lld.with.preprocessing",
        "hostid": "10001",
        "ruleid": "27665",
        "type": 0,
        "delay": "60s",
        "interfaceid": "1155",
        "preprocessing": [
            {
                "type": 20,
                "params": "20",
                "error_handler": 0,
                "error_handler_params": ""
            }
        ]
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "44211"
        ]
    },
    "id": 1
}
```

Création d'une règle LLD avec des remplacements

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryrule.create",
    "params": {
        "name": "Discover database host",
        "key_": "lld.with.overrides",
        "hostid": "10001",
        "type": 0,
        "delay": "60s",
        "interfaceid": "1155",
        "overrides": [
            {
                "name": "Discover MySQL host",
                "step": "1",
                "stop": "1",
                "filter": {
                    "evaltype": "2",
                    "conditions": [
                        {
                            "macro": "{#UNIT.NAME}",
                            "operator": "8",
                            "value": "^mysqld\\.service$"
                        },
                        {
                            "macro": "{#UNIT.NAME}",
                            "operator": "8",
                            "value": "^mariadb\\.service$"
                        }
                    ]
                },
                "operations": [
                    {
                        "operationobject": "3",
                        "operator": "2",
                        "value": "Database host",
                        "opstatus": {
                            "status": "0"
                        },
                        "optemplate": [
                            {
                                "templateid": "10170"
                            }
                        ],
                        "optag": [
                            {
                                "tag": "database",
                                "value": "mysql"
                            }
                        ]
                    }
                ]
            },
            {
                "name": "Discover PostgreSQL host",
                "step": "2",
                "stop": "1",
                "filter": {
                    "evaltype": "0",
                    "conditions": [
                        {
```

```
                            "macro": "{#UNIT.NAME}",
                            "operator": "8",
                            "value": "^postgresql\\.service$"
                        }
                    ]
                },
                "operations": [
                    {
                        "operationobject": "3",
                        "operator": "2",
                        "value": "Database host",
                        "opstatus": {
                            "status": "0"
                        },
                        "optemplate": [
                            {
                                "templateid": "10263"
                            }
                        ],
                        "optag": [
                            {
                                "tag": "database",
                                "value": "postgresql"
                            }
                        ]
                    }
                ]
            }
        ]
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "30980"
        ]
    },
    "id": 1
}
```

Créer une règle LLD de script

Créez une collecte de données simple à l'aide d'une règle LLD de script.

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryrule.create",
    "params": {
        "name": "Script example",
        "key_": "custom.script.lldrule",
        "hostid": "12345",
        "type": 21,
        "params": "var request = new HttpRequest();\nreturn request.post(\"https://postman-echo.com/post\"
        "parameters": [{
            "name": "host",
            "value": "{HOST.CONN}"
        }],
        "timeout": "6s",
```

```
        "delay": "30s"
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "23865"
        ]
    },
    "id": 1
}
```

Créer une règle LLD avec une période spécifiée pour la désactivation et sans suppression

Créez une règle LLD avec une période personnalisée pour désactiver l'entité après qu'elle ne soit plus découverte, avec le paramètre indiquant qu'elle ne sera jamais supprimée.

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryrule.create",
    "params": {
        "name": "lld disable after 1h",
        "key_": "lld.disable",
        "hostid": "10001",
        "type": 2,
        "lifetime_type": 1,
        "enabled_lifetime_type": 0,
        "enabled_lifetime": "1h"
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "46864"
        ]
    },
    "id": 1
}
```

Voir aussi

- Filtre de règle LLD
- Chemins de macro LLD
- Prétraitement de règle LLD

Source

CDiscoveryRule::create() dans *ui/include/classes/api/services/CDiscoveryRule.php*.

discoveryrule.delete

Description

`object discoveryrule.delete(array lldRuleIds)`

Cette méthode permet de supprimer des règles de découverte de bas niveau (LLD).

Note:
Cette méthode est uniquement disponible pour les types d'utilisateurs *Admin* et *Super admin*. Les autorisations d'appeler cette méthode peuvent être révoquées dans les paramètres des rôles utilisateur. Consultez la page Rôles utilisateur pour plus d'informations.
 L'utilisateur ne pourra pas supprimer une règle LLD qui découvre des entités à l'aide d'un proxy inaccessible. Consultez la page Configuration des groupes d'utilisateurs (*Liste d'accès aux proxy*) pour plus d'informations.

Paramètres

`(array)` ID des règles de découverte bas niveau à supprimer.

Valeurs de retour

`(object)` Renvoie un objet contenant les ID des règles LLD supprimées dans la propriété `ruleids`.

Exemples

Suppression de plusieurs règles LLD

Supprimez deux règles LLD.

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "discoveryrule.delete",
    "params": [
        "27665",
        "27668"
    ],
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "ruleids": [
            "27665",
            "27668"
        ]
    },
    "id": 1
}
```

Source

CDiscoveryRule::delete() dans *ui/include/classes/api/services/CDiscoveryRule.php*.

discoveryrule.get

Description

`integer/array discoveryrule.get(object parameters)`

La méthode permet de récupérer les règles LLD selon les paramètres fournis.

Note:
Cette méthode est disponible pour les utilisateurs de tout type. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres des rôles utilisateur. Consultez Rôles utilisateur pour plus d'informations.

Paramètres

`(object)` Paramètres définissant la sortie souhaitée.

La méthode prend en charge les paramètres suivants.

Paramètre	Type	Description
itemids	ID/array	Retourner uniquement les règles LLD avec les ID donnés.
groupids	ID/array	Retourner uniquement les règles LLD qui appartiennent aux hôtes des groupes donnés.
hostids	ID/array	Retourner uniquement les règles LLD qui appartiennent aux hôtes donnés.
inherited	boolean	Si défini sur `true`, retourner uniquement les règles LLD héritées d'un modèle.
interfaceids	ID/array	Retourner uniquement les règles LLD qui utilisent les interfaces d'hôte données.
monitored	boolean	Si défini sur `true`, retourner uniquement les règles LLD activées qui appartiennent à des hôtes surveillés.
templated	boolean	Si défini sur `true`, retourner uniquement les règles LLD qui appartiennent à des modèles.
templateids	ID/array	Retourner uniquement les règles LLD qui appartiennent aux modèles donnés.
selectDiscoveryData	query	Retourner une propriété `discoveryData` avec les données de l'objet de découverte de la règle LLD. L'objet de découverte de la règle LLD relie une règle LLD découverte à un prototype de règle LLD à partir duquel elle a été découverte.
selectDiscoveryRulePrototypes	query	Retourner une propriété `discoveryRulePrototypes` avec les prototypes de règle LLD qui appartiennent à la règle LLD. Prend en charge `count`.
selectFilter	query	Retourner une propriété `filter` avec les données du filtre utilisé par la règle LLD.
selectGraphs	query	Retourne une propriété `graphs` avec les prototypes de graphique qui appartiennent à la règle LLD. Prend en charge `count`.
selectHostPrototypes	query	Retourner une propriété `hostPrototypes` avec les prototypes d'hôte qui appartiennent à la règle LLD. Prend en charge `count`.
selectHosts	query	Retourner une propriété `hosts` avec un tableau d'hôtes auxquels la règle LLD appartient.
selectItems	query	Retourner une propriété `items` avec les prototypes d'élément qui appartiennent à la règle LLD. Prend en charge `count`.
selectTriggers	query	Retourner une propriété `triggers` avec les prototypes de déclencheur qui appartiennent à la règle LLD. Prend en charge `count`.
selectLLDMacroPaths	query	Retourner une propriété `lld_macro_paths` avec une liste de macros LLD et les chemins vers les valeurs attribuées à chaque macro correspondante.
selectPreprocessing	query	Retourner une propriété `preprocessing` avec les options de prétraitement de la règle LLD.
selectOverrides	query	Retourner une propriété `lld_rule_overrides` avec une liste de filtres de substitution, de conditions et d'opérations exécutées sur les objets prototypes.
filter	object	Retourner uniquement les résultats qui correspondent exactement au filtre donné. Accepte un objet, où les clés sont des noms de propriété et les valeurs sont soit une valeur unique, soit un tableau de valeurs à comparer. Ne prend pas en charge les propriétés de type de données `text` data type. Prend en charge des propriétés supplémentaires : `host` - nom technique de l'hôte auquel la règle LLD appartient.

Paramètre	Type	Description
limitSelects	integer	Limite le nombre d'enregistrements renvoyés par les sous-sélections. S'applique aux sous-sélections suivantes : `selectItems`, `selectGraphs`, `selectTriggers`.
sortfield	string/array	Trier le résultat selon les propriétés données. Valeurs possibles : `itemid`, `name`, `key_`, `delay`, `type`, `status`.
countOutput	boolean	Ces paramètres sont décrits dans le commentaire de référence.
editable	boolean	
excludeSearch	boolean	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Valeurs de retour

`(integer/array)` Renvoie soit :

- un tableau d'objets
- le nombre d'objets récupérés, si le paramètre `countOutput` a été utilisé.

Exemples

Récupération des règles de découverte d'un hôte

Récupérez toutes les règles de découverte pour un ID d'hôte spécifique.

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryrule.get",
    "params": {
        "output": "extend",
        "hostids": "10202"
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "itemid": "27425",
            "type": "0",
            "snmp_oid": "",
            "hostid": "10202",
            "name": "Découverte des interfaces réseau",
            "key_": "net.if.discovery",
            "delay": "1h",
            "status": "0",
            "trapper_hosts": "",
            "templateid": "22444",
            "valuemapid": "0",
            "params": "",
            "ipmi_sensor": "",
            "authtype": "0",
            "username": "",
```

```
            "password": "",
            "publickey": "",
            "privatekey": "",
            "flags": "1",
            "interfaceid": "119",
            "description": "Découverte des interfaces réseau telles que définies dans l'expression régulièr
            "lifetime": "30d",
            "jmx_endpoint": "",
            "master_itemid": "0",
            "timeout": "",
            "url": "",
            "query_fields": [],
            "posts": "",
            "status_codes": "200",
            "follow_redirects": "1",
            "post_type": "0",
            "http_proxy": "",
            "headers": [],
            "retrieve_mode": "0",
            "request_method": "0",
            "output_format": "0",
            "ssl_cert_file": "",
            "ssl_key_file": "",
            "ssl_key_password": "",
            "verify_peer": "0",
            "verify_host": "0",
            "allow_traps": "0",
            "uuid": "",
            "lifetime_type": "0",
            "enabled_lifetime_type": "2",
            "enabled_lifetime": "0",
            "state": "0",
            "error": "",
            "parameters": []
        },
        {
            "itemid": "27426",
            "type": "0",
            "snmp_oid": "",
            "hostid": "10202",
            "name": "Découverte des systèmes de fichiers montés",
            "key_": "vfs.fs.discovery",
            "delay": "1h",
            "status": "0",
            "trapper_hosts": "",
            "templateid": "22450",
            "valuemapid": "0",
            "params": "",
            "ipmi_sensor": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "flags": "1",
            "interfaceid": "119",
            "description": "Découverte des systèmes de fichiers de différents types tels que définis dans
            "lifetime": "30d",
            "jmx_endpoint": "",
            "master_itemid": "0",
            "timeout": "",
            "url": "",
```

```
            "query_fields": [],
            "posts": "",
            "status_codes": "200",
            "follow_redirects": "1",
            "post_type": "0",
            "http_proxy": "",
            "headers": [],
            "retrieve_mode": "0",
            "request_method": "0",
            "output_format": "0",
            "ssl_cert_file": "",
            "ssl_key_file": "",
            "ssl_key_password": "",
            "verify_peer": "0",
            "verify_host": "0",
            "allow_traps": "0",
            "uuid": "",
            "lifetime_type": "0",
            "enabled_lifetime_type": "2",
            "enabled_lifetime": "0",
            "state": "0",
            "error": "",
            "parameters": []
        }
    ],
    "id": 1
}
```

Récupération des conditions de filtre

Récupérez le nom de la règle LLD « 24681 » et ses conditions de filtre. Le filtre utilise le type d'évaluation « and », donc la propriété `formula` est vide et `eval_formula` est générée automatiquement.

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryrule.get",
    "params": {
        "output": ["name"],
        "selectFilter": "extend",
        "itemids": ["24681"]
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "itemid": "24681",
            "name": "Filtered LLD rule",
            "filter": {
                "evaltype": "1",
                "formula": "",
                "conditions": [
                    {
                        "macro": "{#MACRO1}",
                        "value": "@regex1",
                        "operator": "8",
                        "formulaid": "A"
                    },
                    {
```

```
                        "macro": "{#MACRO2}",
                        "value": "@regex2",
                        "operator": "9",
                        "formulaid": "B"
                    },
                    {
                        "macro": "{#MACRO3}",
                        "value": "",
                        "operator": "12",
                        "formulaid": "C"
                    },
                    {
                        "macro": "{#MACRO4}",
                        "value": "",
                        "operator": "13",
                        "formulaid": "D"
                    }
                ],
                "eval_formula": "A and B and C and D"
            }
        }
    ],
    "id": 1
}
```

Récupérer une règle LLD par URL

Récupérez la règle LLD d'un hôte à partir de la valeur du champ URL de la règle. Seule la correspondance exacte de la chaîne URL définie pour la règle LLD est prise en charge.

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "discoveryrule.get",
    "params": {
        "hostids": "10257",
        "filter": {
            "type": 19,
            "url": "http://127.0.0.1/discoverer.php"
        }
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "itemid": "28336",
            "type": "19",
            "snmp_oid": "",
            "hostid": "10257",
            "name": "API HTTP agent",
            "key_": "api_discovery_rule",
            "delay": "5s",
            "status": "0",
            "trapper_hosts": "",
            "templateid": "0",
            "valuemapid": "0",
            "params": "",
            "ipmi_sensor": "",
            "authtype": "0",
```

```
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "flags": "1",
            "interfaceid": "5",
            "description": "",
            "lifetime": "30d",
            "jmx_endpoint": "",
            "master_itemid": "0",
            "timeout": "",
            "url": "http://127.0.0.1/discoverer.php",
            "query_fields": [
                {
                    "name": "mode",
                    "value": "json"
                },
                {
                    "name": "elements",
                    "value": "2"
                }
            ],
            "posts": "",
            "status_codes": "200",
            "follow_redirects": "1",
            "post_type": "0",
            "http_proxy": "",
            "headers": [
                {
                    "name" : "X-Type",
                    "value": "api"
                },
                {
                    "name": "Authorization",
                    "value": "Bearer mF_A.B5f-2.1JcM"
                }
            ],
            "retrieve_mode": "0",
            "request_method": "1",
            "output_format": "0",
            "ssl_cert_file": "",
            "ssl_key_file": "",
            "ssl_key_password": "",
            "verify_peer": "0",
            "verify_host": "0",
            "allow_traps": "0",
            "uuid": "",
            "lifetime_type": "0",
            "enabled_lifetime_type": "2",
            "enabled_lifetime": "0",
            "state": "0",
            "error": "",
            "parameters": []
        }
    ],
    "id": 1
}
```


Récupérer une règle LLD avec des remplacements

Récupérez une règle LLD qui possède divers paramètres de remplacement.

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryrule.get",
    "params": {
        "output": ["name"],
        "itemids": "30980",
        "selectOverrides": ["name", "step", "stop", "filter", "operations"]
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "name": "Discover database host",
            "overrides": [
                {
                    "name": "Discover MySQL host",
                    "step": "1",
                    "stop": "1",
                    "filter": {
                        "evaltype": "2",
                        "formula": "",
                        "conditions": [
                            {
                                "macro": "{#UNIT.NAME}",
                                "operator": "8",
                                "value": "^mysqld\\.service$",
                                "formulaid": "A"
                            },
                            {
                                "macro": "{#UNIT.NAME}",
                                "operator": "8",
                                "value": "^mariadb\\.service$",
                                "formulaid": "B"
                            }
                        ],
                        "eval_formula": "A or B"
                    },
                    "operations": [
                        {
                            "operationobject": "3",
                            "operator": "2",
                            "value": "Database host",
                            "opstatus": {
                                "status": "0"
                            },
                            "optag": [
                                {
                                    "tag": "database",
                                    "value": "mysql"
                                }
                            ],
                            "optemplate": [
                                {
                                    "templateid": "10170"
                                }
                            ]
                        }
                    ]
```

```
                },
                {
                    "name": "Discover PostgreSQL host",
                    "step": "2",
                    "stop": "1",
                    "filter": {
                        "evaltype": "0",
                        "formula": "",
                        "conditions": [
                            {
                                "macro": "{#UNIT.NAME}",
                                "operator": "8",
                                "value": "^postgresql\\.service$",
                                "formulaid": "A"
                            }
                        ],
                        "eval_formula": "A"
                    },
                    "operations": [
                        {
                            "operationobject": "3",
                            "operator": "2",
                            "value": "Database host",
                            "opstatus": {
                                "status": "0"
                            },
                            "optag": [
                                {
                                    "tag": "database",
                                    "value": "postgresql"
                                }
                            ],
                            "optemplate": [
                                {
                                    "templateid": "10263"
                                }
                            ]
                        }
                    ]
                }
            ]
        }
    ],
    "id": 1
}
```


Voir aussi

- Prototype de graphique
- Hôte
- Prototype d'élément
- Filtre de règle LLD
- Prototype de déclencheur

Source

CDiscoveryRule::get() dans *ui/include/classes/api/services/CDiscoveryRule.php*.

discoveryrule.update

Description

`object discoveryrule.update(object/array lldRules)`

Cette méthode vous permet de mettre à jour des règles LLD existantes.

Note:
Cette méthode est disponible uniquement pour les types d'utilisateur *Admin* et *Super admin*. Les autorisations d'appel de cette méthode peuvent être révoquées dans les paramètres des rôles utilisateur. Consultez Rôles utilisateur pour plus d'informations.

Paramètres

`(object/array)` Propriétés de règle LLD à mettre à jour.

La propriété `itemid` doit être définie pour chaque règle LLD, toutes les autres propriétés sont facultatives. Seules les propriétés transmises seront mises à jour, toutes les autres resteront inchangées.

En plus des propriétés de règle LLD standard, la méthode accepte les paramètres suivants.

Paramètre	Type	Description
filter	object	Filtre de règle LLD pour remplacer le filtre existant.
preprocessing	object/array	Options de prétraitement de règle LLD pour remplacer les options de prétraitement existantes. Comportement des paramètres: - *lecture seule* pour les objets hérités
lld_macro_paths	object/array	Options lld_macro_path de règle LLD pour remplacer les options lld_macro_path existantes. Comportement des paramètres: - *lecture seule* pour les objets hérités
overrides	object/array	Options de surcharge de règle LLD pour remplacer les options de surcharge existantes. Comportement des paramètres: - *lecture seule* pour les objets hérités

Valeurs retournées

`(object)` Retourne un objet contenant les identifiants des règles mises à jour sous la propriété `itemids`.

Exemples

Ajout d'un filtre à une règle LLD

Ajoutez un filtre afin que le contenu de la macro *{#FSTYPE}* corresponde à l'expression régulière *@File systems for discovery*.

Request :

```
{
    "jsonrpc": "2.0",
    "method": "discoveryrule.update",
    "params": {
        "itemid": "22450",
        "filter": {
            "evaltype": 1,
            "conditions": [
                {
                    "macro": "{#FSTYPE}",
                    "value": "@File systems for discovery"
                }
            ]
        }
    },
    "id": 1
}
```

Response :

```
{
    "jsonrpc": "2.0",
```

```
    "result": {
        "itemids": [
            "22450"
        ]
    },
    "id": 1
}
```

Ajout de chemins de macro LLD

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryrule.update",
    "params": {
        "itemid": "22450",
        "lld_macro_paths": [
            {
                "lld_macro": "{#MACRO1}",
                "path": "$.json.path"
            }
        ]
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "22450"
        ]
    },
    "id": 1
}
```

Désactiver le trapping

Désactivez le trapping LLD pour la règle de découverte.

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryrule.update",
    "params": {
        "itemid": "28336",
        "allow_traps": 0
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "28336"
        ]
    },
    "id": 1
}
```

Mise à jour des options de prétraitement d’une règle LLD

Mettez à jour une règle LLD avec la règle de prétraitement « JSONPath ».

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryrule.update",
    "params": {
        "itemid": "44211",
        "preprocessing": [
            {
                "type": 12,
                "params": "$.path.to.json",
                "error_handler": 2,
                "error_handler_params": "5"
            }
        ]
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "44211"
        ]
    },
    "id": 1
}
```

Mise à jour du script de la règle LLD

Mettez à jour un script de règle LLD avec un script différent et supprimez les paramètres inutiles qui étaient utilisés par le script précédent.

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryrule.update",
    "params": {
        "itemid": "23865",
        "parameters": [],
        "script": "Zabbix.log(3, 'Log test');\nreturn 1;"
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "23865"
        ]
    },
    "id": 1
}
```

Mise à jour de la durée de vie d'une règle LLD

Mettez à jour la règle LLD pour désactiver l'entité qui n'est plus découverte après 12 heures et la supprimer après 7 jours.

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryrule.update",
    "params": {
        "itemid": "46864",
        "lifetime_type": 0,
        "lifetime": "7d",
        "enabled_lifetime_type": 0,
        "enabled_lifetime": "12h"
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "46864"
        ]
    },
    "id": 1
}
```

Source

CDiscoveryRule::update() dans *ui/include/classes/api/services/CDiscoveryRule.php*.

Répertoire d'utilisateurs

Cette classe est conçue pour fonctionner avec les répertoires d'utilisateurs.

Références d'objet :

- Répertoire d'utilisateurs
  - Mappages des types de média
  - Mappages des groupes de provisionnement

Méthodes disponibles :

- userdirectory.create - créer un nouveau répertoire d'utilisateurs
- userdirectory.delete - supprimer un répertoire d'utilisateurs
- userdirectory.get - récupérer un répertoire d'utilisateurs
- userdirectory.update - mettre à jour un répertoire d'utilisateurs
- userdirectory.test - tester la connexion au répertoire d'utilisateurs

Objet user directory

Les objets suivants sont directement liés à l'API `userdirectory`.

Répertoire d'utilisateurs

L'objet de répertoire d'utilisateurs possède les propriétés suivantes.

Propriété	Type	Description
userdirectoryid	ID	ID du répertoire d'utilisateurs. Si un répertoire d'utilisateurs est supprimé, la valeur de la propriété `userdirectoryid` de l'objet User est définie sur « 0 » pour tous les utilisateurs liés au répertoire d'utilisateurs supprimé. Comportement de la propriété : - *en lecture seule* - *obligatoire* pour les opérations de mise à jour

Propriété	Type	Description
idp_type	integer	Type de protocole d'authentification utilisé par le fournisseur d'identité pour le répertoire d'utilisateurs. Notez qu'un seul répertoire d'utilisateurs de type SAML peut exister. Valeurs possibles : 1 - Répertoire d'utilisateurs de type LDAP ; 2 - Répertoire d'utilisateurs de type SAML. Comportement de la propriété : - *obligatoire* pour les opérations de création
group_name	string	Attribut du répertoire d'utilisateurs LDAP/SAML contenant le nom du groupe utilisé pour mapper les groupes entre le répertoire d'utilisateurs LDAP/SAML et Zabbix. Exemple : *cn* Comportement de la propriété : - *obligatoire* si `provision_status` est défini sur « Enabled » et si `saml_jit_status` de l'objet Authentication est défini sur « Enabled for configured SAML IdPs »
user_username	string	Attribut du répertoire d'utilisateurs LDAP/SAML (également attribut SCIM si `scim_status` est défini sur « SCIM provisioning is enabled ») contenant le nom de l'utilisateur, utilisé comme valeur de la propriété `name` de l'objet User lors du provisionnement de l'utilisateur. Exemples : *cn*, *commonName*, *displayName*, *name*
user_lastname	string	Attribut du répertoire d'utilisateurs LDAP/SAML (également attribut SCIM si `scim_status` est défini sur « SCIM provisioning is enabled ») contenant le nom de famille de l'utilisateur, utilisé comme valeur de la propriété `surname` de l'objet User lors du provisionnement de l'utilisateur. Exemples : *sn*, *surname*, *lastName*
provision_status	integer	État du provisionnement du répertoire d'utilisateurs. Valeurs possibles : 0 - *(par défaut)* Désactivé (le provisionnement des utilisateurs créés par ce répertoire d'utilisateurs est désactivé) ; 1 - Activé (le provisionnement des utilisateurs créés par ce répertoire d'utilisateurs est activé ; en outre, l'état du provisionnement LDAP ou SAML (`ldap_jit_status` ou `saml_jit_status` de l'objet Authentication) doit être activé).
provision_groups	array	Tableau d'objets de mappages des groupes de provisionnement permettant de mapper un modèle de groupe d'utilisateurs LDAP/SAML à un groupe d'utilisateurs et à un rôle d'utilisateur Zabbix. Comportement de la propriété : - *obligatoire* si `provision_status` est défini sur « Enabled »
provision_media	array	Tableau d'objets de mappages des types de média permettant de mapper les attributs de média LDAP/SAML de l'utilisateur (par exemple, l'adresse e-mail) aux médias utilisateur Zabbix pour l'envoi de notifications.
Propriétés spécifiques à LDAP :		
name	string	Nom unique du répertoire d'utilisateurs. Comportement de la propriété : - *obligatoire* si `idp_type` est défini sur « User directory of type LDAP »

Propriété	Type	Description
host	string	Nom d'hôte, adresse IP ou URI du serveur LDAP. L'URI doit contenir le schéma (`ldap://` ou `ldaps://`), l'hôte et éventuellement le port. Exemples : *host.example.com* *127.0.0.1* *ldap://ldap.example.com:389* Comportement de la propriété : - *obligatoire* si `idp_type` est défini sur « User directory of type LDAP »
port	integer	Port du serveur LDAP. Comportement de la propriété : - *obligatoire* si `idp_type` est défini sur « User directory of type LDAP »
base_dn	string	Chemin de base du répertoire d'utilisateurs LDAP vers les comptes utilisateur. Exemples : *ou=Users,dc=example,dc=org* *ou=Users,ou=system* (pour OpenLDAP) *DC=company,DC=com* (pour Microsoft Active Directory) *uid=%{user},dc=example,dc=com* (pour la liaison directe de l'utilisateur ; l'espace réservé « *%{user}* » est obligatoire) Comportement de la propriété : - *obligatoire* si `idp_type` est défini sur « User directory of type LDAP »
search_attribute	string	Attribut du répertoire d'utilisateurs LDAP permettant d'identifier le compte utilisateur à partir des informations fournies dans la demande de connexion. Exemples : *uid* (pour OpenLDAP) *sAMAccountName* (pour Microsoft Active Directory) Comportement de la propriété : - *obligatoire* si `idp_type` est défini sur « User directory of type LDAP »
bind_dn	string	Compte du serveur LDAP utilisé pour établir la liaison et effectuer des recherches sur le serveur LDAP. Pour la liaison directe de l'utilisateur et la liaison anonyme, `bind_dn` doit être vide. Exemples : *uid=ldap_search,ou=system* (pour OpenLDAP) *CN=ldap_search,OU=user_group,DC=company,DC=com* (pour Microsoft Active Directory) *CN=Admin,OU=Users,OU=Zabbix,DC=zbx,DC=local* Comportement de la propriété : - *pris en charge* si `idp_type` est défini sur « User directory of type LDAP »
bind_password	string	Mot de passe LDAP du compte utilisé pour établir la liaison et effectuer des recherches sur le serveur LDAP. Pour la liaison directe de l'utilisateur et la liaison anonyme, `bind_password` doit être vide. Comportement de la propriété : - *pris en charge* si `idp_type` est défini sur « User directory of type LDAP »

Propriété	Type	Description
description	string	Description du répertoire d'utilisateurs. Comportement de la propriété : - *pris en charge* si `idp_type` est défini sur « User directory of type LDAP »
group_basedn	string	Chemin de base du répertoire d'utilisateurs LDAP vers les groupes ; utilisé pour configurer une vérification de l'appartenance d'un utilisateur dans le répertoire d'utilisateurs LDAP. Ignoré lors du provisionnement d'un utilisateur si `group_membership` est défini. Exemple : *ou=Groups,dc=example,dc=com* Comportement de la propriété : - *pris en charge* si `idp_type` est défini sur « User directory of type LDAP »
group_filter	string	Chaîne de filtre permettant de récupérer les groupes du répertoire d'utilisateurs LDAP dont l'utilisateur est membre ; utilisée pour configurer une vérification de l'appartenance d'un utilisateur dans le répertoire d'utilisateurs LDAP. Ignorée lors du provisionnement d'un utilisateur si `group_membership` est défini. Espaces réservés `group_filter` pris en charge : *%{attr}* - attribut de recherche (remplacé par la valeur de la propriété `search_attribute`) ; *%{groupattr}* - attribut du groupe (remplacé par la valeur de la propriété `group_member`) ; *%{host}* - nom d'hôte, adresse IP ou URI du serveur LDAP (remplacé par la valeur de la propriété `host`) ; *%{user}* - nom d'utilisateur Zabbix. Valeur par défaut : *(%{groupattr}=%{user})* Exemples : - *(member=uid=%{ref},ou=Users,dc=example,dc=com)* correspondra à « User1 » si un objet de groupe LDAP contient l'attribut « *member* » avec la valeur « *uid=User1,ou=Users,dc=example,dc=com* », et renverra le groupe dont « User1 » est membre ; - *(%{groupattr}=cn=%{ref},ou=Users,ou=Zabbix,DC=example,DC=com)* correspondra à « User1 » si un objet de groupe LDAP contient l'attribut spécifié dans la propriété `group_member` avec la valeur « *cn=User1,ou=Users,ou=Zabbix,DC=example,DC=com* », et renverra le groupe dont « User1 » est membre. Comportement de la propriété : - *pris en charge* si `idp_type` est défini sur « User directory of type LDAP »
group_member	string	Attribut du répertoire d'utilisateurs LDAP contenant les informations sur les membres du groupe ; utilisé pour configurer une vérification de l'appartenance d'un utilisateur dans le répertoire d'utilisateurs LDAP. Ignoré lors du provisionnement d'un utilisateur si `group_membership` est défini. Comportement de la propriété : - *pris en charge* si `idp_type` est défini sur « User directory of type LDAP »

Propriété	Type	Description
group_membership	string	Attribut du répertoire d'utilisateurs LDAP contenant les informations sur les groupes auxquels appartient un utilisateur. Exemple : *memberOf* Comportement de la propriété : - *pris en charge* si `idp_type` est défini sur « User directory of type LDAP »
search_filter	string	Chaîne de filtre personnalisée utilisée pour localiser et authentifier un utilisateur dans un répertoire d'utilisateurs LDAP à partir des informations fournies dans la demande de connexion. Espaces réservés `search_filter` pris en charge : *%{attr}* - nom de l'attribut de recherche (par exemple, *uid*, *sAMAccountName*) ; *%{user}* - nom d'utilisateur Zabbix. Valeur par défaut : *(%{attr}=%{user})* Comportement de la propriété : - *pris en charge* si `idp_type` est défini sur « User directory of type LDAP »
start_tls	integer	Option de configuration du serveur LDAP permettant de sécuriser la communication avec le serveur LDAP à l'aide de Transport Layer Security (TLS). Notez que `start_tls` doit être défini sur « Disabled » pour les hôtes utilisant le protocole `ldaps://`. Valeurs possibles : 0 - *(par défaut)* Désactivé ; 1 - Activé. Comportement de la propriété : - *pris en charge* si `idp_type` est défini sur « User directory of type LDAP »
user_ref_attr	string	Attribut du répertoire d'utilisateurs LDAP utilisé pour référencer un objet utilisateur. La valeur de `user_ref_attr` est utilisée pour obtenir les valeurs de l'attribut spécifié dans le répertoire d'utilisateurs et les placer à la place de l'espace réservé *%{ref}* dans la chaîne `group_filter`. Exemples : *cn*, *uid*, *member*, *uniqueMember* Comportement de la propriété : - *pris en charge* si `idp_type` est défini sur « User directory of type LDAP »
Propriétés spécifiques à SAML :		
idp_entityid	string	URI identifiant le fournisseur d'identité et utilisé pour communiquer avec celui-ci dans les messages SAML. Exemple : *https://idp.example.com/idp* Comportement de la propriété : - *obligatoire* si `idp_type` est défini sur « User directory of type SAML »

Propriété	Type	Description
sp_entityid	string	URL ou chaîne identifiant le fournisseur de services du fournisseur d'identité. Exemples : *https://idp.example.com/sp* *zabbix* Comportement de la propriété : - *obligatoire* si `idp_type` est défini sur « User directory of type SAML »
username_attribute	string	Attribut du répertoire d'utilisateurs SAML (également attribut SCIM si `scim_status` est défini sur « SCIM provisioning is enabled ») contenant le nom d'utilisateur, comparé à la valeur de la propriété `username` de l'objet User lors de l'authentification. Exemples : *uid*, *userprincipalname*, *samaccountname*, *username*, *userusername*, *urn:oid:0.9.2342.19200300.100.1.1*, *urn:oid:1.3.6.1.4.1.5923.1.1.1.13*, *urn:oid:0.9.2342.19200300.100.1.44*
sso_url	string	Comportement de la propriété : - *obligatoire* si `idp_type` est défini sur « User directory of type SAML » URL du service SAML d'authentification unique du fournisseur d'identité, vers lequel Zabbix envoie les demandes d'authentification SAML. Exemple : *http://idp.example.com/idp/sso/saml*
slo_url	string	Comportement de la propriété : - *obligatoire* si `idp_type` est défini sur « User directory of type SAML » URL du service SAML de déconnexion unique du fournisseur d'identité, vers lequel Zabbix envoie les demandes de déconnexion SAML. Exemple : *https://idp.example.com/idp/slo/saml*
encrypt_nameid	integer	Comportement de la propriété : - *pris en charge* si `idp_type` est défini sur « User directory of type SAML » Indique si l'ID de nom SAML doit être chiffré. Valeurs possibles : 0 - *(par défaut)* Ne pas chiffrer l'ID de nom ; 1 - Chiffrer l'ID de nom.
encrypt_assertions	integer	Comportement de la propriété : - *pris en charge* si `idp_type` est défini sur « User directory of type SAML » Indique si les assertions SAML doivent être chiffrées. Valeurs possibles : 0 - *(par défaut)* Ne pas chiffrer les assertions ; 1 - Chiffrer les assertions. Comportement de la propriété : - *pris en charge* si `idp_type` est défini sur « User directory of type SAML »

Propriété	Type	Description
nameid_format	string	Format de l'ID de nom du fournisseur de services du fournisseur d'identité SAML. Exemples : *urn:oasis:names:tc:SAML:2.0:nameid-format:persistent* *urn:oasis:names:tc:SAML:2.0:nameid-format:transient* *urn:oasis:names:tc:SAML:2.0:nameid-format:kerberos* *urn:oasis:names:tc:SAML:2.0:nameid-format:entity* Comportement de la propriété : - *pris en charge* si `idp_type` est défini sur « User directory of type SAML »
scim_status	integer	Indique si le provisionnement SCIM pour SAML est activé ou désactivé. Valeurs possibles : 0 - *(par défaut)* Le provisionnement SCIM est désactivé ; 1 - Le provisionnement SCIM est activé. Comportement de la propriété : - *pris en charge* si `idp_type` est défini sur « User directory of type SAML »
sign_assertions	integer	Indique si les assertions SAML doivent être signées avec une signature SAML. Valeurs possibles : 0 - Ne pas signer les assertions ; 1 - *(par défaut)* Signer les assertions. Comportement de la propriété : - *pris en charge* si `idp_type` est défini sur « User directory of type SAML »
sign_authn_requests	integer	Indique si les demandes AuthN SAML doivent être signées avec une signature SAML. Valeurs possibles : 0 - Ne pas signer les demandes AuthN ; 1 - *(par défaut)* Signer les demandes AuthN. Comportement de la propriété : - *pris en charge* si `idp_type` est défini sur « User directory of type SAML »
sign_messages	integer	Indique si les messages SAML doivent être signés avec une signature SAML. Valeurs possibles : 0 - Ne pas signer les messages ; 1 - *(par défaut)* Signer les messages. Comportement de la propriété : - *pris en charge* si `idp_type` est défini sur « User directory of type SAML »
sign_logout_requests	integer	Indique si les demandes de déconnexion SAML doivent être signées avec une signature SAML. Valeurs possibles : 0 - Ne pas signer les demandes de déconnexion ; 1 - *(par défaut)* Signer les demandes de déconnexion. Comportement de la propriété : - *pris en charge* si `idp_type` est défini sur « User directory of type SAML »

Propriété	Type	Description
sign_logout_responses	integer	Indique si les réponses de déconnexion SAML doivent être signées avec une signature SAML. Valeurs possibles : 0 - Ne pas signer les réponses de déconnexion ; 1 - *(par défaut)* Signer les réponses de déconnexion. Comportement de la propriété : - *pris en charge* si `idp_type` est défini sur « User directory of type SAML »
idp_certificate	string	Contenu du certificat du fournisseur de services (SP) pour la configuration du service SAML d'authentification unique (SSO). Exemples : *-----BEGIN CERTIFICATE----- ...more encoded data... -----END CERTIFICATE-----* Comportement de la propriété : - *pris en charge* si `idp_type` est défini sur « User directory of type SAML » et si `$SSO['CERT_STORAGE']` est défini sur `database` dans le fichier de configuration de l'interface - *en écriture seule*
sp_private_key	string	Contenu de la clé privée du fournisseur de services (SP) pour la configuration du service SAML d'authentification unique (SSO). Elle fournit une authentification sécurisée et un échange de données avec le fournisseur d'identité (IdP). Exemples : *-----BEGIN CERTIFICATE----- ...more encoded data... -----END CERTIFICATE-----* Comportement de la propriété : - *pris en charge* si `idp_type` est défini sur « User directory of type SAML » et si `$SSO['CERT_STORAGE']` est défini sur `database` dans le fichier de configuration de l'interface - *en écriture seule*
sp_certificate	string	Contenu du certificat du fournisseur de services (SP) pour la configuration du service SAML d'authentification unique (SSO). Exemples : *-----BEGIN CERTIFICATE----- ...more encoded data... -----END CERTIFICATE-----* Comportement de la propriété : - *pris en charge* si `idp_type` est défini sur « User directory of type SAML » et si `$SSO['CERT_STORAGE']` est défini sur `database` dans le fichier de configuration de l'interface - *en écriture seule*
idp_certificate_hash	string	Hachage md5 de la valeur de idp_certificate. Renvoie une chaîne vide si idp_certificate est vide. Comportement de la propriété : - *pris en charge* si `idp_type` est défini sur « User directory of type SAML » et si `$SSO['CERT_STORAGE']` est défini sur `database` dans le fichier de configuration de l'interface - *en lecture seule*

Propriété	Type	Description
sp_private_key_hash	string	Hachage md5 de la valeur de sp_private_key. Renvoie une chaîne vide si sp_private_key est vide. Comportement de la propriété : - *pris en charge* si `idp_type` est défini sur « User directory of type SAML » et si `$SSO['CERT_STORAGE']` est défini sur `database` dans le fichier de configuration de l'interface - *en lecture seule*
sp_certificate_hash	string	Hachage md5 de la valeur de sp_certificate. Renvoie une chaîne vide si sp_certificate est vide. Comportement de la propriété : - *pris en charge* si `idp_type` est défini sur « User directory of type SAML » et si `$SSO['CERT_STORAGE']` est défini sur `database` dans le fichier de configuration de l'interface - *en lecture seule*

Mappages des types de média

L'objet de mappages des types de média possède les propriétés suivantes.

Propriété	Type	Description
userdirectory_mediaid	ID	ID du mappage du type de média. Comportement de la propriété: - *lecture seule*
name	string	Nom visible dans la liste des mappages des types de média. Comportement de la propriété: - *obligatoire*
mediatypeid	ID	ID du type de média à créer ; utilisée comme valeur de la propriété `mediatypeid` de l'objet Media. Comportement de la propriété: - *obligatoire*
attribute	string	Attribut du répertoire d'utilisateurs LDAP/SAML (également attribut SCIM si `scim_status` est défini sur "SCIM provisioning is enabled") qui contient le média de l'utilisateur (par exemple, *user@example.com*), utilisé comme valeur de la propriété `sendto` de l'objet Media. S'il est présent dans les données reçues du fournisseur d'identité LDAP/SAML et que sa valeur n'est pas vide, cela déclenchera la création du média pour l'utilisateur provisionné. Comportement de la propriété: - *obligatoire*
active	integer	Valeur de la propriété `active` du média utilisateur lorsque le média est créé pour l'utilisateur provisionné. Valeurs possibles : 0 - *(par défaut)* activé ; 1 - désactivé.
severity	integer	Valeur de la propriété `severity` du média utilisateur lorsque le média est créé pour l'utilisateur provisionné. Par défaut : 63.
period	string	Valeur de la propriété `period` du média utilisateur lorsque le média est créé pour l'utilisateur provisionné. Par défaut : 1-7,00:00-24:00.

Mappages des groupes de provisionnement

Les mappages des groupes de provisionnement possèdent les propriétés suivantes.

Propriété	Type	Description
name	string	Nom complet d'un groupe (par exemple, *Zabbix administrators*) dans l'annuaire d'utilisateurs LDAP/SAML (également SCIM si `scim_status` est défini sur "SCIM provisioning is enabled"). Prend en charge le caractère générique "*". Unique parmi tous les mappages des groupes de provisionnement. Comportement de la propriété: - *obligatoire*
roleid	ID	ID du rôle utilisateur à attribuer à l'utilisateur. Si plusieurs mappages des groupes de provisionnement correspondent, le rôle du type d'utilisateur le plus élevé (*User*, *Admin* ou *Super admin*) est attribué à l'utilisateur. S'il existe plusieurs rôles avec le même type d'utilisateur, le premier rôle (trié par ordre alphabétique) est attribué à l'utilisateur. Comportement de la propriété: - *obligatoire*
user_groups	array	Tableau d'objets d'ID de groupes d'utilisateurs Zabbix. Chaque objet possède les propriétés suivantes : `usrgrpid` - `(ID)` ID du groupe d'utilisateurs Zabbix à attribuer à l'utilisateur. Si plusieurs mappages des groupes de provisionnement correspondent, les groupes d'utilisateurs Zabbix de tous les mappages correspondants sont attribués à l'utilisateur. Comportement de la propriété: - *obligatoire*

userdirectory.create

Description

`object userdirectory.create(object/array userDirectory)`

Cette méthode permet de créer de nouveaux répertoires d'utilisateurs.

> **Note:**
> Cette méthode est disponible uniquement pour le type d'utilisateur *Super admin*.

Paramètres

`(object/array)` Répertoires utilisateur à créer.

La méthode accepte des répertoires utilisateur avec les propriétés standard des répertoires utilisateur.

Valeurs de retour

`(object)` Renvoie un objet contenant les ID des répertoires utilisateurs créés dans la propriété `userdirectoryids`. L'ordre des ID renvoyés correspond à l'ordre des répertoires utilisateurs transmis.

Exemples

Création d'un répertoire d'utilisateurs

Créez un répertoire d'utilisateurs pour authentifier les utilisateurs avec StartTLS via LDAP. Notez que pour authentifier les utilisateurs via LDAP, l'authentification LDAP doit être activée.

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "userdirectory.create",
    "params": {
        "idp_type": "1",
        "name": "LDAP API server #1",
```

```
        "host": "ldap://local.ldap",
        "port": "389",
        "base_dn": "ou=Users,dc=example,dc=org",
        "bind_dn": "cn=ldap_search,dc=example,dc=org",
        "bind_password": "ldapsecretpassword",
        "search_attribute": "uid",
        "start_tls": "1"
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "userdirectoryids": [
            "3"
        ]
    },
    "id": 1
}
```

Création d'un annuaire d'utilisateurs (provisionnement JIT activé)

Créez un annuaire d'utilisateurs pour authentifier les utilisateurs via LDAP (avec le provisionnement JIT activé). Notez que pour authentifier les utilisateurs via LDAP, l'authentification LDAP doit être activée.

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "userdirectory.create",
    "params": {
            "idp_type": "1",
            "name": "AD server",
            "provision_status": "1",
            "description": "",
            "host": "host.example.com",
            "port": "389",
            "base_dn": "DC=zbx,DC=local",
            "search_attribute": "sAMAccountName",
            "bind_dn": "CN=Admin,OU=Users,OU=Zabbix,DC=zbx,DC=local",
            "start_tls": "0",
            "search_filter": "",
            "group_basedn": "OU=Zabbix,DC=zbx,DC=local",
            "group_name": "CN",
            "group_member": "member",
            "group_filter": "(%{groupattr}=CN=%{ref},OU=Users,OU=Zabbix,DC=zbx,DC=local)",
            "group_membership": "",
            "user_username": "givenName",
            "user_lastname": "sn",
            "user_ref_attr": "CN",
            "provision_media": [
                {
                    "name": "example.com",
                    "mediatypeid": "1",
                    "attribute": "user@example.com"
                }
            ],
            "provision_groups": [
                {
                    "name": "*",
                    "roleid": "4",
                    "user_groups": [
```

```
                        {
                            "usrgrpid": "8"
                        }
                    ]
                },
                {
                    "name": "Zabbix administrators",
                    "roleid": "2",
                    "user_groups": [
                        {
                            "usrgrpid": "7"
                        },
                        {
                            "usrgrpid": "8"
                        }
                    ]
                }
            ]
        },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "userdirectoryids": [
            "2"
        ]
    },
    "id": 1
}
```

Source

CUserDirectory::create() dans *ui/include/classes/api/services/CUserDirectory.php*.

userdirectory.delete

Description

`object userdirectory.delete(array userDirectoryIds)`

Cette méthode vous permet de supprimer des répertoires d'utilisateurs. Un répertoire d'utilisateurs ne peut pas être supprimé lorsqu'il est utilisé directement par au moins un groupe d'utilisateurs.
 Le répertoire d'utilisateurs LDAP par défaut ne peut pas être supprimé lorsque `authentication.ldap_configured` est défini sur 1 ou lorsqu'il reste d'autres répertoires d'utilisateurs.

> **Note:**
> Cette méthode est disponible uniquement pour le type d'utilisateur *Super admin*.

Paramètres

`(array)` ID des répertoires d'utilisateurs à supprimer.

Valeurs de retour

`(object)` Renvoie un objet contenant les ID des répertoires utilisateurs supprimés dans la propriété `userdirectoryids`.

Exemples

Suppression de plusieurs annuaires d'utilisateurs

Supprimez deux annuaires d'utilisateurs.

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "userdirectory.delete",
    "params": [
        "2",
        "12"
    ],
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "userdirectoryids": [
            "2",
            "12"
        ]
    },
    "id": 1
}
```

Source

CUserDirectory::delete() dans *ui/include/classes/api/services/CUserDirectory.php*.

userdirectory.get

Description

`integer/array userdirectory.get(object parameters)`

Cette méthode permet de récupérer les répertoires d'utilisateurs selon les paramètres fournis.

Note:
Cette méthode est disponible uniquement pour les types d'utilisateur *Super admin*.

Paramètres

`(object)` Paramètres définissant la sortie souhaitée.

La méthode prend en charge les paramètres suivants.

Paramètre	Type	Description
userdirectoryids	ID/array	Retourner uniquement les répertoires d'utilisateurs avec les ID donnés.
selectUsrgrps	query	Retourner une propriété `usrgrps` avec les groupes d'utilisateurs associés à un répertoire d'utilisateurs. Prend en charge `count`.
selectProvisionMedia	query	Retourner une propriété `provision_media` avec les correspondances de types de média associées à un répertoire d'utilisateurs.
selectProvisionGroups	query	Retourner une propriété `provision_groups` avec les correspondances des groupes de provisionnement associées à un répertoire d'utilisateurs.
sortfield	string/array	Trier le résultat selon les propriétés données. Valeurs possibles : `name`.

Paramètre	Type	Description
filter	object	Retourner uniquement les résultats qui correspondent exactement au filtre donné. Accepte un objet où les clés sont des noms de propriété et les valeurs sont soit une valeur unique, soit un tableau de valeurs. Prend en charge les propriétés suivantes : `userdirectoryid`, `idp_type`, `provision_status`.
search	object	Retourner les résultats correspondant au motif donné (insensible à la casse). Accepte un objet où les clés sont des noms de propriété et les valeurs sont des chaînes à rechercher. Si aucune option supplémentaire n'est fournie, cela effectuera une recherche `LIKE "%…%"`. Propriétés prises en charge : `name`, `description`. Un répertoire d'utilisateurs de type SAML aura une valeur vide pour les propriétés `name` et `description`. Les deux propriétés peuvent être modifiées avec l'opération `userdirectory.update`.
countOutput	boolean	Ces paramètres sont décrits dans le commentaire de référence.
excludeSearch	boolean	
limit	integer	
output	query	
preservekeys	boolean	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Valeurs de retour

`(integer/array)` Renvoie soit :

- un tableau d'objets
- le nombre d'objets récupérés, si le paramètre `countOutput` a été utilisé

Exemples

Récupération des annuaires d'utilisateurs

Récupérez tous les annuaires d'utilisateurs avec des propriétés supplémentaires affichant les correspondances des types de média et les correspondances des groupes de provisionnement associées à chaque annuaire d'utilisateurs.

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "userdirectory.get",
    "params": {
        "output": "extend",
        "selectProvisionMedia": "extend",
        "selectProvisionGroups": "extend"
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": [
       {
            "userdirectoryid": "1",
            "idp_type": "2",
```

```
        "name": "",
        "provision_status": "1",
        "description": "",
        "group_name": "groups",
        "user_username": "",
        "user_lastname": "",
        "idp_entityid": "http://example.com/simplesaml/saml2/idp/metadata.php",
        "sso_url": "http://example.com/simplesaml/saml2/idp/SSOService.php",
        "slo_url": "",
        "username_attribute": "uid",
        "sp_entityid": "zabbix",
        "nameid_format": "",
        "sign_messages": "0",
        "sign_assertions": "0",
        "sign_authn_requests": "0",
        "sign_logout_requests": "0",
        "sign_logout_responses": "0",
        "encrypt_nameid": "0",
        "encrypt_assertions": "0",
        "scim_status": "1",
        "provision_media": [
            {
                "userdirectory_mediaid": "1",
                "name": "example.com",
                "mediatypeid": "1",
                "attribute": "user@example.com",
                "active": "0",
                "severity": "63",
                "period": "1-7,00:00-24:00"
            }
        ],
        "provision_groups": [
            {
                "name": "*",
                "roleid": "1",
                "user_groups": [
                    {
                        "usrgrpid": "13"
                    }
                ]
            }
        ]
    },
    {
        "userdirectoryid": "2",
        "idp_type": "1",
        "name": "AD server",
        "provision_status": "1",
        "description": "",
        "host": "host.example.com",
        "port": "389",
        "base_dn": "DC=zbx,DC=local",
        "search_attribute": "sAMAccountName",
        "bind_dn": "CN=Admin,OU=Users,OU=Zabbix,DC=zbx,DC=local",
        "start_tls": "0",
        "search_filter": "",
        "group_basedn": "OU=Zabbix,DC=zbx,DC=local",
        "group_name": "CN",
        "group_member": "member",
        "group_filter": "(%{groupattr}=CN=%{ref},OU=Users,OU=Zabbix,DC=zbx,DC=local)",
        "group_membership": "",
        "user_username": "givenName",
```

```
            "user_lastname": "sn",
            "user_ref_attr": "CN",
            "provision_media": [
                {
                    "userdirectory_mediaid": "2",
                    "name": "example.com",
                    "mediatypeid": "1",
                    "attribute": "user@example.com",
                    "active": "0",
                    "severity": "63",
                    "period": "1-7,00:00-24:00"
                }
            ],
            "provision_groups": [
                {
                    "name": "*",
                    "roleid": "4",
                    "user_groups": [
                        {
                            "usrgrpid": "8"
                        }
                    ]
                },
                {
                    "name": "Zabbix administrators",
                    "roleid": "2",
                    "user_groups": [
                        {
                            "usrgrpid": "7"
                        },
                        {
                            "usrgrpid": "8"
                        }
                    ]
                }
            ]
        },
        {
            "userdirectoryid": "3",
            "idp_type": "1",
            "name": "LDAP API server #1",
            "provision_status": "0",
            "description": "",
            "host": "ldap://local.ldap",
            "port": "389",
            "base_dn": "ou=Users,dc=example,dc=org",
            "search_attribute": "uid",
            "bind_dn": "cn=ldap_search,dc=example,dc=org",
            "start_tls": "1",
            "search_filter": "",
            "group_basedn": "",
            "group_name": "",
            "group_member": "",
            "group_filter": "",
            "group_membership": "",
            "user_username": "",
            "user_lastname": "",
            "user_ref_attr": "",
            "provision_media": [],
            "provision_groups": []
        }
    ],
```

```
    "id": 1
}
```

Voir aussi

- Groupe d'utilisateurs

Source

CUserDirectory::get() dans *ui/include/classes/api/services/CUserDirectory.php*.

userdirectory.test

Description

`object userdirectory.test(array userDirectory)`

Cette méthode vous permet de tester les paramètres de connexion du répertoire d'utilisateurs.

Note:
Cette méthode permet également de tester quelles données configurées correspondent aux paramètres du répertoire d'utilisateurs pour le provisionnement des utilisateurs (par exemple, quel rôle utilisateur, quels groupes d'utilisateurs, quels médias utilisateur seront attribués à l'utilisateur). Pour ce type de test, la requête API doit être effectuée pour un répertoire d'utilisateurs dont `provision_status` est défini sur enabled.

Note:
Cette méthode est disponible uniquement pour le type d'utilisateur *Super admin*.

Paramètres

`(object)` Propriétés du répertoire d'utilisateurs.

Comme l'API `userdirectory.get` ne renvoie pas le champ `bind_password`, `userdirectoryid` et/ou `bind_password` doivent être fournis.
 En plus des propriétés standard du répertoire d'utilisateurs, la méthode accepte les paramètres suivants.

Paramètre	Type	Description
test_username	string	Nom d'utilisateur à tester dans le répertoire d'utilisateurs.
test_password	string	Mot de passe associé au nom d'utilisateur à tester dans le répertoire d'utilisateurs.

Valeurs de retour

`(bool)` Renvoie true en cas de succès.

Exemples

Tester le répertoire d'utilisateurs pour un utilisateur existant

Tester le répertoire d'utilisateurs « 3 » pour « user1 ».

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "userdirectory.test",
    "params": {
        "userdirectoryid": "3",
        "host": "127.0.0.1",
        "port": "389",
        "base_dn": "ou=Users,dc=example,dc=org",
        "search_attribute": "uid",
        "bind_dn": "cn=ldap_search,dc=example,dc=org",
        "bind_password": "password",
        "test_username": "user1",
        "test_password": "password"
    },
```

```
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": true,
    "id": 1
}
```

Tester le répertoire d'utilisateurs pour un utilisateur inexistant

Tester le répertoire d'utilisateurs « 3 » pour l'utilisateur inexistant « user2 ».

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "userdirectory.test",
    "params": {
        "userdirectoryid": "3",
        "host": "127.0.0.1",
        "port": "389",
        "base_dn": "ou=Users,dc=example,dc=org",
        "search_attribute": "uid",
        "bind_dn": "cn=ldap_search,dc=example,dc=org",
        "test_username": "user2",
        "test_password": "password"
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "error": {
        "code": -32500,
        "message": "Erreur d'application.",
        "data": "Nom d'utilisateur ou mot de passe incorrect, ou compte temporairement bloqué."
    },
    "id": 1
}
```

Tester l'annuaire utilisateur pour le provisionnement des utilisateurs

Tester l'annuaire utilisateur « 3 » afin de vérifier quelles données configurées correspondent aux paramètres de l'annuaire utilisateur pour le provisionnement de « user3 » (par exemple, quel rôle utilisateur, quels groupes d'utilisateurs et quels médias utilisateur seront attribués à l'utilisateur).

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "userdirectory.test",
    "params": {
        "userdirectoryid": "2",
        "host": "host.example.com",
        "port": "389",
        "base_dn": "DC=zbx,DC=local",
        "search_attribute": "sAMAccountName",
        "bind_dn": "CN=Admin,OU=Users,OU=Zabbix,DC=zbx,DC=local",
        "test_username": "user3",
        "test_password": "password"
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "username": "user3",
        "name": "John",
        "surname": "Doe",
        "medias": [],
        "usrgrps": [
            {
                "usrgrpid": "8"
            },
            {
                "usrgrpid": "7"
            }
        ],
        "roleid": "2",
        "userdirectoryid": "2"
    },
    "id": 1
}
```

Source

CUserDirectory::test() dans *ui/include/classes/api/services/CUserDirectory.php*.

userdirectory.update

Description

`object userdirectory.update(object/array userDirectory)`

Cette méthode vous permet de mettre à jour des répertoires d'utilisateurs existants.

> **Note:**
> Cette méthode est disponible uniquement pour le type d'utilisateur *Super admin*.

Paramètres

`(object/array)` Propriétés du répertoire d'utilisateurs à mettre à jour.

La propriété `userdirectoryid` doit être définie pour chaque répertoire d'utilisateurs, toutes les autres propriétés sont facultatives. Seules les propriétés transmises seront mises à jour, toutes les autres resteront inchangées.

Valeurs de retour

`(object)` Renvoie un objet contenant les identifiants des répertoires utilisateurs mis à jour dans la propriété `userdirectoryids`.

Exemples

Mettre à jour le mot de passe de liaison pour un annuaire d'utilisateurs

Définissez un nouveau mot de passe de liaison pour un annuaire d'utilisateurs.

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "userdirectory.update",
    "params": {
        "userdirectoryid": "3",
        "bind_password": "newldappassword"
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "userdirectoryids": [
            "3"
        ]
    },
    "id": 1
}
```

Mettre à jour les correspondances du répertoire d'utilisateurs

Mettez à jour les correspondances des groupes de provisionnement et des types de média pour le répertoire d'utilisateurs « 2 ».

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "userdirectory.update",
    "params": {
        "userdirectoryid": "2",
        "provision_media": [
                {
                    "userdirectory_mediaid": "2"
                }
            ],
            "provision_groups": [
                {
                    "name": "Zabbix administrators",
                    "roleid": "2",
                    "user_groups": [
                        {
                            "usrgrpid": "7"
                        },
                        {
                            "usrgrpid": "8"
                        },
                        {
                            "usrgrpid": "11"
                        }
                    ]
                }
            ]
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "userdirectoryids": [
            "2"
        ]
    },
    "id": 1
}
```

Source

CUserDirectory::update() dans *ui/include/classes/api/services/CUserDirectory.php*.

Rôle

Cette classe est conçue pour fonctionner avec les rôles utilisateur.

Références d'objet :

- Role
- Règles de rôle
  - Élément d'interface
  - Service
  - Tag de service
  - Module
  - Action

Méthodes disponibles :

- role.create - créer de nouveaux rôles utilisateur
- role.delete - supprimer des rôles utilisateur
- role.get - récupérer des rôles utilisateur
- role.update - mettre à jour des rôles utilisateur

Objet rôle

Les objets suivants sont directement liés à l'API `role`.

Rôle

L'objet rôle possède les propriétés suivantes :

Propriété	Type	Description
roleid	ID	ID du rôle. Comportement de la propriété : - *lecture seule* - *obligatoire* pour les opérations de mise à jour
name	string	Nom du rôle. Comportement de la propriété : - *obligatoire* pour les opérations de création
type	integer	Type d'utilisateur. Valeurs possibles : 1 - *(par défaut)* Utilisateur ; 2 - Admin ; 3 - Super admin. Comportement de la propriété : - *obligatoire* pour les opérations de création
readonly	integer	Indique si le rôle est en lecture seule. Valeurs possibles : 0 - *(par défaut)* Non ; 1 - Oui. Comportement de la propriété : - *lecture seule*

Règles de rôle

L'objet des règles de rôle possède les propriétés suivantes :

Propriété	Type	Description
ui	array	Tableau d'objets élément d'interface.

Propriété	Type	Description
ui.default_access	integer	Indique si l'accès aux nouveaux éléments d'interface est activé. Valeurs possibles : 0 - Désactivé ; 1 - *(par défaut)* Activé.
profile.redirect.enforce	integer	Indique si l'URL de redirection définie au niveau du rôle après l'authentification est imposée et ne peut pas être remplacée par l'utilisateur. Valeurs possibles : 0 - *(par défaut)* Désactivé ; 1 - Activé.
profile.redirect.url	string	URL utilisée pour rediriger l'utilisateur après l'authentification.
services.read.mode	integer	Accès en lecture seule aux services. Valeurs possibles : 0 - Accès en lecture seule aux services spécifiés par les propriétés `services.read.list` ou correspondant à la propriété `services.read.tag` ; 1 - *(par défaut)* Accès en lecture seule à tous les services.
services.read.list	array	Tableau d'objets Service. Un accès en lecture seule aux services spécifiés, y compris leurs services enfants, sera accordé au rôle utilisateur. L'accès en lecture seule ne remplacera pas l'accès en lecture-écriture aux services. Comportement de la propriété : - *prise en charge* si `services.read.mode` est défini sur « 0 ».
services.read.tag	object	Tableau d'objets étiquette de service. Un accès en lecture seule aux services correspondant à l'étiquette, y compris leurs services enfants, sera accordé au rôle utilisateur. L'accès en lecture seule ne remplacera pas l'accès en lecture-écriture aux services. Comportement de la propriété : - *prise en charge* si `services.read.mode` est défini sur « 0 ».
services.write.mode	integer	Accès en lecture-écriture aux services. Valeurs possibles : 0 - *(par défaut)* Accès en lecture-écriture aux services spécifiés par les propriétés `services.write.list` ou correspondant à la propriété `services.write.tag` ; 1 - Accès en lecture-écriture à tous les services.
services.write.list	array	Tableau d'objets Service. Un accès en lecture-écriture aux services spécifiés, y compris leurs services enfants, sera accordé au rôle utilisateur. L'accès en lecture-écriture remplacera l'accès en lecture seule aux services. Comportement de la propriété : - *prise en charge* si `services.write.mode` est défini sur « 0 ».
services.write.tag	object	Tableau d'objets étiquette de service. Un accès en lecture-écriture aux services correspondant à l'étiquette, y compris leurs services enfants, sera accordé au rôle utilisateur. L'accès en lecture-écriture remplacera l'accès en lecture seule aux services. Comportement de la propriété : - *prise en charge* si `services.write.mode` est défini sur « 0 ».
modules	array	Tableau d'objets module.

Propriété	Type	Description
modules.default_access	integer	Indique si l'accès aux nouveaux modules est activé. Valeurs possibles : 0 - Désactivé ; 1 - *(par défaut)* Activé.
api.access	integer	Indique si l'accès à l'API est activé. Valeurs possibles : 0 - *(par défaut)* Désactivé ; 1 - Activé.
api.mode	integer	Mode de traitement des méthodes API répertoriées dans la propriété `api`. Valeurs possibles : 0 - *(par défaut)* Liste de refus ; 1 - Liste d'autorisation.
api	array	Comportement de la propriété : - *prise en charge* si `api.access` est défini sur « 1 ». Tableau de méthodes API. Les caractères génériques sont pris en charge, par exemple : `*` (toute méthode), `dashboard.*`, `*.export`. Si aucune méthode n'est spécifiée, l'accès à toutes les méthodes API est refusé si `api.mode` est défini sur « 1 » (Liste d'autorisation), et autorisé si `api.mode` est défini sur « 0 » (Liste de refus). Comportement de la propriété : - *prise en charge* si `api.access` est défini sur « 1 ».
actions	array	Tableau d'objets action.
actions.default_access	integer	Indique si l'accès aux nouvelles actions est activé. Valeurs possibles : 0 - Désactivé ; 1 - *(par défaut)* Activé.
devices.access	integer	Indique si l'accès aux propres appareils mobiles de l'utilisateur pour les utiliser avec l'application mobile Zabbix est activé. Valeurs possibles : 0 - (*par défaut pour Utilisateur et Administrateur*) Désactivé ; 1 - (*par défaut pour le Super administrateur*) Activé. Comportement de la propriété : - *prise en charge* si `EnableMobileDevices` est activé
devices.actions	array	Tableau d'objets action d'appareil. Comportement de la propriété : - *prise en charge* si `devices.access` est défini sur « 1 »
devices.actions.default_access	integer	Indique si l'accès aux nouvelles actions d'appareil est activé. Valeurs possibles : 0 - (*par défaut pour Utilisateur et Administrateur*) Désactivé ; 1 - (*par défaut pour le Super administrateur*) Activé. Comportement de la propriété : - *prise en charge* si `EnableMobileDevices` est activé

Élément d'interface

L'objet élément d'interface possède les propriétés suivantes :

Property	Type	Description
name	string	Nom de l'élément d'interface. Valeurs possibles si `type` de l'objet Role est défini sur « User », « Admin » ou « Super admin » : `monitoring.dashboard` - *Tableaux de bord* ; `monitoring.problems` - *Surveillance* > *Problèmes* ; `monitoring.hosts` - *Surveillance* > *Hôtes* ; `monitoring.latest_data` - *Surveillance* > *Dernières données* ; `monitoring.maps` - *Surveillance* > *Cartes* ; `services.services` - *Services* > *Services* ; `services.sla_report` - *Services* > *Rapport SLA* ; `inventory.overview` - *Inventaire* > *Vue d'ensemble* ; `inventory.hosts` - *Inventaire* > *Hôtes* ; `reports.availability_report` - *Rapports* > *Rapport de disponibilité* ; `reports.top_triggers` - *Rapports* > *Top 100 des déclencheurs*. Valeurs possibles si `type` de l'objet Role est défini sur « Admin » ou « Super admin » : `monitoring.discovery` - *Surveillance* > *Découverte* ; `services.sla` - *Services* > *SLA* ; `reports.scheduled_reports` - *Rapports* > *Rapports planifiés* ; `reports.notifications` - *Rapports* > *Notifications* ; `configuration.template_groups` - *Collecte de données* > *Groupes de modèles* ; `configuration.host_groups` - *Collecte de données* > *Groupes d'hôtes* ; `configuration.templates` - *Collecte de données* > *Modèles* ; `configuration.hosts` - *Collecte de données* > *Hôtes* ; `configuration.maintenance` - *Collecte de données* > *Maintenance* ; `configuration.discovery` - *Collecte de données* > *Découverte* ; `configuration.trigger_actions` - *Alertes* > *Actions* > *Actions de déclencheur* ; `configuration.service_actions` - *Alertes* > *Actions* > *Actions de service* ; `configuration.discovery_actions` - *Alertes* > *Actions* > *Actions de découverte* ; `configuration.autoregistration_actions` - *Alertes* > *Actions* > *Actions d'enregistrement automatique* ; `configuration.internal_actions` - *Alertes* > *Actions* > *Actions internes*. Valeurs possibles si `type` de l'objet Role est défini sur « Super admin » : `reports.system_info` - *Rapports* > *Informations système* ; `reports.audit` - *Rapports* > *Journal d'audit* ; `reports.action_log` - *Rapports* > *Journal des actions* ; `configuration.event_correlation` - *Collecte de données* > *Corrélation des événements* ; `administration.media_types` - *Alertes* > *Types de média* ; `administration.scripts` - *Alertes* > *Scripts* ; `administration.user_groups` - *Utilisateurs* > *Groupes d'utilisateurs* ; `administration.user_roles` - *Utilisateurs* > *Rôles utilisateur* ; `administration.users` - *Utilisateurs* > *Utilisateurs* ; `administration.api_tokens` - *Utilisateurs* > *Jetons API* ; `administration.authentication` - *Utilisateurs* > *Authentification* ; `administration.linked_devices` - *Utilisateurs* > *Appareils* ; `administration.general` - *Administration* > *Général* ; `administration.audit_log` - *Administration* > *Journal d'audit* ; `administration.housekeeping` - *Administration* > *Maintenance* ; `administration.proxy_groups` - *Administration* > *Groupes de proxy* ; `administration.proxies` - *Administration* > *Proxys* ;

Property	Type	Description
status	integer	Indique si l'accès à l'élément d'interface est activé. Valeurs possibles : 0 - Désactivé ; 1 - *(par défaut)* Activé.

Service

Propriété	Type	Description
serviceid	ID	ID du service. Comportement de la propriété : - *requis*

Tag de service

Propriété	Type	Description
tag	string	Nom du tag. Si une chaîne vide est spécifiée, le tag de service ne sera pas utilisé pour la correspondance des services. Comportement de la propriété : - *obligatoire*
value	string	Valeur du tag. Si aucune valeur ou une chaîne vide est spécifiée, seul le nom du tag sera utilisé pour la correspondance des services.

Module

L'objet module possède les propriétés suivantes :

Propriété	Type	Description
moduleid	ID	ID du module. Comportement de la propriété : - *obligatoire*
status	integer	Indique si l'accès au module est activé. Valeurs possibles : 0 - Désactivé ; 1 - *(par défaut)* Activé.

Action

L'objet action possède les propriétés suivantes :

Property	Type	Description
name	string	Nom de l'action. Valeurs possibles si `type` de l'objet Role est défini sur « User », « Admin » ou « Super admin » : `edit_dashboards` - Créer et modifier des tableaux de bord ; `edit_maps` - Créer et modifier des cartes ; `add_problem_comments` - Ajouter des commentaires aux problèmes ; `change_severity` - Modifier la gravité des problèmes ; `acknowledge_problems` - Accuser réception des problèmes ; `suppress_problems` - Supprimer les problèmes ; `close_problems` - Fermer les problèmes ; `execute_scripts` - Exécuter des scripts ; `manage_api_tokens` - Gérer les jetons d'API ; `change_problem_ranking` - Modifier le classement du problème, de cause à symptôme et inversement ; `edit_own_media` - Autoriser les utilisateurs à créer et modifier leurs propres médias. Valeurs possibles si `type` de l'objet Role est défini sur « Admin » ou « Super admin » : `edit_maintenance` - Créer et modifier des maintenances ; `manage_scheduled_reports` - Gérer les rapports planifiés, `manage_sla` - Gérer les SLA, `select_server_for_monitoring` - Autoriser les utilisateurs à créer des hôtes surveillés par le serveur et par les règles de découverte réseau exécutées par le serveur. Valeurs possibles si `type` de l'objet Role est défini sur « User » ou « Admin » : `invoke_execute_now` - Autoriser les utilisateurs disposant uniquement d'autorisations de lecture sur un hôte à exécuter des vérifications d'éléments. Valeurs possibles si `type` de l'objet Role est défini sur « Super admin » : `edit_user_media` - Autoriser les utilisateurs à créer et modifier les médias d'autres utilisateurs. Comportement de la propriété : - *required*
status	integer	Indique si l'accès permettant d'effectuer l'action est activé. Valeurs possibles : 0 - Désactivé ; 1 - *(par défaut)* Activé.

Action sur les appareils

L'objet d'action sur les appareils possède les propriétés suivantes :

Property	Type	Description
name	string	Nom de l'action sur les appareils. Valeurs possibles si `type` de l'objet Role est défini sur « User », « Admin » ou « Super admin » : `manage_own` - Autoriser les utilisateurs à ajouter ou supprimer leurs propres appareils mobiles. Valeurs possibles si `type` de l'objet Role est défini sur « Super admin » : `manage_user` - Autoriser les utilisateurs à ajouter ou supprimer des appareils mobiles pour tous les utilisateurs. Comportement de la propriété : - *required*
status	integer	Indique si l'accès permettant d'effectuer l'action est activé. Valeurs possibles : 0 - Désactivé ; 1 - (*par défaut*) Activé.

role.create

Description

`object role.create(object/array roles)`

Cette méthode vous permet de créer de nouveaux rôles.

> **Note:**
> Cette méthode est disponible uniquement pour le type d'utilisateur *Super admin*. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres du rôle utilisateur. Consultez User roles pour plus d'informations.

Paramètres

`(object/array)` Rôles à créer.

En plus des propriétés de rôle standard, la méthode accepte les paramètres suivants.

Paramètre	Type	Description
rules	array	Règles de rôle à créer pour le rôle.

Valeurs retournées

`(object)` Retourne un objet contenant les identifiants des rôles créés sous la propriété `roleids`. L'ordre des ID retournés correspond à l'ordre des rôles transmis.

Exemples

Création d'un rôle

Créez un rôle de type « Utilisateur » avec un accès refusé à deux éléments de l'interface utilisateur.

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "role.create",
    "params": {
        "name": "Operator",
        "type": "1",
        "rules": {
            "ui": [
                {
                    "name": "monitoring.hosts",
```

```
                    "status": "0"
                },
                {
                    "name": "monitoring.maps",
                    "status": "0"
                }
            ]
        }
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "roleids": [
            "5"
        ]
    },
    "id": 1
}
```

Voir aussi

- Règles de rôle
- Élément d'interface
- Module
- Action

Source

CRole::create() dans *ui/include/classes/api/services/CRole.php*.

role.delete

Description

`object role.delete(array roleids)`

Cette méthode vous permet de supprimer des rôles.

> **Note:**
> Cette méthode est disponible uniquement pour le type d'utilisateur *Super admin*. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres du rôle utilisateur. Consultez User roles pour plus d'informations.

Paramètres

`(array)` ID des rôles à supprimer.

Valeurs de retour

`(object)` Renvoie un objet contenant les ID des rôles supprimés dans la propriété `roleids`.

Exemples

Suppression de plusieurs rôles utilisateur

Supprimez deux rôles utilisateur.

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "role.delete",
    "params": [
        "4",
        "5"
```

```
    ],
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "roleids": [
            "4",
            "5"
        ]
    },
    "id": 1
}
```

Source

CRole::delete() dans *ui/include/classes/api/services/CRole.php*.

role.get

Description

`integer/array role.get(object parameters)`

La méthode permet de récupérer les rôles selon les paramètres fournis.

> **Note:**
> Cette méthode est disponible pour les utilisateurs de tout type. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres du rôle utilisateur. Consultez Rôles utilisateur pour plus d'informations.

Paramètres

`(object)` Paramètres définissant le résultat souhaité.

La méthode prend en charge les paramètres suivants.

Paramètre	Type	Description
roleids	ID/array	Retourne uniquement les rôles correspondant aux ID indiqués.
current	boolean	Si défini sur `true`, retourne uniquement le rôle de l'utilisateur actuel.
selectRules	query	Retourne une propriété `rules` contenant les règles du rôle.
selectUsers	query	Retourne une propriété `users` contenant les utilisateurs auxquels le rôle est attribué. Voir `user.get` pour connaître les restrictions en fonction du type d'utilisateur.
sortfield	string/array	Trie le résultat selon les propriétés indiquées. Valeurs possibles : `roleid`, `name`.
countOutput	boolean	Ces paramètres sont décrits dans la documentation de référence.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Valeurs retournées

`(integer/array)` Retourne soit :

- un tableau d'objets ;
- le nombre d'objets récupérés, si le paramètre `countOutput` a été utilisé.

Exemples

Récupération des données du rôle

Récupérer les données du rôle « Super admin role » ainsi que ses règles d'accès.

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "role.get",
    "params": {
        "output": "extend",
        "selectRules": "extend",
        "roleids": "3"
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "roleid": "3",
            "name": "Super admin role",
            "type": "3",
            "readonly": "1",
            "rules": {
                "ui": [
                    {
                        "name": "monitoring.dashboard",
                        "status": "1"
                    },
                    {
                        "name": "monitoring.problems",
                        "status": "1"
                    },
                    {
                        "name": "monitoring.hosts",
                        "status": "1"
                    },
                    {
                        "name": "monitoring.latest_data",
                        "status": "1"
                    },
                    {
                        "name": "monitoring.maps",
                        "status": "1"
                    },
                    {
                        "name": "services.services",
                        "status": "1"
                    },
                    {
                        "name": "services.sla_report",
                        "status": "1"
                    },
                    {
```

```
            "name": "inventory.overview",
            "status": "1"
        },
        {
            "name": "inventory.hosts",
            "status": "1"
        },
        {
            "name": "reports.availability_report",
            "status": "1"
        },
        {
            "name": "reports.top_triggers",
            "status": "1"
        },
        {
            "name": "monitoring.discovery",
            "status": "1"
        },
        {
            "name": "services.sla",
            "status": "1"
        },
        {
            "name": "reports.scheduled_reports",
            "status": "1"
        },
        {
            "name": "reports.notifications",
            "status": "1"
        },
        {
            "name": "configuration.template_groups",
            "status": "1"
        },
        {
            "name": "configuration.host_groups",
            "status": "1"
        },
        {
            "name": "configuration.templates",
            "status": "1"
        },
        {
            "name": "configuration.hosts",
            "status": "1"
        },
        {
            "name": "configuration.maintenance",
            "status": "1"
        },
        {
            "name": "configuration.discovery",
            "status": "1"
        },
        {
            "name": "configuration.trigger_actions",
            "status": "1"
        },
        {
            "name": "configuration.service_actions",
            "status": "1"
```

```
            },
            {
                "name": "configuration.discovery_actions",
                "status": "1"
            },
            {
                "name": "configuration.autoregistration_actions",
                "status": "1"
            },
            {
                "name": "configuration.internal_actions",
                "status": "1"
            },
            {
                "name": "reports.system_info",
                "status": "1"
            },
            {
                "name": "reports.audit",
                "status": "1"
            },
            {
                "name": "reports.action_log",
                "status": "1"
            },
            {
                "name": "configuration.event_correlation",
                "status": "1"
            },
            {
                "name": "administration.media_types",
                "status": "1"
            },
            {
                "name": "administration.scripts",
                "status": "1"
            },
            {
                "name": "administration.user_groups",
                "status": "1"
            },
            {
                "name": "administration.user_roles",
                "status": "1"
            },
            {
                "name": "administration.users",
                "status": "1"
            },
            {
                "name": "administration.api_tokens",
                "status": "1"
            },
            {
                "name": "administration.authentication",
                "status": "1"
            },
            {
                "name": "administration.general",
                "status": "1"
            },
            {
```

```
                "name": "administration.audit_log",
                "status": "1"
            },
            {
                "name": "administration.housekeeping",
                "status": "1"
            },
            {
                "name": "administration.proxy_groups",
                "status": "1"
            },
            {
                "name": "administration.proxies",
                "status": "1"
            },
            {
                "name": "administration.macros",
                "status": "1"
            },
            {
                "name": "administration.queue",
                "status": "1"
            }
        ],
        "ui.default_access": "1",
        "profile.redirect.enforce": "0",
        "profile.redirect.url": "",
        "services.read.mode": "1",
        "services.read.list": [],
        "services.read.tag": {
            "tag": "",
            "value": ""
        },
        "services.write.mode": "1",
        "services.write.list": [],
        "services.write.tag": {
            "tag": "",
            "value": ""
        },
        "modules": [
            {
                "moduleid": 1,
                "status": "1"
            },
            {
                "moduleid": 2,
                "status": "1"
            },
            {
                "moduleid": 3,
                "status": "1"
            },
            {
                "moduleid": 4,
                "status": "1"
            },
            {
                "moduleid": 5,
                "status": "1"
            },
            {
                "moduleid": 6,
```

```
            "status": "1"
        },
        {
            "moduleid": 7,
            "status": "1"
        },
        {
            "moduleid": 8,
            "status": "1"
        },
        {
            "moduleid": 9,
            "status": "1"
        },
        {
            "moduleid": 10,
            "status": "1"
        },
        {
            "moduleid": 11,
            "status": "1"
        },
        {
            "moduleid": 12,
            "status": "1"
        },
        {
            "moduleid": 13,
            "status": "1"
        },
        {
            "moduleid": 14,
            "status": "1"
        },
        {
            "moduleid": 15,
            "status": "1"
        },
        {
            "moduleid": 16,
            "status": "1"
        },
        {
            "moduleid": 17,
            "status": "1"
        },
        {
            "moduleid": 18,
            "status": "1"
        },
        {
            "moduleid": 19,
            "status": "1"
        },
        {
            "moduleid": 20,
            "status": "1"
        },
        {
            "moduleid": 21,
            "status": "1"
        },
```

```
            {
                "moduleid": 22,
                "status": "1"
            },
            {
                "moduleid": 23,
                "status": "1"
            },
            {
                "moduleid": 24,
                "status": "1"
            },
            {
                "moduleid": 25,
                "status": "1"
            },
            {
                "moduleid": 26,
                "status": "1"
            },
            {
                "moduleid": 27,
                "status": "1"
            },
            {
                "moduleid": 28,
                "status": "1"
            },
            {
                "moduleid": 29,
                "status": "1"
            },
            {
                "moduleid": 30,
                "status": "1"
            },
            {
                "moduleid": 31,
                "status": "1"
            },
            {
                "moduleid": 32,
                "status": "1"
            },
            {
                "moduleid": 33,
                "status": "1"
            }
        ],
        "modules.default_access": "1",
        "api.access": "1",
        "api.mode": "0",
        "api": [],
        "actions": [
            {
                "name": "edit_dashboards",
                "status": "1"
            },
            {
                "name": "edit_maps",
                "status": "1"
            },
```

```
                {
                    "name": "acknowledge_problems",
                    "status": "1"
                },
                {
                    "name": "suppress_problems",
                    "status": "1"
                },
                {
                    "name": "close_problems",
                    "status": "1"
                },
                {
                    "name": "change_severity",
                    "status": "1"
                },
                {
                    "name": "add_problem_comments",
                    "status": "1"
                },
                {
                    "name": "execute_scripts",
                    "status": "1"
                },
                {
                    "name": "manage_api_tokens",
                    "status": "1"
                },
                {
                    "name": "edit_maintenance",
                    "status": "1"
                },
                {
                    "name": "manage_scheduled_reports",
                    "status": "1"
                },
                {
                    "name": "manage_sla",
                    "status": "1"
                },
                {
                    "name": "invoke_execute_now",
                    "status": "1"
                },
                {
                    "name": "change_problem_ranking",
                    "status": "1"
                },
                {
                    "name": "edit_own_media",
                    "status": "1"
                },
                {
                    "name": "edit_user_media",
                    "status": "1"
                }
            ],
            "actions.default_access": "1"
        }
    }
],
"id": 1
```

```
}
```

Voir aussi

- Règles de rôle
- Utilisateur

Source

CRole::get() dans *ui/include/classes/api/services/CRole.php*.

role.update

Description

`object role.update(object/array roles)`

Cette méthode vous permet de mettre à jour des rôles existants.

> **Note:**
> Cette méthode est disponible uniquement pour le type d'utilisateur *Super admin*. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres du rôle utilisateur. Consultez User roles pour plus d'informations.

Paramètres

`(object/array)` Propriétés du rôle à mettre à jour.

La propriété `roleid` doit être définie pour chaque rôle, toutes les autres propriétés sont facultatives. Seules les propriétés transmises seront mises à jour, toutes les autres resteront inchangées.

En plus des propriétés de rôle standard, la méthode accepte les paramètres suivants.

Paramètre	Type	Description
rules	array	Règles d'accès à mettre à jour pour le rôle.

Valeurs retournées

`(object)` Retourne un objet contenant les identifiants des rôles mis à jour sous la propriété `roleids`.

Exemples

Désactivation de la possibilité d'exécuter des scripts

Mettez à jour le rôle avec l'ID « 5 » afin de désactiver la possibilité d'exécuter des scripts.

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "role.update",
    "params": [
        {
            "roleid": "5",
            "rules": {
                "actions": [
                    {
                        "name": "execute_scripts",
                        "status": "0"
                    }
                ]
            }
        }
    ],
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "roleids": [
            "5"
        ]
    },
    "id": 1
}
```

Limitation de l'accès à l'API

Mettez à jour le rôle avec l'ID « 5 » afin d'interdire l'appel de toutes les méthodes « create », « update » ou « delete ».

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "role.update",
    "params": [
        {
            "roleid": "5",
            "rules": {
                "api.access": "1",
                "api.mode": "0",
                "api": ["*.create", "*.update", "*.delete"]
            }
        }
    ],
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "roleids": [
            "5"
        ]
    },
    "id": 1
}
```

Source

CRole::update() dans *ui/include/classes/api/services/CRole.php*.

Script

Cette classe est conçue pour fonctionner avec des scripts.

Références d'objet :

- Script
  - Paramètres du webhook
- Debug
  - Entrée de journal

Méthodes disponibles :

- script.create - créer de nouveaux scripts
- script.delete - supprimer des scripts
- script.execute - exécuter des scripts
- script.get - récupérer des scripts
- script.getscriptsbyevents - récupérer des scripts pour des événements
- script.getscriptsbyhosts - récupérer des scripts pour des hôtes

- script.update - mettre à jour des scripts

Objet script

Les objets suivants sont directement liés à l'API `script`.

Script

L'objet script possède les propriétés suivantes.

Propriété	Type	Description
scriptid	ID	ID du script. Comportement de la propriété : - *en lecture seule* - *obligatoire* pour les opérations de mise à jour
name	string	Nom du script. Comportement de la propriété : - *obligatoire* pour les opérations de création
type	integer	Type de script. Valeurs possibles si `scope` est défini sur « action operation » : 0 - Script ; 1 - IPMI ; 2 - SSH ; 3 - TELNET ; 5 - Webhook. Valeurs possibles si `scope` est défini sur « manual host action » ou « manual event action » : 6 - URL. Comportement de la propriété : - *obligatoire* pour les opérations de création
command	string	Commande à exécuter. Comportement de la propriété : - *obligatoire* si `type` est défini sur « Script », « IPMI », « SSH », « TELNET » ou « Webhook »
scope	integer	Portée du script. Valeurs possibles : 1 - action operation ; 2 - manual host action ; 4 - manual event action. Comportement de la propriété : - *obligatoire* pour les opérations de création
execute_on	integer	Emplacement d'exécution du script. Valeurs possibles : 0 - exécution sur l'agent Zabbix ; 1 - exécution sur le serveur Zabbix. Cette valeur est *prise en charge* uniquement si l'exécution des scripts globaux est activée sur le serveur Zabbix ; 2 - *(par défaut)* exécution sur le serveur Zabbix ou le proxy. Comportement de la propriété : - *prise en charge* si `type` est défini sur « Script »

Propriété	Type	Description
menu_path	string	Dossiers séparés par des barres obliques formant une navigation de type menu dans l'interface lorsqu'un hôte ou un événement est sélectionné. Comportement de la propriété : - *prise en charge* si `scope` est défini sur « manual host action » ou « manual event action »
authtype	integer	Méthode d'authentification utilisée pour les scripts de type SSH. Valeurs possibles : 0 - mot de passe ; 1 - clé publique. Comportement de la propriété : - *prise en charge* si `type` est défini sur « SSH »
username	string	Nom d'utilisateur utilisé pour l'authentification. Comportement de la propriété : - *obligatoire* si `type` est défini sur « SSH » ou « TELNET »
password	string	Mot de passe utilisé pour les scripts SSH avec authentification par mot de passe et pour les scripts TELNET. Comportement de la propriété : - *prise en charge* si `type` est défini sur « SSH » et `authtype` sur « password », ou si `type` est défini sur « TELNET »
publickey	string	Nom du fichier de clé publique utilisé pour les scripts SSH avec authentification par clé publique. Comportement de la propriété : - *obligatoire* si `type` est défini sur « SSH » et `authtype` sur « public key »
privatekey	string	Nom du fichier de clé privée utilisé pour les scripts SSH avec authentification par clé publique. Comportement de la propriété : - *obligatoire* si `type` est défini sur « SSH » et `authtype` sur « public key »
port	string	Numéro de port utilisé pour les scripts SSH et TELNET. Comportement de la propriété : - *prise en charge* si `type` est défini sur « SSH » ou « TELNET »
groupid	ID	ID du groupe d'hôtes sur lequel le script peut être exécuté. Si cette valeur est définie sur « 0 », le script sera disponible pour tous les groupes d'hôtes. Valeur par défaut : 0.
usrgrpid	ID	ID du groupe d'utilisateurs autorisé à exécuter le script. Si cette valeur est définie sur « 0 », le script sera disponible pour tous les groupes d'utilisateurs. Valeur par défaut : 0. Comportement de la propriété : - *prise en charge* si `scope` est défini sur « manual host action » ou « manual event action »

Propriété	Type	Description
host_access	integer	Permissions sur l'hôte nécessaires pour exécuter le script. Valeurs possibles : 2 - *(par défaut)* lecture ; 3 - écriture. Comportement de la propriété : - *prise en charge* si `scope` est défini sur « manual host action » ou « manual event action »
confirmation	string	Texte de la fenêtre contextuelle de confirmation. La fenêtre contextuelle s'affiche lorsque vous tentez d'exécuter le script depuis l'interface Zabbix. Comportement de la propriété : - *prise en charge* si `scope` est défini sur « manual host action » ou « manual event action »
timeout	string	Délai d'expiration de l'exécution du script webhook, en secondes. Les suffixes de temps sont pris en charge (par exemple, `30s`, `1m`). Valeurs possibles : 1-60s. Valeur par défaut : 30s. Comportement de la propriété : - *obligatoire* si `type` est défini sur « Webhook »
parameters	array	Tableau de paramètres d'entrée du webhook. Comportement de la propriété : - *prise en charge* si `type` est défini sur « Webhook »
description	string	Description du script.
url	string	URL définie par l'utilisateur. Comportement de la propriété : - *obligatoire* si `type` est défini sur « URL »
new_window	integer	Ouvrir l'URL dans une nouvelle fenêtre. Valeurs possibles : 0 - Non ; 1 - *(par défaut)* Oui. Comportement de la propriété : - *prise en charge* si `type` est défini sur « URL »
manualinput	integer	Indique si le script accepte une saisie fournie par l'utilisateur. Valeurs possibles : 0 - *(par défaut)* Désactivée ; 1 - Activée ; Comportement de la propriété : - *prise en charge* si `scope` est défini sur « manual host action » ou « manual event action »
manualinput_prompt	string	Texte de l'invite de saisie manuelle. Comportement de la propriété : - *obligatoire* si `manualinput` est défini sur « Enabled »
manualinput_validator	string	Champ de type chaîne utilisé pour valider la saisie fournie par l'utilisateur. La chaîne se compose soit d'une expression régulière, soit d'un ensemble de valeurs séparées par des virgules. Comportement de la propriété : - *obligatoire* si `manualinput` est défini sur « Enabled »

Propriété	Type	Description
manualinput_validator_type	integer	Détermine le type de saisie utilisateur attendu. Valeurs possibles : 0 - *(par défaut)* String. Indique que manualinput_validator doit être traité comme une expression régulière ; 1 - List. Indique que manualinput_validator doit être traité comme une liste de valeurs d'entrée possibles séparées par des virgules. Comportement de la propriété : - *prise en charge* si `manualinput` est défini sur « Enabled »
manualinput_default_value	string	Valeur par défaut utilisée pour remplir automatiquement la saisie utilisateur. Comportement de la propriété : - *prise en charge* si `manualinput_validator_type` est défini sur « String »

Paramètres du webhook

Les paramètres transmis au script de webhook lors de son appel possèdent les propriétés suivantes.

Propriété	Type	Description
name	string	Nom du paramètre. Comportement de la propriété : - *obligatoire*
value	string	Valeur du paramètre. Prend en charge les macros.

Débogage

Informations de débogage du script webhook exécuté. L'objet debug possède les propriétés suivantes.

Propriété	Type	Description
logs	array	Tableau des entrées de journal.
ms	string	Durée d'exécution du script en millisecondes.

Entrée de journal

L'objet d'entrée de journal possède les propriétés suivantes.

Propriété	Type	Description
level	integer	Niveau de journalisation.
ms	string	Temps écoulé en millisecondes depuis l'exécution du script avant l'ajout de l'entrée de journal.
message	string	Message du journal.

script.create

Description

`object script.create(object/array scripts)`

Cette méthode permet de créer de nouveaux scripts.

Note:
Cette méthode est disponible uniquement pour le type d'utilisateur *Super admin*. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres des rôles utilisateur. Voir User roles pour plus d'informations.

Paramètres

(object/array) Scripts à créer.

La méthode accepte les scripts avec les propriétés de script standard.

Valeurs retournées

(object) Retourne un objet contenant les identifiants des scripts créés sous la propriété scriptids. L'ordre des ID retournés correspond à l'ordre des scripts transmis.

Exemples

Créer un script de webhook

Créez un script de webhook qui envoie une requête HTTP à un service externe.

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "script.create",
    "params": {
        "name": "Script de webhook",
        "command": "try {\n var request = new HttpRequest(),\n response,\n data;\n\n request.addHeader('Co
        "scope": 1,
        "type": 5,
        "timeout": "40s",
        "parameters": [
            {
                "name": "token",
                "value": "{$WEBHOOK.TOKEN}"
            },
            {
                "name": "host",
                "value": "{HOST.HOST}"
            },
            {
                "name": "v",
                "value": "2.2"
            }
        ]
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "scriptids": [
            "3"
        ]
    },
    "id": 1
}
```

Créer un script SSH

Créez un script SSH avec authentification par clé publique pouvant être exécuté sur un hôte et disposant d'un menu contextuel.

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "script.create",
    "params": {
        "name": "SSH script",
        "command": "my script command",
```

```
        "type": 2,
        "authtype": 1,
        "username": "John",
        "publickey": "pub.key",
        "privatekey": "priv.key",
        "password": "secret",
        "port": "12345",
        "scope": 2,
        "menu_path": "All scripts/SSH",
        "usrgrpid": "7",
        "groupid": "4"
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "scriptids": [
            "5"
        ]
    },
    "id": 1
}
```

Créer un script personnalisé

Créez un script personnalisé qui redémarrera un serveur. Le script nécessitera un accès en écriture à l'hôte et invitera l'utilisateur à saisir une entrée manuelle. Une fois la saisie envoyée avec succès, le script affichera un message de confirmation dans le frontend.

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "script.create",
    "params": {
        "name": "Reboot server",
        "command": "reboot server {MANUALINPUT}",
        "type": 0,
        "scope": 2,
        "confirmation": "Are you sure you would like to reboot the server {MANUALINPUT}?",
        "manualinput": 1,
        "manualinput_prompt": "Which server you want to reboot?",
        "manualinput_validator": "[1-9]",
        "manualinput_validator_type": 0,
        "manualinput_default_value": "1"
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "scriptids": [
            "4"
        ]
    },
    "id": 1
}
```

Créer un script de type URL

Créez un script de type URL pour la portée hôte, qui reste dans la même fenêtre et comporte un texte de confirmation.

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "script.create",
    "params": {
        "name": "URL script",
        "type": 6,
        "scope": 2,
        "url": "http://zabbix/ui/zabbix.php?action=host.edit&hostid={HOST.ID}",
        "confirmation": "Edit host {HOST.NAME}?",
        "new_window": 0
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "scriptids": [
            "56"
        ]
    },
    "id": 1
}
```

Créer un script de type URL avec saisie manuelle

Créez un script de type URL pour la portée d'événement, qui s'ouvre dans une nouvelle fenêtre et dispose d'une saisie manuelle.

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "script.create",
    "params": {
        "name": "URL script with manual input",
        "type": 6,
        "scope": 4,
        "url": "http://zabbix/ui/zabbix.php?action={MANUALINPUT}",
        "new_window": 1,
        "manualinput": 1,
        "manualinput_prompt": "Select a page to open:",
        "manualinput_validator": "dashboard.view,script.list,actionlog.list",
        "manualinput_validator_type": 1
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "scriptids": [
            "57"
        ]
    },
    "id": 1
}
```

Source

CScript::create() dans *ui/include/classes/api/services/CScript.php*.

script.delete

Description

`object script.delete(array scriptIds)`

Cette méthode vous permet de supprimer des scripts.

> **Note:**
> Cette méthode est disponible uniquement pour le type d'utilisateur *Super admin*. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres des rôles utilisateur. Voir User roles pour plus d'informations.

Paramètres

`(array)` ID des scripts à supprimer.

Valeurs de retour

`(object)` Renvoie un objet contenant les ID des scripts supprimés dans la propriété `scriptids`.

Exemples

Supprimer plusieurs scripts

Supprimez deux scripts.

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "script.delete",
    "params": [
        "3",
        "4"
    ],
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "scriptids": [
            "3",
            "4"
        ]
    },
    "id": 1
}
```

Source

CScript::delete() dans *ui/include/classes/api/services/CScript.php*.

script.execute

Description

`object script.execute(object parameters)`

Cette méthode vous permet d'exécuter un script sur un hôte ou un événement. À l'exception des scripts de type URL ; ceux-ci ne sont pas exécutables.

> **Note:**
> Cette méthode est disponible pour les utilisateurs de tout type. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres des rôles utilisateur. Consultez Rôles utilisateur pour plus d'informations.

Paramètres

(object) Paramètres contenant l'ID du script à exécuter, soit l'ID de l'hôte, soit l'ID de l'événement et la valeur manualinput.

Paramètre	Type	Description
scriptid	ID	ID du script à exécuter. Comportement du paramètre : - *obligatoire*
hostid	ID	ID de l'hôte sur lequel exécuter le script. Comportement du paramètre : - *obligatoire* si `eventid` n'est pas défini
eventid	ID	ID de l'événement sur lequel exécuter le script. Comportement du paramètre : - *obligatoire* si `hostid` n'est pas défini
manualinput	string	Valeur fournie par l'utilisateur pour exécuter le script, en remplaçant la macro {MANUALINPUT}.

Valeurs de retour

(object) Renvoie le résultat de l'exécution du script.

Propriété	Type	Description
response	string	Indique si le script a été exécuté avec succès. Valeur possible - `success`.
value	string	Sortie du script.
debug	object	Contient un objet debug si un script webhook est exécuté. Pour les autres types de scripts, il contient un objet vide.

Exemples

Exécuter un script webhook

Exécutez un script webhook qui envoie une requête HTTP à un service externe.

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "script.execute",
    "params": {
        "scriptid": "4",
        "hostid": "30079"
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "response": "success",
        "value": "{\"status\":\"sent\",\"timestamp\":\"1611235391\"}",
        "debug": {
            "logs": [
                {
                    "level": 3,
                    "ms": 480,
                    "message": "[Webhook Script] HTTP status: 200."
                }
            ],
```

```
                "ms": 495
            }
        },
        "id": 1
}
```


Exécuter un script personnalisé

Exécutez un script « ping » sur un hôte.

Requête:


```
{
    "jsonrpc": "2.0",
    "method": "script.execute",
    "params": {
        "scriptid": "1",
        "hostid": "30079"
    },
    "id": 1
}
```


Réponse :


```
{
    "jsonrpc": "2.0",
    "result": {
        "response": "success",
        "value": "PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.\n64 bytes from 127.0.0.1: icmp_req=1 tt
        "debug": []
    },
    "id": 1
}
```


Exécuter un script personnalisé avec saisie manuelle

Exécutez un script ”ping” avec la commande ”ping -c {MANUALINPUT} {HOST.CONN}; case $? in [01]) true;; *) false;; esac” sur un hôte.

Requête :


```
{
    "jsonrpc": "2.0",
    "method": "script.execute",
    "params": {
        "scriptid": "7",
        "hostid": "30079",
        "manualinput": "2"
    },
    "id": 1
}
```


Réponse :


```
{
    "jsonrpc": "2.0",
    "result": {
        "response": "success",
        "value": "PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.\n64 bytes from 127.0.0.1: icmp_seq=1 tt
        "debug": []
    },
    "id": 1
}
```


Source

CScript::execute() dans *ui/include/classes/api/services/CScript.php*.

script.get

Description

`integer/array script.get(object parameters)`

La méthode permet de récupérer des scripts selon les paramètres fournis.

> **Note:**
> Cette méthode est disponible pour les utilisateurs de tout type. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres des rôles utilisateur. Consultez Rôles d'utilisateur pour plus d'informations.

Paramètres

`(object)` Paramètres définissant la sortie souhaitée.

La méthode prend en charge les paramètres suivants.

Paramètre	Type	Description
groupids	ID/array	Retourner uniquement les scripts pouvant être exécutés sur les groupes d'hôtes donnés.
hostids	ID/array	Retourner uniquement les scripts pouvant être exécutés sur les hôtes donnés.
scriptids	ID/array	Retourner uniquement les scripts avec les ID donnés.
usrgrpids	ID/array	Retourner uniquement les scripts pouvant être exécutés par les utilisateurs des groupes d'utilisateurs donnés.
selectHostGroups	query	Retourner une propriété `hostgroups` avec les groupes d'hôtes sur lesquels le script peut être exécuté.
selectHosts	query	Retourner une propriété `hosts` avec les hôtes sur lesquels le script peut être exécuté.
selectActions	query	Retourner une propriété `actions` avec les actions auxquelles le script est associé.
sortfield	string/array	Trier le résultat selon les propriétés données. Valeurs possibles : `scriptid`, `name`.
countOutput	boolean	Ces paramètres sont décrits dans le commentaire de référence.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Valeurs retournées

`(integer/array)` Retourne soit :

- un tableau d'objets ;
- le nombre d'objets récupérés, si le paramètre `countOutput` a été utilisé.

Exemples

Récupérer tous les scripts

Récupère tous les scripts configurés.

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "script.get",
```

```
    "params": {
        "output": "extend"
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "scriptid": "1",
            "name": "Ping",
            "command": "/bin/ping -c 3 {HOST.CONN} 2>&1",
            "host_access": "2",
            "usrgrpid": "0",
            "groupid": "0",
            "description": "",
            "confirmation": "",
            "type": "0",
            "execute_on": "1",
            "timeout": "30s",
            "scope": "2",
            "port": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "menu_path": "",
            "url": "",
            "new_window": "1",
            "manualinput": "0",
            "manualinput_prompt": "",
            "manualinput_validator": "",
            "manualinput_validator_type": "0",
            "manualinput_default_value": "",
            "parameters": []
        },
        {
            "scriptid": "2",
            "name": "Traceroute",
            "command": "/usr/bin/traceroute {HOST.CONN} 2>&1",
            "host_access": "2",
            "usrgrpid": "0",
            "groupid": "0",
            "description": "",
            "confirmation": "",
            "type": "0",
            "execute_on": "1",
            "timeout": "30s",
            "scope": "2",
            "port": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "menu_path": "",
            "url": "",
            "new_window": "1",
            "manualinput": "0",
```

```
            "manualinput_prompt": "",
            "manualinput_validator": "",
            "manualinput_validator_type": "0",
            "manualinput_default_value": "",
            "parameters": []
        },
        {
            "scriptid": "3",
            "name": "Detect operating system",
            "command": "sudo /usr/bin/nmap -O {HOST.CONN} 2>&1",
            "host_access": "2",
            "usrgrpid": "7",
            "groupid": "0",
            "description": "",
            "confirmation": "",
            "type": "0",
            "execute_on": "1",
            "timeout": "30s",
            "scope": "2",
            "port": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "menu_path": "",
            "url": "",
            "new_window": "1",
            "manualinput": "0",
            "manualinput_prompt": "",
            "manualinput_validator": "",
            "manualinput_validator_type": "0",
            "manualinput_default_value": "",
            "parameters": []
        },
        {
            "scriptid": "4",
            "name": "Webhook",
            "command": "try {\n var request = new HttpRequest(),\n response,\n data;\n\n request.addHeader
            "host_access": "2",
            "usrgrpid": "7",
            "groupid": "0",
            "description": "",
            "confirmation": "",
            "type": "5",
            "execute_on": "1",
            "timeout": "30s",
            "scope": "2",
            "port": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "menu_path": "",
            "url": "",
            "new_window": "1",
            "manualinput": "0",
            "manualinput_prompt": "",
            "manualinput_validator": "",
            "manualinput_validator_type": "0",
            "manualinput_default_value": "",
```

```
            "parameters": [
                {
                    "name": "token",
                    "value": "{$WEBHOOK.TOKEN}"
                },
                {
                    "name": "host",
                    "value": "{HOST.HOST}"
                },
                {
                    "name": "v",
                    "value": "2.2"
                }
            ]
        },
        {
            "scriptid": "5",
            "name": "URL",
            "command": "",
            "host_access": "2",
            "usrgrpid": "0",
            "groupid": "0",
            "description": "",
            "confirmation": "Go to {HOST.NAME}?",
            "type": "6",
            "execute_on": "1",
            "timeout": "30s",
            "scope": "4",
            "port": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "menu_path": "",
            "url": "http://zabbix/ui/zabbix.php?action=latest.view&hostids[]={HOST.ID}",
            "new_window": "0",
            "manualinput": "0",
            "manualinput_prompt": "",
            "manualinput_validator": "",
            "manualinput_validator_type": "0",
            "manualinput_default_value": "",
            "parameters": []
        },
        {
            "scriptid": "6",
            "name": "URL with user input",
            "command": "",
            "host_access": "2",
            "usrgrpid": "0",
            "groupid": "0",
            "description": "",
            "confirmation": "Open zabbix page {MANUALINPUT}?",
            "type": "6",
            "execute_on": "1",
            "timeout": "30s",
            "scope": "2",
            "port": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
```

```
            "privatekey": "",
            "menu_path": "",
            "url": "http://zabbix/ui/zabbix.php?action={MANUALINPUT}",
            "new_window": "0",
            "manualinput": "1",
            "manualinput_prompt": "Select a page to open:",
            "manualinput_validator": "dashboard.view,script.list,actionlog.list",
            "manualinput_validator_type": "1",
            "parameters": []
        }
    ],
    "id": 1
}
```

Voir aussi

- Hôte
- Groupe d'hôtes

Source

CScript::get() dans *ui/include/classes/api/services/CScript.php*.

script.getscriptsbyevents

Description

`object script.getscriptsbyevents(object parameters)`

Cette méthode permet de récupérer tous les scripts disponibles pour l'événement donné ou un script spécifique si l'ID du script est fourni. Lorsque manualinput est fourni, il remplace la macro `{MANUALINPUT}` par la valeur spécifiée.

> **Note:**
> Cette méthode est disponible pour les utilisateurs de tout type. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres des rôles utilisateur. Voir Rôles d'utilisateur pour plus d'informations.

Paramètres

`(object/array)` La méthode accepte un objet ou un tableau d'objets avec les paramètres suivants.

Paramètre	Type	Description
eventid	ID	ID de l'événement pour lequel retourner les scripts. Doit être unique. Comportement des paramètres: - *obligatoire*
scriptid	ID	ID du script à retourner.
manualinput	string	Valeur de la macro {MANUALINPUT} fournie par l'utilisateur.

Valeurs de retour

`(object)` Renvoie un objet avec les ID d'événement comme propriétés et des tableaux de scripts disponibles comme valeurs. Si l'ID du script est fourni, la valeur associée est un tableau contenant le script spécifique.

> **Note:**
> La méthode développera automatiquement les macros dans le texte `confirmation`, le texte `manualinput prompt` et l'`url`.
> Si le paramètre manualinput est fourni, la macro {MANUALINPUT} sera résolue avec la valeur spécifiée.

Exemples

Récupérer les scripts par ID d'événement

Récupérez tous les scripts disponibles pour les événements « 632 » et « 614 ».

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "script.getscriptsbyevents",
    "params": [
      {
         "eventid":  "632"
      },
      {
         "eventid":  "614"
      }
    ],
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "632": [
            {
                "scriptid": "3",
                "name": "Détecter le système d'exploitation",
                "command": "sudo /usr/bin/nmap -O {HOST.CONN} 2>&1",
                "host_access": "2",
                "usrgrpid": "7",
                "groupid": "0",
                "description": "",
                "confirmation": "",
                "type": "0",
                "execute_on": "1",
                "timeout": "30s",
                "scope": "4",
                "port": "",
                "authtype": "0",
                "username": "",
                "password": "",
                "publickey": "",
                "privatekey": "",
                "menu_path": "",
                "url": "",
                "new_window": "1",
                "manualinput": "0",
                "manualinput_prompt": "",
                "manualinput_validator_type": "0",
                "manualinput_validator": "",
                "manualinput_default_value": "",
                "parameters": []
            },
            {
                "scriptid": "1",
                "name": "Ping",
                "command": "/bin/ping -c 3 {HOST.CONN} 2>&1",
                "host_access": "2",
                "usrgrpid": "0",
                "groupid": "0",
                "description": "",
                "confirmation": "",
                "type": "0",
                "execute_on": "1",
                "timeout": "30s",
                "scope": "4",
                "port": "",
```

```
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "menu_path": "",
            "url": "",
            "new_window": "1",
            "manualinput": "0",
            "manualinput_prompt": "",
            "manualinput_validator_type": "0",
            "manualinput_validator": "",
            "manualinput_default_value": "",
            "parameters": []
        },
        {
            "scriptid": "4",
            "name": "Ouvrir la page Zabbix",
            "command": "",
            "host_access": "2",
            "usrgrpid": "0",
            "groupid": "0",
            "description": "",
            "confirmation": "Êtes-vous sûr de vouloir ouvrir la page *UNKNOWN* ?",
            "type": "6",
            "execute_on": "2",
            "timeout": "30s",
            "scope": "4",
            "port": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "menu_path": "",
            "url": "http://localhost/ui/zabbix.php?action=*UNKNOWN*",
            "new_window": "1",
            "manualinput": "1",
            "manualinput_prompt": "Page Zabbix à ouvrir :",
            "manualinput_validator_type": "1",
            "manualinput_validator": "dashboard.view,discovery.view",
            "manualinput_default_value": "",
            "parameters": []
        },
        {
            "scriptid": "2",
            "name": "Traceroute",
            "command": "/usr/bin/traceroute {HOST.CONN} 2>&1",
            "host_access": "2",
            "usrgrpid": "0",
            "groupid": "0",
            "description": "",
            "confirmation": "",
            "type": "0",
            "execute_on": "1",
            "timeout": "30s",
            "scope": "4",
            "port": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
```

```
                "privatekey": "",
                "menu_path": "",
                "url": "",
                "new_window": "1",
                "manualinput": "0",
                "manualinput_prompt": "",
                "manualinput_validator_type": "0",
                "manualinput_validator": "",
                "manualinput_default_value": "",
                "parameters": []
            }
        ],
        "614": [
            {
                "scriptid": "3",
                "name": "Détecter le système d'exploitation",
                "command": "sudo /usr/bin/nmap -O {HOST.CONN} 2>&1",
                "host_access": "2",
                "usrgrpid": "7",
                "groupid": "0",
                "description": "",
                "confirmation": "",
                "type": "0",
                "execute_on": "1",
                "timeout": "30s",
                "scope": "4",
                "port": "",
                "authtype": "0",
                "username": "",
                "password": "",
                "publickey": "",
                "privatekey": "",
                "menu_path": "",
                "url": "",
                "new_window": "1",
                "manualinput": "0",
                "manualinput_prompt": "",
                "manualinput_validator_type": "1",
                "manualinput_validator": "",
                "manualinput_default_value": "",
                "parameters": []
            },
            {
                "scriptid": "1",
                "name": "Ping",
                "command": "/bin/ping -c 3 {HOST.CONN} 2>&1",
                "host_access": "2",
                "usrgrpid": "0",
                "groupid": "0",
                "description": "",
                "confirmation": "",
                "type": "0",
                "execute_on": "1",
                "timeout": "30s",
                "scope": "4",
                "port": "",
                "authtype": "0",
                "username": "",
                "password": "",
                "publickey": "",
                "privatekey": "",
                "menu_path": "",
```

```
            "url": "",
            "new_window": "1",
            "manualinput": "0",
            "manualinput_prompt": "",
            "manualinput_validator_type": "0",
            "manualinput_validator": "",
            "manualinput_default_value": "",
            "parameters": []
        },
        {
            "scriptid": "4",
            "name": "Ouvrir la page Zabbix",
            "command": "",
            "host_access": "2",
            "usrgrpid": "0",
            "groupid": "0",
            "description": "",
            "confirmation": "Êtes-vous sûr de vouloir ouvrir la page *UNKNOWN* ?",
            "type": "6",
            "execute_on": "2",
            "timeout": "30s",
            "scope": "4",
            "port": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "menu_path": "",
            "url": "http://localhost/ui/zabbix.php?action=*UNKNOWN*",
            "new_window": "1",
            "manualinput": "1",
            "manualinput_prompt": "Page Zabbix à ouvrir :",
            "manualinput_validator_type": "1",
            "manualinput_validator": "dashboard.view,discovery.view",
            "manualinput_default_value": "",
            "parameters": []
        },
        {
            "scriptid": "2",
            "name": "Traceroute",
            "command": "/usr/bin/traceroute {HOST.CONN} 2>&1",
            "host_access": "2",
            "usrgrpid": "0",
            "groupid": "0",
            "description": "",
            "confirmation": "",
            "type": "0",
            "execute_on": "1",
            "timeout": "30s",
            "scope": "4",
            "port": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "menu_path": "",
            "url": "",
            "new_window": "1",
            "manualinput": "0",
            "manualinput_prompt": "",
```

```
                "manualinput_validator_type": "0",
                "manualinput_validator": "",
                "manualinput_default_value": "",
                "parameters": []
            }
        ]
    },
    "id": 1
}
```

Récupérer un script spécifique avec la valeur manualinput.

Récupérez le script avec l'ID "4" sur l'événement "632" avec la valeur manualinput "dashboard.view".

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "script.getscriptsbyevents",
    "params": [
      {
        "eventid":  "632",
        "scriptid": "4",
        "manualinput": "dashboard.view"
      }
  ],
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
      "632": [
        {
          "scriptid": "4",
          "name": "Open Zabbix page",
          "command": "",
          "host_access": "2",
          "usrgrpid": "0",
          "groupid": "0",
          "description": "",
          "confirmation": "Are you sure you want to open page dashboard.view?",
          "type": "6",
          "execute_on": "2",
          "timeout": "30s",
          "scope": "4",
          "port": "",
          "authtype": "0",
          "username": "",
          "password": "",
          "publickey": "",
          "privatekey": "",
          "menu_path": "",
          "url": "http://localhost/ui/zabbix.php?action=dashboard.view",
          "new_window": "1",
          "manualinput": "1",
          "manualinput_prompt": "Zabbix page to open:",
          "manualinput_validator_type": "1",
          "manualinput_validator": "dashboard.view,discovery.view",
          "manualinput_default_value": "",
          "parameters": []
        }
      ]
```

```
    },
    "id": 1
}
```

Source

CScript::getScriptsByEvents() dans *ui/include/classes/api/services/CScript.php*.

script.getscriptsbyhosts

Description

`object script.getscriptsbyhosts(object parameters)`

Cette méthode permet de récupérer tous les scripts disponibles sur l'hôte donné ou un script spécifique si l'ID du script est fourni. Lorsque manualinput est fourni, il remplace la macro `{MANUALINPUT}` par la valeur spécifiée.

Note:
Cette méthode est disponible pour les utilisateurs de tout type. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres des rôles utilisateur. Voir Rôles d'utilisateur pour plus d'informations.

Paramètres

`(object/array)` La méthode accepte un objet ou un tableau d'objets avec les paramètres suivants.

Paramètre	Type	Description
hostid	ID	ID de l'hôte pour lequel retourner les scripts. Doit être unique. Comportement des paramètres: - *obligatoire*
scriptid	ID	ID du script à retourner.
manualinput	string	Valeur de la macro {MANUALINPUT} fournie par l'utilisateur.

Valeurs de retour

`(object)` Renvoie un objet avec les ID d'hôte comme propriétés et des tableaux de scripts disponibles comme valeurs. Si l'ID du script est fourni, la valeur associée est un tableau contenant le script spécifique.

Note:
La méthode développera automatiquement les macros dans le texte `confirmation`, le texte `manualinput prompt` et l'`url`.
Si le paramètre manualinput est fourni, la macro {MANUALINPUT} sera résolue avec la valeur spécifiée.

Exemples

Récupérer les scripts par ID d'hôte

Récupérez tous les scripts disponibles sur les hôtes "30079" et "30073".

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "script.getscriptsbyhosts",
    "params": [
       {
          "hostid":  "30079"
       },
       {
          "hostid":  "30073"
       }
    ],
    "id": 1
}
```

Réponse :


```
{
    "jsonrpc": "2.0",
    "result": {
        "30079": [
            {
                "scriptid": "3",
                "name": "Détecter le système d'exploitation",
                "command": "sudo /usr/bin/nmap -O {HOST.CONN} 2>&1",
                "host_access": "2",
                "usrgrpid": "7",
                "groupid": "0",
                "description": "",
                "confirmation": "",
                "type": "0",
                "execute_on": "1",
                "timeout": "30s",
                "scope": "2",
                "port": "",
                "authtype": "0",
                "username": "",
                "password": "",
                "publickey": "",
                "privatekey": "",
                "menu_path": "",
                "url": "",
                "new_window": "1",
                "manualinput": "0",
                "manualinput_prompt": "",
                "manualinput_validator_type": "0",
                "manualinput_validator": "",
                "manualinput_default_value": "",
                "parameters": []
            },
            {
                "scriptid": "1",
                "name": "Ping",
                "command": "/bin/ping -c 3 {HOST.CONN} 2>&1",
                "host_access": "2",
                "usrgrpid": "0",
                "groupid": "0",
                "description": "",
                "confirmation": "",
                "type": "0",
                "execute_on": "1",
                "timeout": "30s",
                "scope": "2",
                "port": "",
                "authtype": "0",
                "username": "",
                "password": "",
                "publickey": "",
                "privatekey": "",
                "menu_path": "",
                "url": "",
                "new_window": "1",
                "manualinput": "0",
                "manualinput_prompt": "",
                "manualinput_validator_type": "0",
                "manualinput_validator": "",
                "manualinput_default_value": "",
                "parameters": []
```

```
        },
        {
            "scriptid": "4",
            "name": "Ouvrir la page Zabbix",
            "command": "",
            "host_access": "2",
            "usrgrpid": "0",
            "groupid": "0",
            "description": "",
            "confirmation": "Êtes-vous sûr de vouloir ouvrir la page *UNKNOWN* ?",
            "type": "6",
            "execute_on": "2",
            "timeout": "30s",
            "scope": "2",
            "port": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "menu_path": "",
            "url": "http://localhost/ui/zabbix.php?action=*UNKNOWN*",
            "new_window": "1",
            "manualinput": "0",
            "manualinput_prompt": "Page Zabbix à ouvrir :",
            "manualinput_validator_type": "0",
            "manualinput_validator": "dashboard.view,discovery.view",
            "manualinput_default_value": "",
            "parameters": []
        },
        {
            "scriptid": "2",
            "name": "Traceroute",
            "command": "/usr/bin/traceroute {HOST.CONN} 2>&1",
            "host_access": "2",
            "usrgrpid": "0",
            "groupid": "0",
            "description": "",
            "confirmation": "",
            "type": "0",
            "execute_on": "1",
            "timeout": "30s",
            "scope": "2",
            "port": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "menu_path": "",
            "url": "",
            "new_window": "1",
            "manualinput": "0",
            "manualinput_prompt": "",
            "manualinput_validator_type": "0",
            "manualinput_validator": "",
            "manualinput_default_value": "",
            "parameters": []
        }
    ],
    "30073": [
        {
```

```
            "scriptid": "3",
            "name": "Détecter le système d'exploitation",
            "command": "sudo /usr/bin/nmap -O {HOST.CONN} 2>&1",
            "host_access": "2",
            "usrgrpid": "7",
            "groupid": "0",
            "description": "",
            "confirmation": "",
            "type": "0",
            "execute_on": "1",
            "timeout": "30s",
            "scope": "2",
            "port": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "menu_path": "",
            "url": "",
            "new_window": "1",
            "manualinput": "0",
            "manualinput_prompt": "",
            "manualinput_validator_type": "0",
            "manualinput_validator": "",
            "manualinput_default_value": "",
            "parameters": []
        },
        {
            "scriptid": "1",
            "name": "Ping",
            "command": "/bin/ping -c 3 {HOST.CONN} 2>&1",
            "host_access": "2",
            "usrgrpid": "0",
            "groupid": "0",
            "description": "",
            "confirmation": "",
            "type": "0",
            "execute_on": "1",
            "timeout": "30s",
            "scope": "2",
            "port": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "menu_path": "",
            "url": "",
            "new_window": "1",
            "manualinput": "0",
            "manualinput_prompt": "",
            "manualinput_validator_type": "0",
            "manualinput_validator": "",
            "manualinput_default_value": "",
            "parameters": []
        },
        {
            "scriptid": "4",
            "name": "Ouvrir la page Zabbix",
            "command": "",
            "host_access": "2",
```

```
                "usrgrpid": "0",
                "groupid": "0",
                "description": "",
                "confirmation": "Êtes-vous sûr de vouloir ouvrir la page *UNKNOWN* ?",
                "type": "6",
                "execute_on": "2",
                "timeout": "30s",
                "scope": "2",
                "port": "",
                "authtype": "0",
                "username": "",
                "password": "",
                "publickey": "",
                "privatekey": "",
                "menu_path": "",
                "url": "http://localhost/ui/zabbix.php?action=*UNKNOWN*",
                "new_window": "1",
                "manualinput": "1",
                "manualinput_prompt": "Page Zabbix à ouvrir :",
                "manualinput_validator_type": "1",
                "manualinput_validator": "dashboard.view,discovery.view",
                "manualinput_default_value": "",
                "parameters": []
            },
            {
                "scriptid": "2",
                "name": "Traceroute",
                "command": "/usr/bin/traceroute {HOST.CONN} 2>&1",
                "host_access": "2",
                "usrgrpid": "0",
                "groupid": "0",
                "description": "",
                "confirmation": "",
                "type": "0",
                "execute_on": "1",
                "timeout": "30s",
                "scope": "2",
                "port": "",
                "authtype": "0",
                "username": "",
                "password": "",
                "publickey": "",
                "privatekey": "",
                "menu_path": "",
                "url": "",
                "new_window": "1",
                "manualinput": "0",
                "manualinput_prompt": "",
                "manualinput_validator_type": "0",
                "manualinput_validator": "",
                "manualinput_default_value": "",
                "parameters": []
            }
        ]
    },
    "id": 1
}
```


Récupérer un script spécifique avec une valeur de manualinput.

Récupérez le script avec l'ID "4" sur l'hôte "30079" avec la valeur de manualinput "dashboard.view".

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "script.getscriptsbyhosts",
    "params": [
      {
        "hostid":  "30079",
        "scriptid": "4",
        "manualinput": "dashboard.view"
      }
  ],
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
      "30079": [
        {
          "scriptid": "4",
          "name": "Open Zabbix page",
          "command": "",
          "host_access": "2",
          "usrgrpid": "0",
          "groupid": "0",
          "description": "",
          "confirmation": "Are you sure you want to open page dashboard.view?",
          "type": "6",
          "execute_on": "2",
          "timeout": "30s",
          "scope": "2",
          "port": "",
          "authtype": "0",
          "username": "",
          "password": "",
          "publickey": "",
          "privatekey": "",
          "menu_path": "",
          "url": "http://localhost/ui/zabbix.php?action=dashboard.view",
          "new_window": "1",
          "manualinput": "1",
          "manualinput_prompt": "Zabbix page to open:",
          "manualinput_validator_type": "1",
          "manualinput_validator": "dashboard.view,discovery.view",
          "manualinput_default_value": "",
          "parameters": []
        }
      ]
    },
    "id": 1
}
```

Source

CScript::getScriptsByHosts() dans *ui/include/classes/api/services/CScript.php*.

script.update

Description

`object script.update(object/array scripts)`

Cette méthode permet de mettre à jour des scripts existants.

Note:
Cette méthode est disponible uniquement pour le type d'utilisateur *Super admin*. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres des rôles utilisateur. Consultez User roles pour plus d'informations.

Paramètres

`(object/array)` Propriétés du script à mettre à jour.

La propriété `scriptid` doit être définie pour chaque script, toutes les autres propriétés sont facultatives. Seules les propriétés transmises seront mises à jour, toutes les autres resteront inchangées. Une exception est le changement de la propriété `type` de 5 (Webhook) à other : la propriété `parameters` sera nettoyée.

Valeurs retournées

`(object)` Retourne un objet contenant les identifiants des scripts mis à jour sous la propriété `scriptids`.

Exemples

Modifier la commande du script

Modifiez la commande du script en "/bin/ping -c 10 {HOST.CONN} 2>&1".

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "script.update",
    "params": {
        "scriptid": "1",
        "command": "/bin/ping -c 10 {HOST.CONN} 2>&1"
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "scriptids": [
            "1"
        ]
    },
    "id": 1
}
```

Modifier la commande du script et ajouter une entrée manuelle

Modifiez la commande du script en "/bin/ping -c {MANUALINPUT} {HOST.CONN} 2>&1".

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "script.update",
    "params": {
      "scriptid": "1",
      "command": "/bin/ping -c {MANUALINPUT} {HOST.CONN} 2>&1",
      "manualinput": "1",
      "manualinput_prompt": "Indiquez le nombre de paquets ICMP à envoyer avec la commande ping",
      "manualinput_validator": "^(?:[1-9]|10)$",
      "manualinput_validator_type": "0",
      "manualinput_default_value": "10"
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "scriptids": [
            "1"
        ]
    },
    "id": 1
}
```

Source

CScript::update() dans *ui/include/classes/api/services/CScript.php*.

Scénario web

Cette classe est conçue pour fonctionner avec les scénarios web.

Références d'objet :

- Scénario web
- Étape de scénario
  - Champ HTTP
- Tag de scénario web

Méthodes disponibles :

- httptest.create - créer de nouveaux scénarios web
- httptest.delete - supprimer des scénarios web
- httptest.get - récupérer des scénarios web
- httptest.update - mettre à jour des scénarios web

Objet de scénario web

Les objets suivants sont directement liés à l'API `webcheck`.

Scénario web

L'objet scénario web possède les propriétés suivantes.

Property	Type	Description
httptestid	ID	ID du scénario web. Comportement de la propriété : - *en lecture seule* - *requis* pour les opérations de mise à jour
hostid	ID	ID de l'hôte auquel appartient le scénario web. Comportement de la propriété : - *constant* - *requis* pour les opérations de création
name	string	Nom du scénario web. Comportement de la propriété : - *requis* pour les opérations de création
agent	string	Chaîne user agent qui sera utilisée par le scénario web. Valeur par défaut : Zabbix

Property	Type	Description
authentication	integer	Méthode d'authentification qui sera utilisée par le scénario web. Valeurs possibles : 0 - *(par défaut)* aucune ; 1 - authentification HTTP de base ; 2 - authentification NTLM ; 3 - authentification Kerberos ; 4 - authentification Digest.
delay	string	Intervalle d'exécution du scénario web. Accepte un nombre de secondes (par exemple, 1, 60, 300), une valeur avec un suffixe temporel (par exemple, 30s, 1m, 2h, 1d) ou une macro utilisateur. Valeur par défaut : 1m.
headers	array	En-têtes HTTP qui seront envoyés lors de l'exécution d'une requête.
http_password	string	Mot de passe utilisé pour l'authentification HTTP de base, NTLM, Kerberos ou Digest.
http_proxy	string	Proxy qui sera utilisé par le scénario web, au format *http://[username[:password]@]proxy.example.com[:port]*.
http_user	string	Nom d'utilisateur utilisé pour l'authentification HTTP de base, NTLM, Kerberos ou Digest.
retries	integer	Nombre de fois qu'un scénario web tentera d'exécuter chaque étape avant d'échouer. Valeur par défaut : 1.
ssl_cert_file	string	Nom du fichier de certificat SSL utilisé pour l'authentification du client (doit être au format PEM).
ssl_key_file	string	Nom du fichier de clé privée SSL utilisé pour l'authentification du client (doit être au format PEM).
ssl_key_password	string	Mot de passe de la clé privée SSL.
status	integer	Indique si le scénario web est activé. Valeurs possibles : 0 - *(par défaut)* activé ; 1 - désactivé.
templateid	ID	ID du scénario web du modèle parent. Comportement de la propriété : - *en lecture seule*
variables	array	Variables du scénario web.
verify_host	integer	Indique s'il faut vérifier que le nom d'hôte de la connexion correspond à celui indiqué dans le certificat de l'hôte. Valeurs possibles : 0 - *(par défaut)* ne pas vérifier l'hôte ; 1 - vérifier l'hôte.
verify_peer	integer	Indique s'il faut vérifier que le certificat de l'hôte est authentique. Valeurs possibles : 0 - *(par défaut)* ne pas vérifier le pair ; 1 - vérifier le pair.
uuid	string	Identifiant unique global, utilisé pour lier les scénarios web importés à ceux qui existent déjà. Généré automatiquement s'il n'est pas fourni. Comportement de la propriété : - *pris en charge* si le scénario web appartient à un modèle

Étape du scénario

L'objet d'étape du scénario définit une vérification spécifique d'un scénario web. Il possède les propriétés suivantes.

Property	Type	Description
name	string	Nom de l'étape du scénario. Comportement de la propriété : - *required*
no	integer	Numéro de séquence de l'étape dans un scénario web. Comportement de la propriété : - *required*
url	string	URL à vérifier. Comportement de la propriété : - *required*
follow_redirects	integer	Indique s'il faut suivre les redirections HTTP. Valeurs possibles : 0 - ne pas suivre les redirections ; 1 - *(par défaut)* suivre les redirections.
headers	array	En-têtes HTTP qui seront envoyés lors de l'exécution d'une requête. Les en-têtes de l'étape du scénario remplaceront ceux spécifiés pour le scénario web.
posts	string/array	Variables HTTP POST sous forme de chaîne (données POST brutes) ou de tableau de champs HTTP (données de champs de formulaire).
required	string	Texte qui doit être présent dans la réponse.
retrieve_mode	integer	Partie de la réponse HTTP que l'étape du scénario doit récupérer. Valeurs possibles : 0 - *(par défaut)* corps uniquement ; 1 - en-têtes uniquement ; 2 - en-têtes et corps.
status_codes	string	Plages de codes d'état HTTP requis, séparées par des virgules.
timeout	string	Délai d'expiration de la requête en secondes. Accepte un nombre de secondes (par exemple, 1, 60, 300), une valeur avec un suffixe de temps (par exemple, 30s, 1m, 1h) ou une macro utilisateur. Valeur par défaut : 15s. Maximum : 1h. Minimum : 1s.
variables	array	Variables de l'étape du scénario.
query_fields	array	Champs de requête - tableau de champs HTTP qui seront ajoutés à l'URL lors de l'exécution d'une requête.

Champ HTTP

L'objet champ HTTP définit le nom et la valeur utilisés pour spécifier les variables du scénario web, les en-têtes HTTP, ainsi que les champs POST ou les champs de requête. Il possède les propriétés suivantes.

Property	Type	Description
name	string	Nom de l'en-tête/de la variable/du champ POST ou GET. Comportement de la propriété: - *obligatoire*
value	string	Valeur de l'en-tête/de la variable/du champ POST ou GET. Comportement de la propriété: - *obligatoire*

Balise de scénario web

L'objet de balise de scénario web possède les propriétés suivantes.

Propriété	Type	Description
tag	string	Nom de la balise du scénario web. Comportement de la propriété: - *obligatoire*
value	string	Valeur de la balise du scénario web. Comportement de la propriété: - *lecture seule*.
object	integer	Type d'objet dont la balise a été héritée. Valeurs possibles : 0 - modèle; 1 - hôte. Comportement de la propriété: - *lecture seule*.
objectid	ID	ID de l'objet dont la balise a été héritée. Comportement de la propriété: - *lecture seule*.

httptest.create

Description

`object httptest.create(object/array webScenarios)`

Cette méthode vous permet de créer de nouveaux scénarios web.

Note:
La création d'un scénario web créera automatiquement un ensemble d'éléments de surveillance web.

Note:
Cette méthode est disponible uniquement pour les types d'utilisateur *Admin* et *Super admin*. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres des rôles utilisateur. Consultez Rôles utilisateur pour plus d'informations.

Paramètres

`(object/array)` Scénarios web à créer.

En plus des propriétés standard des scénarios web, la méthode accepte les paramètres suivants.

Paramètre	Type	Description
steps	array	Étapes du scénario. Comportement du paramètre: - *obligatoire*
tags	array	Tags du scénario web.

Valeurs retournées

`(object)` Retourne un objet contenant les identifiants des scénarios Web créés sous la propriété `httptestids`. L'ordre des ID retournés correspond à l'ordre des scénarios Web transmis.

Exemples

Création d'un scénario web

Créez un scénario web pour surveiller la page d'accueil de l'entreprise. Le scénario comportera deux étapes, pour vérifier la page d'accueil et la page « À propos » et s'assurer qu'elles renvoient le code d'état HTTP 200.

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "httptest.create",
    "params": {
        "name": "Homepage check",
        "hostid": "10085",
        "steps": [
            {
                "name": "Homepage",
                "url": "http://example.com",
                "status_codes": "200",
                "no": 1
            },
            {
                "name": "Homepage / About",
                "url": "http://example.com/about",
                "status_codes": "200",
                "no": 2
            }
        ]
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "httptestids": [
            "5"
        ]
    },
    "id": 1
}
```

Voir aussi

- Étape de scénario

Source

CHttpTest::create() dans *ui/include/classes/api/services/CHttpTest.php*.

httptest.delete

Description

`object httptest.delete(array webScenarioIds)`

Cette méthode permet de supprimer des scénarios web.

> **Note:**
> Cette méthode est disponible uniquement pour les types d'utilisateur *Admin* et *Super admin*. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres du rôle utilisateur. Consultez Rôles d'utilisateur pour plus d'informations.

Paramètres

`(array)` ID des scénarios web à supprimer.

Valeurs de retour

`(object)` Renvoie un objet contenant les ID des scénarios web supprimés dans la propriété `httptestids`.

Exemples

Suppression de plusieurs scénarios web

Supprimez deux scénarios web.

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "httptest.delete",
    "params": [
        "2",
        "3"
    ],
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "httptestids": [
            "2",
            "3"
        ]
    },
    "id": 1
}
```

Source

CHttpTest::delete() dans *ui/include/classes/api/services/CHttpTest.php*.

httptest.get

Description

`integer/array httptest.get(object parameters)`

La méthode permet de récupérer des scénarios web selon les paramètres fournis.

> **Note:**
> Cette méthode est disponible pour les utilisateurs de tout type. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres du rôle utilisateur. Consultez Rôles utilisateur pour plus d'informations.

Paramètres

`(object)` Paramètres définissant la sortie souhaitée.

La méthode prend en charge les paramètres suivants.

Paramètre	Type	Description
groupids	ID/array	Retourner uniquement les scénarios web qui appartiennent aux groupes d'hôtes donnés.
hostids	ID/array	Retourner uniquement les scénarios web qui appartiennent aux hôtes donnés.
httptestids	ID/array	Retourner uniquement les scénarios web avec les ID donnés.
inherited	boolean	Si défini sur `true`, retourner uniquement les scénarios web hérités d'un modèle.
inheritedTags	boolean	Retourner les scénarios web qui ont les `tags` donnés également dans le modèle/l'hôte/les modèles liés. Valeurs possibles : `true` - le modèle/l'hôte/les modèles liés doivent également avoir les tags donnés ; `false` - *(par défaut)* les tags du modèle/de l'hôte/des modèles liés sont ignorés.

Paramètre	Type	Description
monitored	boolean	Si défini sur `true`, retourner uniquement les scénarios web activés qui appartiennent à des hôtes surveillés.
selectInheritedTags	query	Retourner une propriété `inheritedTags` avec les tags présents sur le modèle/l'hôte/les modèles liés.
templated	boolean	Si défini sur `true`, retourner uniquement les scénarios web qui appartiennent à des modèles.
templateids	ID/array	Retourner uniquement les scénarios web qui appartiennent aux modèles donnés.
expandName	flag	Développer les macros dans le nom du scénario web.
expandStepName	flag	Développer les macros dans les noms des étapes du scénario.
evaltype	integer	Méthode d'évaluation des tags. Valeurs possibles : 0 - *(par défaut)* Et/Ou ; 2 - Ou.
tags	array	Retourner uniquement les scénarios web avec les tags donnés. Format : `[{"tag": "<tag>", "value": "<value>", "operator": "<operator>"}, ...]`. Un tableau vide retourne tous les scénarios web. Valeurs possibles de operator : 0 - *(par défaut)* Contient ; 1 - Égal à ; 2 - Ne contient pas ; 3 - N'est pas égal à ; 4 - Existe ; 5 - N'existe pas.
selectHosts	query	Retourner les hôtes auxquels appartient le scénario web sous forme de tableau dans la propriété `hosts`.
selectSteps	query	Retourner les étapes du scénario web dans la propriété `steps`. Prend en charge `count`.
selectTags	query	Retourner les tags du scénario web dans la propriété `tags`.
sortfield	string/array	Trier le résultat selon les propriétés données. Valeurs possibles : `httptestid`, `name`.
countOutput	boolean	Ces paramètres sont décrits dans le commentaire de référence.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Valeurs de retour

`(integer/array)` Renvoie soit :

- un tableau d'objets ;
- le nombre d'objets récupérés, si le paramètre `countOutput` a été utilisé.

Exemples

Récupération d'un scénario web

Récupérez toutes les données du scénario web « 9 ».

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "httptest.get",
    "params": {
        "output": "extend",
        "selectSteps": "extend",
        "httptestids": "9"
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "httptestid": "9",
            "name": "Homepage check",
            "delay": "1m",
            "status": "0",
            "agent": "Zabbix",
            "authentication": "0",
            "http_user": "",
            "http_password": "",
            "hostid": "10084",
            "templateid": "0",
            "http_proxy": "",
            "retries": "1",
            "ssl_cert_file": "",
            "ssl_key_file": "",
            "ssl_key_password": "",
            "verify_peer": "0",
            "verify_host": "0",
            "uuid": "",
            "headers": [],
            "variables": [],
            "steps": [
                {
                    "httpstepid": "36",
                    "httptestid": "9",
                    "name": "Homepage",
                    "no": "1",
                    "url": "http://example.com",
                    "timeout": "15s",
                    "posts": "",
                    "required": "",
                    "status_codes": "200",
                    "follow_redirects": "1",
                    "retrieve_mode": "0",
                    "post_type": "1",
                    "headers": [],
                    "variables": [
                        {
                            "name":"{var}",
                            "value":"12"
                        }
                    ],
                    "query_fields": []
                },
                {
                    "httpstepid": "37",
                    "httptestid": "9",
```

```
                    "name": "Homepage / About",
                    "no": "2",
                    "url": "http://example.com/about",
                    "timeout": "15s",
                    "posts": "",
                    "required": "",
                    "status_codes": "200",
                    "follow_redirects": "1",
                    "retrieve_mode": "0",
                    "post_type": "1",
                    "headers": [],
                    "variables": [],
                    "query_fields": []
                }
            ]
        }
    ],
    "id": 1
}
```


Voir aussi

- Hôte
- Étape de scénario

Source

CHttpTest::get() dans *ui/include/classes/api/services/CHttpTest.php*.

httptest.update

Description

`object httptest.update(object/array webScenarios)`

Cette méthode vous permet de mettre à jour des scénarios web existants.

Note:
Cette méthode est disponible uniquement pour les types d'utilisateur *Admin* et *Super admin*. Les autorisations d'appel de cette méthode peuvent être révoquées dans les paramètres des rôles utilisateur. Consultez Rôles utilisateur pour plus d'informations.

Paramètres

`(object/array)` Propriétés du scénario web à mettre à jour.

La propriété `httptestid` doit être définie pour chaque scénario web, toutes les autres propriétés sont facultatives. Seules les propriétés transmises seront mises à jour, toutes les autres resteront inchangées.

En plus des propriétés standard du scénario web, la méthode accepte les paramètres suivants.

Parameter	Type	Description
steps	array	Étapes du scénario pour remplacer les étapes existantes.
tags	array	Tags du scénario web.

Valeurs retournées

`(object)` Retourne un objet contenant les identifiants des scénarios Web mis à jour sous la propriété `httptestids`.

Exemples

Activation d'un scénario web

Activez un scénario web, c'est-à-dire définissez son statut sur « 0 ».

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "httptest.update",
    "params": {
        "httptestid": "5",
        "status": 0
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "httptestids": [
            "5"
        ]
    },
    "id": 1
}
```

Voir aussi

- Étape de scénario

Source

CHttpTest::update() dans *ui/include/classes/api/services/CHttpTest.php*.

Service

Cette classe est conçue pour fonctionner avec les services métier/l'infrastructure IT.

Références d'objet :

- Service
- Règle de statut
- Tag de service
- Alarme de service
- Tag de problème

Méthodes disponibles :

- service.create - créer de nouveaux services
- service.delete - supprimer des services
- service.get - récupérer des services
- service.update - mettre à jour des services

Objet service

Les objets suivants sont directement liés à l'API `service`.

Service

L'objet service possède les propriétés suivantes.

Property	Type	Description
serviceid	ID	ID du service. Property behavior: - *read-only* - *required* for update operations

Property	Type	Description
algorithm	integer	Règle de calcul de l'état. Applicable uniquement si des services enfants existent. Valeurs possibles : 0 - définir l'état sur OK; 1 - le plus critique si tous les enfants ont des problèmes; 2 - le plus critique des services enfants. Property behavior: - *required* for create operations
name	string	Nom du service. Property behavior: - *required* for create operations
sortorder	integer	Position du service utilisée pour le tri. Valeurs possibles : 0-999. Property behavior: - *required* for create operations
weight	integer	Poids du service. Valeurs possibles : 0-1000000. Par défaut : 0.
propagation_rule	integer	Règle de propagation de l'état. Valeurs possibles : 0 - *(default)* propager l'état du service tel quel - sans aucune modification; 1 - augmenter l'état propagé d'une valeur `propagation_value` donnée (de 1 à 5 niveaux de gravité); 2 - diminuer l'état propagé d'une valeur `propagation_value` donnée (de 1 à 5 niveaux de gravité); 3 - ignorer ce service - l'état n'est pas du tout propagé au service parent; 4 - définir un état de service fixe à l'aide d'une valeur `propagation_value` donnée. Property behavior: - *required* if `propagation_value` is set

Property	Type	Description
propagation_value	integer	Valeur de propagation de l'état. Valeurs possibles si `propagation_rule` est défini sur "0" ou "3" : 0 - Non classé. Valeurs possibles si `propagation_rule` est défini sur "1" ou "2" : 1 - Information; 2 - Avertissement; 3 - Moyen; 4 - Élevé; 5 - Désastre. Valeurs possibles si `propagation_rule` est défini sur "4" : -1 - OK; 0 - Non classé; 1 - Information; 2 - Avertissement; 3 - Moyen; 4 - Élevé; 5 - Désastre. Property behavior: - *required* if `propagation_rule` is set
status	integer	Indique si le service est dans un état OK ou en problème. Si le service est dans un état de problème, `status` est égal soit à la gravité du problème le plus critique, soit à l'état le plus élevé d'un service enfant en état de problème. Si le service est dans un état OK, `status` est égal à : -1. Property behavior: - *read-only*
description	string	Description du service.
uuid	string	Identifiant unique universel, utilisé pour lier les services importés à ceux déjà existants. Généré automatiquement s'il n'est pas fourni.
created_at	integer	Horodatage Unix de la création du service.
readonly	integer	Accès au service. Valeurs possibles : 0 - Lecture-écriture; 1 - Lecture seule. Property behavior: - *read-only*

Règle de statut

L'objet de règle de statut possède les propriétés suivantes.

Propriété	Type	Description
type	integer	Condition pour définir le statut (Nouveau statut). Valeurs possibles : 0 - si au moins (N) services enfants ont le statut (Statut) ou supérieur ; 1 - si au moins (N%) des services enfants ont le statut (Statut) ou supérieur ; 2 - si moins de (N) services enfants ont le statut (Statut) ou inférieur ; 3 - si moins de (N%) des services enfants ont le statut (Statut) ou inférieur ; 4 - si le poids des services enfants ayant le statut (Statut) ou supérieur est au moins égal à (W) ; 5 - si le poids des services enfants ayant le statut (Statut) ou supérieur est au moins égal à (N%) ; 6 - si le poids des services enfants ayant le statut (Statut) ou inférieur est inférieur à (W) ; 7 - si le poids des services enfants ayant le statut (Statut) ou inférieur est inférieur à (N%). Où : - N (W) correspond à `limit_value` ; - (Statut) correspond à `limit_status` ; - (Nouveau statut) correspond à `new_status`. Comportement des propriétés : - *obligatoire*
limit_value	integer	Valeur limite. Valeurs possibles : - pour N et W : 1-100000 ; - pour N% : 1-100. Comportement des propriétés : - *obligatoire*
limit_status	integer	Statut limite. Valeurs possibles : -1 - OK ; 0 - Non classé ; 1 - Information ; 2 - Avertissement ; 3 - Moyen ; 4 - Élevé ; 5 - Désastre. Comportement des propriétés : - *obligatoire*
new_status	integer	Valeur du nouveau statut. Valeurs possibles : 0 - Non classé ; 1 - Information ; 2 - Avertissement ; 3 - Moyen ; 4 - Élevé ; 5 - Désastre. Comportement des propriétés : - *obligatoire*

Balise de service

L'objet de balise de service possède les propriétés suivantes.

Propriété	Type	Description
tag	string	Nom de la balise de service. Comportement de la propriété : - *obligatoire*
value	string	Valeur de la balise de service.

Alarme de service

Note:
Les alarmes de service ne peuvent pas être créées, mises à jour ou supprimées directement via l'API Zabbix.

Les objets d'alarme de service représentent un changement d'état d'un service. Il possède les propriétés suivantes.

Property	Type	Description
clock	timestamp	Heure à laquelle le changement d'état du service s'est produit.
value	integer	Statut du service. Reportez-vous à la propriété `status` du service pour obtenir la liste des valeurs possibles.

Balise de problème

Les balises de problème permettent de lier des services avec des événements de problème. L'objet balise de problème possède les propriétés suivantes.

Property	Type	Description
tag	string	Nom de la balise de problème. Comportement de la propriété: - *required*
operator	integer	Opérateur de la condition de mappage. Valeurs possibles : 0 - *(par défaut)* Égal à; 2 - Contient.
value	string	Valeur de la balise de problème.

service.create

Description

`object service.create(object/array services)`

Cette méthode vous permet de créer de nouveaux services.

Note:
Cette méthode est disponible pour les utilisateurs de tout type. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres des rôles utilisateur. Consultez Rôles utilisateur pour plus d'informations.

Paramètres

`(object/array)` services à créer.

En plus des propriétés de service standard, la méthode accepte les paramètres suivants.

Paramètre	Type	Description
children	array	Services enfants à lier au service. Les services enfants doivent avoir uniquement la propriété `serviceid` définie.
parents	array	Services parents à lier au service. Les services parents doivent avoir uniquement la propriété `serviceid` définie.
tags	array	Tags de service à créer pour le service.
problem_tags	array	Tags de problème à créer pour le service.
status_rules	array	Règles de statut à créer pour le service.

Valeurs retournées

`(object)` Retourne un objet contenant les identifiants des services créés sous la propriété `servicesids`. L'ordre des ID retournés correspond à l'ordre des services transmis.

Exemples

Création d'un service

Créez un service qui passera à l'état de problème si au moins un enfant a un problème.

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "service.create",
    "params": {
        "name": "Server 1",
        "algorithm": 1,
        "sortorder": 1
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "serviceids": [
            "5"
        ]
    },
    "id": 1
}
```

Source

CService::create() dans *ui/include/classes/api/services/CService.php*.

service.delete

Description

`object service.delete(array serviceIds)`

Cette méthode vous permet de supprimer des services.

> **Note:**
> Cette méthode est disponible pour les utilisateurs de tout type. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres des rôles d'utilisateur. Consultez Rôles d'utilisateur pour plus d'informations.

Paramètres

(array) ID des services à supprimer.

Valeurs de retour

(object) Renvoie un objet contenant les ID des services supprimés dans la propriété serviceids.

Exemples

Suppression de plusieurs services

Supprimez deux services.

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "service.delete",
    "params": [
        "4",
        "5"
    ],
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "serviceids": [
            "4",
            "5"
        ]
    },
    "id": 1
}
```

Source

CService::delete() dans *ui/include/classes/api/services/CService.php*.

service.get

Description

integer/array service.get(object parameters)

La méthode permet de récupérer des services selon les paramètres fournis.

Note:
Cette méthode est disponible pour les utilisateurs de tout type. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres du rôle utilisateur. Consultez Rôles d'utilisateur pour plus d'informations.

Paramètres

(object) Paramètres définissant la sortie souhaitée.

La méthode prend en charge les paramètres suivants.

Paramètre	Type	Description
serviceids	ID/array	Retourner uniquement les services avec les ID donnés.
parentids	ID/array	Retourner uniquement les services liés aux services parents donnés.
deep_parentids	flag	Retourner tous les services enfants directs et indirects. Utilisé avec parentids.
childids	ID/array	Retourner uniquement les services liés aux services enfants donnés.

Paramètre	Type	Description
evaltype	integer	Méthode d'évaluation des tags. Valeurs possibles : 0 - *(par défaut)* Et/Ou ; 2 - Ou.
tags	object/array	Retourner uniquement les services avec les tags donnés. Format : `[{"tag": "<tag>", "value": "<value>", "operator": "<operator>"}, ...]`. Un tableau vide retourne tous les services. Valeurs possibles de l'operator : 0 - *(par défaut)* Contient ; 1 - Égal ; 2 - Ne contient pas ; 3 - N'est pas égal ; 4 - Existe ; 5 - N'existe pas.
problem_tags	object/array	Retourner uniquement les services avec les tags de problème donnés. Format : `[{"tag": "<tag>", "value": "<value>", "operator": "<operator>"}, ...]`. Un tableau vide retourne tous les services. Valeurs possibles de l'operator : 0 - *(par défaut)* Contient ; 1 - Égal ; 2 - Ne contient pas ; 3 - N'est pas égal ; 4 - Existe ; 5 - N'existe pas.
without_problem_tags	flag	Retourner uniquement les services sans tags de problème.
slaids	ID/array	Retourner uniquement les services liés au(x) SLA spécifié(s).
selectChildren	query	Retourner une propriété `children` avec les services enfants. Prend en charge `count`.
selectParents	query	Retourner une propriété `parents` avec les services parents. Prend en charge `count`.
selectTags	query	Retourner une propriété `tags` avec les tags du service. Prend en charge `count`.
selectProblemEvents	query	Retourner une propriété `problem_events` avec un tableau d'objets d'événement de problème. L'objet d'événement de problème possède les propriétés suivantes : `eventid` - `(ID)` ID de l'événement ; `severity` - `(string)` Sévérité actuelle de l'événement ; `name` - `(string)` Nom de l'événement résolu. Prend en charge `count`.
selectProblemTags	query	Retourner une propriété `problem_tags` avec les tags de problème. Prend en charge `count`.
selectStatusRules	query	Retourner une propriété `status_rules` avec les règles de statut. Prend en charge `count`.

Paramètre	Type	Description
selectStatusTimeline	object/array	Retourner une propriété `status_timeline` contenant les changements d'état du service pour les périodes données. Format `[{"period_from": "<period_from>", "period_to": "<period_to>"}, ...]` - `period_from` étant une date de début (incluse ; horodatage entier) et `period_to` une date de fin (exclue ; horodatage entier) pour la période qui vous intéresse. Retourne un tableau d'entrées contenant une propriété `start_value` et un tableau `alarms` pour les changements d'état au sein des périodes spécifiées.
sortfield	string/array	Trier le résultat selon les propriétés données. Valeurs possibles : `serviceid`, `name`, `status`, `sortorder`, `created_at`.
countOutput	boolean	Ces paramètres sont décrits dans le commentaire de référence.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Valeurs retournées

`(integer/array)` Retourne soit :

- un tableau d'objets ;
- le nombre d'objets récupérés, si le paramètre `countOutput` a été utilisé.

Exemples

Récupération de tous les services

Récupérez toutes les données concernant tous les services et leurs dépendances.

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "service.get",
    "params": {
        "output": "extend",
        "selectChildren": "extend",
        "selectParents": ["serviceid", "name"]
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "serviceid": "1",
            "name": "Zabbix cluster",
            "status": "-1",
            "algorithm": "2",
            "sortorder": "0",
```

```
            "weight": "0",
            "propagation_rule": "0",
            "propagation_value": "0",
            "description": "",
            "uuid": "8d77bd91b62347e4b79382912eb5df95",
            "created_at": "1761742392",
            "readonly": false,
            "parents": [],
            "children": [
                {
                    "serviceid": "2",
                    "name": "Zabbix server node 1",
                    "status": "-1",
                    "algorithm": "2",
                    "sortorder": "0",
                    "weight": "0",
                    "propagation_rule": "0",
                    "propagation_value": "0",
                    "description": "",
                    "uuid": "195911d26d7f4e218d6217079bcd5929",
                    "created_at": "1761742623",
                    "readonly": false
                },
                {
                    "serviceid": "3",
                    "name": "Zabbix server node 2",
                    "status": "-1",
                    "algorithm": "2",
                    "sortorder": "0",
                    "weight": "0",
                    "propagation_rule": "0",
                    "propagation_value": "0",
                    "description": "",
                    "uuid": "9fc659a30fe244f690dff25fc2a9db5c",
                    "created_at": "1761742654",
                    "readonly": false
                }
            ]
        },
        {
            "serviceid": "2",
            "name": "Zabbix server node 1",
            "status": "-1",
            "algorithm": "2",
            "sortorder": "0",
            "weight": "0",
            "propagation_rule": "0",
            "propagation_value": "0",
            "description": "",
            "uuid": "195911d26d7f4e218d6217079bcd5929",
            "created_at": "1761742623",
            "readonly": false,
            "parents": [
                {
                    "serviceid": "1",
                    "name": "Zabbix cluster"
                }
            ],
            "children": []
        },
        {
            "serviceid": "3",
```

```
            "name": "Zabbix server node 2",
            "status": "-1",
            "algorithm": "2",
            "sortorder": "0",
            "weight": "0",
            "propagation_rule": "0",
            "propagation_value": "0",
            "description": "",
            "uuid": "9fc659a30fe244f690dff25fc2a9db5c",
            "created_at": "1761742654",
            "readonly": false,
            "parents": [
                {
                    "serviceid": "1",
                    "name": "Zabbix cluster"
                }
            ],
            "children": []
        }
    ],
    "id": 1
}
```


Source

CService::get() dans *ui/include/classes/api/services/CService.php*.

service.update

Description

`object service.update(object/array services)`

Cette méthode vous permet de mettre à jour des services existants.

> **Note:**
> Cette méthode est disponible pour les utilisateurs de tout type. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres des rôles utilisateur. Consultez Rôles utilisateur pour plus d'informations.

Paramètres

`(object/array)` propriétés du service à mettre à jour.

La propriété `serviceid` doit être définie pour chaque service, toutes les autres propriétés sont facultatives. Seules les propriétés transmises seront mises à jour, toutes les autres resteront inchangées.

En plus des propriétés de service standard, la méthode accepte les paramètres suivants.

Paramètre	Type	Description
children	array	Services enfants remplaçant les services enfants actuels. Seule la propriété `serviceid` doit être définie pour les services enfants.
parents	array	Services parents remplaçant les services parents actuels. Seule la propriété `serviceid` doit être définie pour les services parents.
tags	array	Tags de service remplaçant les tags de service actuels.
problem_tags	array	Tags de problème remplaçant les tags de problème actuels.
status_rules	array	Règles de statut remplaçant les règles de statut actuelles.

Valeurs retournées

`(object)` Retourne un objet contenant les identifiants des services mis à jour sous la propriété `serviceids`.

Exemples

Définir le parent d'un service

Définissez le service avec l'ID "3" comme parent du service avec l'ID "5".

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "service.update",
    "params": {
        "serviceid": "5",
        "parents": [
            {
                "serviceid": "3"
            }
        ]
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "serviceids": [
            "5"
        ]
    },
    "id": 1
}
```

Ajout d'un temps d'arrêt planifié

Ajoutez un temps d'arrêt pour le service avec l'ID « 4 », planifié chaque semaine du lundi 22:00 au mardi 10:00.

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "service.update",
    "params": {
        "serviceid": "4",
        "times": [
            {
                "type": "1",
                "ts_from": "165600",
                "ts_to": "201600"
            }
        ]
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "serviceids": [
            "4"
        ]
    },
    "id": 1
}
```

Source

CService::update() dans *ui/include/classes/api/services/CService.php*.

Service découvert

Cette classe est conçue pour fonctionner avec les services découverts.

Références d'objet :

- Service découvert

Méthodes disponibles :

- dservice.get - récupérer les services découverts

Objet de service découvert

Les objets suivants sont directement liés à l'API `dservice`.

Service découvert

Note:
Les services découverts sont créés par le serveur Zabbix et ne peuvent pas être modifiés via l'API.

L'objet service découvert contient des informations sur un service découvert par une règle de découverte réseau sur un hôte. Il possède les propriétés suivantes.

Propriété	Type	Description
dserviceid	ID	ID du service découvert.
dcheckid	ID	ID de la vérification de découverte utilisée pour détecter le service.
dhostid	ID	ID de l'hôte découvert exécutant le service.
dns	string	DNS de l'hôte exécutant le service.
ip	string	Adresse IP de l'hôte exécutant le service.
lastdown	timestamp	Heure à laquelle le service découvert est tombé en panne pour la dernière fois.
lastup	timestamp	Heure à laquelle le service découvert est redevenu disponible pour la dernière fois.
port	integer	Numéro de port du service.
status	integer	Statut du service. Valeurs possibles : 0 - service disponible ; 1 - service indisponible.
value	string	Valeur renvoyée par le service lors de l'exécution d'une vérification de découverte Zabbix agent, SNMPv1, SNMPv2 ou SNMPv3.

dservice.get

Description

`integer/array dservice.get(object parameters)`

La méthode permet de récupérer les services découverts selon les paramètres fournis.

Note:
Cette méthode est disponible pour les utilisateurs de tout type. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres des rôles utilisateur. Consultez Rôles utilisateur pour plus d'informations.

Paramètres

`(object)` Paramètres définissant la sortie souhaitée.

La méthode prend en charge les paramètres suivants.

Paramètre	Type	Description
dserviceids	ID/array	Retourner uniquement les services découverts avec les ID indiqués.
dhostids	ID/array	Retourner uniquement les services découverts qui appartiennent aux hôtes découverts indiqués.
dcheckids	ID/array	Retourner uniquement les services découverts qui ont été détectés par les vérifications de découverte indiquées.
druleids	ID/array	Retourner uniquement les services découverts qui ont été détectés par les règles de découverte indiquées.
selectDRules	query	Retourner une propriété `drules` contenant un tableau des règles de découverte qui ont détecté le service.
selectDHosts	query	Retourner une propriété `dhosts` contenant un tableau des hôtes découverts auxquels le service appartient.
selectHosts	query	Retourner une propriété `hosts` contenant les hôtes ayant la même adresse IP et le même proxy que le service. Prend en charge `count`.
limitSelects	integer	Limite le nombre d'enregistrements renvoyés par les sous-sélections. S'applique aux sous-sélections suivantes : `selectHosts` - le résultat sera trié par `hostid`.
sortfield	string/array	Trier le résultat selon les propriétés indiquées. Valeurs possibles : `dserviceid`, `dhostid`, `ip`.
countOutput	boolean	Ces paramètres sont décrits dans le commentaire de référence.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Valeurs retournées

`(integer/array)` Retourne soit :

- un tableau d'objets ;
- le nombre d'objets récupérés, si le paramètre `countOutput` a été utilisé.

Exemples

Récupérer les services découverts sur un hôte

Récupérez tous les services découverts détectés sur l'hôte découvert « 11 ».

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "dservice.get",
    "params": {
        "output": "extend",
        "dhostids": "11"
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": [
```

```
        {
            "dserviceid": "12",
            "dhostid": "11",
            "value": "",
            "port": "80",
            "status": "1",
            "lastup": "0",
            "lastdown": "1348650607",
            "dcheckid": "5",
            "ip": "192.168.1.134",
            "dns": "john.local"
        },
        {
            "dserviceid": "13",
            "dhostid": "11",
            "value": "",
            "port": "21",
            "status": "1",
            "lastup": "0",
            "lastdown": "1348650610",
            "dcheckid": "6",
            "ip": "192.168.1.134",
            "dns": "john.local"
        }
    ],
    "id": 1
}
```


Voir aussi

- Hôte découvert
- Vérification de découverte
- Hôte

Source

CDService::get() dans *ui/include/classes/api/services/CDService.php*.

SLA

Cette classe est conçue pour fonctionner avec des objets SLA (Service Level Agreement) utilisés pour évaluer les performances de l'infrastructure informatique et des services métier.

Références d'objet :

- SLA
- Planification SLA
- Temps d'arrêt exclu du SLA
- Balise de service SLA

Méthodes disponibles :

- sla.create - créer de nouveaux SLA
- sla.delete - supprimer des SLA
- sla.get - récupérer des SLA
- sla.getsli - récupérer les données de Service Level Indicator (SLI) pour les SLA
- sla.update - mettre à jour des SLA

Objet SLA

Les objets suivants sont directement liés à l'API `sla` (Service Level Agreement).

SLA

L'objet SLA possède les propriétés suivantes.

Propriété	Type	Description
slaid	ID	ID du SLA. Comportement de la propriété : - *lecture seule* - *obligatoire* pour les opérations de mise à jour
name	string	Nom du SLA. Comportement de la propriété : - *obligatoire* pour les opérations de création
period	integer	Période de rapport du SLA. Valeurs possibles : 0 - quotidien ; 1 - hebdomadaire ; 2 - mensuel ; 3 - trimestriel ; 4 - annuel. Comportement de la propriété : - *obligatoire* pour les opérations de création
slo	float	Objectif de niveau de service minimum acceptable, exprimé en pourcentage. Si l'indicateur de niveau de service (SLI) descend en dessous, le SLA est considéré comme étant en état de problème/non respecté. Valeurs possibles : 0-100 (jusqu'à 4 chiffres après la virgule). Comportement de la propriété : - *obligatoire* pour les opérations de création
effective_date	integer	Date d'entrée en vigueur du SLA. Valeurs possibles : horodatage de date en UTC.
timezone	string	Fuseau horaire de rapport, par exemple : `Europe/London`, `UTC`. Pour la liste complète des fuseaux horaires pris en charge, veuillez consulter la documentation PHP. Comportement de la propriété : - *obligatoire* pour les opérations de création
status	integer	Statut du SLA. Valeurs possibles : 0 - *(par défaut)* SLA désactivé ; 1 - SLA activé.
description	string	Description du SLA.

Planification du SLA

L'objet de planification du SLA définit les périodes pendant lesquelles le ou les services connectés sont censés être en état de fonctionnement. Il possède les propriétés suivantes.

Propriété	Type	Description
period_from	integer	Heure de début de la période hebdomadaire récurrente (incluse). Valeurs possibles : nombre de secondes (à compter de dimanche). Comportement de la propriété : - *obligatoire*

Propriété	Type	Description
period_to	integer	Heure de fin de la période hebdomadaire récurrente (exclusive). Valeurs possibles : nombre de secondes (à compter de dimanche). Comportement de la propriété : - *obligatoire*

Temps d'arrêt exclus du SLA

L'objet de temps d'arrêt exclu définit des périodes pendant lesquelles le ou les services connectés sont censés être hors service, sans affecter le SLI, par exemple lors d'une maintenance planifiée. Il possède les propriétés suivantes.

Propriété	Type	Description
name	string	Nom de la période d'indisponibilité exclue. Comportement de la propriété: - *obligatoire*
period_from	integer	Heure de début de la période d'indisponibilité exclue (incluse). Valeurs possibles : horodatage. Comportement de la propriété: - *obligatoire*
period_to	integer	Heure de fin de la période d'indisponibilité exclue (exclusive). Valeurs possibles : horodatage. Comportement de la propriété: - *obligatoire*

Balise de service SLA

L'objet de balise de service SLA relie les services à inclure dans les calculs du SLA. Il possède les propriétés suivantes.

Propriété	Type	Description
tag	string	Nom du tag de service SLA. Comportement de la propriété: - *obligatoire*
operator	integer	Opérateur du tag de service SLA. Valeurs possibles : 0 - *(par défaut)* Égal à ; 2 - Contient.
value	string	Valeur du tag de service SLA.

sla.create

Description

`object sla.create(object/array SLAs)`

Cette méthode vous permet de créer de nouveaux objets SLA.

Note:
Cette méthode est disponible uniquement pour les types d'utilisateur *Admin* et *Super admin*. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres des rôles utilisateur. Consultez Rôles utilisateur pour plus d'informations.

Paramètres

`(object/array)` Objets SLA à créer.

En plus des propriétés SLA standard, la méthode accepte les paramètres suivants.

Paramètre	Type	Description
service_tags	array	Tags de service SLA à créer pour le SLA. Comportement du paramètre: - *obligatoire*
schedule	array	Planification SLA à créer pour le SLA. La spécification d'un paramètre vide sera interprétée comme une planification 24x7. Par défaut : planification 24x7.
excluded_downtimes	array	Périodes d'arrêt exclues du SLA à créer pour le SLA.

Valeurs de retour

`(object)` Renvoie un objet contenant les ID des SLA créés dans la propriété `slaids`. L'ordre des ID renvoyés correspond à l'ordre des SLA transmis.

Exemples

Création d'un SLA

Indiquez de créer une entrée de SLA pour : * le suivi de la disponibilité des services liés au moteur SQL ; * un calendrier personnalisé couvrant tous les jours ouvrables, à l'exclusion de la dernière heure du samedi ; * une date d'effet fixée au dernier jour de l'année 2022 ; * un temps d'arrêt planifié d'une durée de 1 heure et 15 minutes, commençant à minuit le 4 juillet ; * l'activation du calcul du rapport hebdomadaire du SLA ; * un SLO minimum acceptable de 99,9995 %.

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "sla.create",
    "params": [
        {
            "name": "Database Uptime",
            "slo": "99.9995",
            "period": "1",
            "timezone": "America/Toronto",
            "description": "Assurer une excellente disponibilité pour les principaux moteurs de base de do
            "effective_date": 1672444800,
            "status": 1,
            "schedule": [
                {
                    "period_from": 0,
                    "period_to": 601200
                }
            ],
            "service_tags": [
                {
                    "tag": "database",
                    "operator": "0",
                    "value": "mysql"
                },
                {
                    "tag": "database",
                    "operator": "0",
                    "value": "postgresql"
                }
            ],
            "excluded_downtimes": [
                {
                    "name": "Déploiement de la mise à niveau de la version du logiciel",
```

```
                    "period_from": "1648760400",
                    "period_to": "1648764900"
                }
            ]
        }
    ],
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "slaids": [
            "5"
        ]
    },
    "id": 1
}
```

Source

CSla::create() dans *ui/include/classes/api/services/CSla.php*.

sla.delete

Description

`object sla.delete(array slaids)`

Cette méthode vous permet de supprimer des entrées SLA.

> **Note:**
> Cette méthode est disponible uniquement pour les types d'utilisateur *Admin* et *Super admin*. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres du rôle utilisateur. Consultez Rôles d'utilisateur pour plus d'informations.

Paramètres

`(array)` ID des SLA à supprimer.

Valeurs de retour

`(object)` Renvoie un objet contenant les ID des SLA supprimés dans la propriété `slaids`.

Exemples

Suppression de plusieurs SLA

Supprimez deux entrées de SLA.

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "sla.delete",
    "params": [
        "4",
        "5"
    ],
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
```

```
        "slaids": [
            "4",
            "5"
        ]
    },
    "id": 1
}
```

Source

CSla::delete() dans *ui/include/classes/api/services/CSla.php*.

sla.get

Description

`integer/array sla.get(object parameters)`

La méthode permet de récupérer des objets SLA selon les paramètres fournis.

> **Note:**
> Cette méthode est disponible pour les utilisateurs de tout type. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres du rôle utilisateur. Consultez Rôles utilisateur pour plus d'informations.

Paramètres

`(object)` Paramètres définissant la sortie souhaitée.

La méthode prend en charge les paramètres suivants.

Paramètre	Type	Description
slaids	ID/array	Renvoie uniquement les SLA avec les ID donnés.
serviceids	ID/array	Renvoie uniquement les SLA correspondant aux services spécifiés.
selectSchedule	query	Renvoie une propriété `schedule` avec les planifications SLA.
selectExcludedDowntimes	query	Prend en charge `count`. Renvoie une propriété `excluded_downtimes` avec les temps d'arrêt exclus du SLA.
selectServiceTags	query	Prend en charge `count`. Renvoie une propriété `service_tags` avec les tags de service du SLA.
sortfield	string/array	Prend en charge `count`. Trie le résultat selon les propriétés données. Valeurs possibles : `slaid`, `name`, `period`, `slo`, `effective_date`, `timezone`, `status`, `description`.
countOutput	boolean	Ces paramètres sont décrits dans le commentaire de référence.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Valeurs de retour

`(integer/array)` Renvoie soit :

- un tableau d'objets ;
- le nombre d'objets récupérés, si le paramètre `countOutput` a été utilisé.

Exemples

Récupération de tous les SLA

Récupérez toutes les données sur tous les SLA et leurs propriétés.

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "sla.get",
    "params": {
        "output": "extend",
        "selectSchedule": ["period_from", "period_to"],
        "selectExcludedDowntimes": ["name", "period_from", "period_to"],
        "selectServiceTags": ["tag", "operator", "value"],
        "preservekeys": true
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "1": {
            "slaid": "1",
            "name": "Database Uptime",
            "period": "1",
            "slo": "99.9995",
            "effective_date": "1672444800",
            "timezone": "America/Toronto",
            "status": "1",
            "description": "Provide excellent uptime for main SQL database engines.",
            "service_tags": [
                {
                    "tag": "database",
                    "operator": "0",
                    "value": "mysql"
                },
                {
                    "tag": "database",
                    "operator": "0",
                    "value": "postgresql"
                }
            ],
            "schedule": [
                {
                    "period_from": "0",
                    "period_to": "601200"
                }
            ],
            "excluded_downtimes": [
                {
                    "name": "Software version upgrade rollout",
                    "period_from": "1648760400",
                    "period_to": "1648764900"
                }
            ]
        }
    },
    "id": 1
```

```
}
```

Source

CSla:get() dans *ui/include/classes/api/services/CSla.php*.

sla.getsli

Description

`object sla.getsli(object parameters)`

Cette méthode vous permet de calculer les données de l'indicateur de niveau de service (SLI) pour un accord de niveau de service (SLA).

Note:
Cette méthode est disponible pour les utilisateurs de tout type. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres des rôles utilisateur. Consultez User roles pour plus d'informations.

Paramètres

`(object)` Paramètres contenant l'ID du SLA, les périodes de rapport et, éventuellement, les ID des services - pour lesquels calculer le SLI.

Paramètre	Type	Description
slaid	ID	ID du SLA pour lequel retourner les informations de disponibilité. Comportement du paramètre: - *obligatoire*
period_from	timestamp	Horodatage de début (inclus) pour lequel rapporter le SLI. Valeurs possibles : horodatage Unix.
period_to	timestamp	Horodatage de fin (inclus) pour lequel rapporter le SLI. Valeurs possibles : horodatage Unix.
periods	integer	Nombre de périodes à rapporter. Valeurs possibles : 1-100
serviceids	ID/array	ID des services pour lesquels retourner le SLI.

Partitionnement des périodes

Le tableau suivant illustre l'organisation des tranches de périodes renvoyées selon les combinaisons de paramètres.

Note:
Les périodes renvoyées ne précèdent pas la première période disponible (en fonction de la date d'effet du SLA) et ne dépassent pas la période en cours.

Paramètres			Périodes renvoyées
period_from	**period_to**	**periods**	
-	-	-	20 dernières périodes, y compris la période en cours.
-	-	N	N dernières périodes.
-	specified	-	20 dernières périodes avant `period_to`.
-	specified	N	N dernières périodes avant `period_to`.
specified	-	-	20 premières périodes à partir de `period_from`.
specified	-	N	N premières périodes à partir de `period_from`.
specified	specified	-	Jusqu'à 100 périodes dans la plage spécifiée.
specified	specified	N	N périodes dans la plage spécifiée.

Valeurs de retour

`(object)` Renvoie les résultats du calcul.

Propriété	Type	Description
periods	array	Liste des périodes rapportées. Chaque période rapportée est représentée comme un objet composé de : - `period_from` - (`timestamp`) Horodatage de début (inclusif) de la période rapportée. - `period_to` - (`timestamp`) Horodatage de fin (exclusif) de la période rapportée. Les périodes sont triées par `period_from`, les périodes les plus anciennes apparaissant en premier.
serviceids	array	Liste des ID de service dans les périodes rapportées. L'ordre de tri de la liste n'est pas défini. Même si le paramètre `serviceids` a été transmis à la méthode `sla.getsli`.
sli	array	Données SLI (sous la forme d'un **tableau à deux dimensions**) pour chaque période rapportée et service. L'index de la propriété `periods` est utilisé comme **première** dimension de la propriété `sli`. L'index de la propriété `serviceids` est utilisé comme **deuxième** dimension de la propriété `sli`.

Données SLI

Les données SLI renvoyées pour chaque période signalée et service comprennent :

Propriété	Type	Description
uptime	integer	Durée pendant laquelle le service est resté dans un état *OK* pendant le temps de disponibilité planifié, moins les temps d'arrêt exclus.
downtime	integer	Durée pendant laquelle le service est resté dans un état *not OK* pendant le temps de disponibilité planifié, moins les temps d'arrêt exclus.
sli	float	SLI (pourcentage du temps de disponibilité total), basé sur la disponibilité et l'indisponibilité.
error_budget	integer	Budget d'erreur (en secondes), basé sur le SLI et le SLO.
excluded_downtimes	array	Tableau des temps d'arrêt exclus dans cette période de rapport. Chaque objet contiendra les paramètres suivants : - `name` - (`string`) Nom du temps d'arrêt exclu. - `period_from` - (`timestamp`) Horodatage de début (inclusif) du temps d'arrêt exclu. - `period_to` - (`timestamp`) Horodatage de fin (exclusif) du temps d'arrêt exclu. Les temps d'arrêt exclus sont triés par `period_from`, les périodes les plus anciennes apparaissant en premier.

Exemples

Calcul du SLI pour un SLA quotidien

Récupérez les données SLI pour les services ayant les ID « 1 » et « 4 » qui sont liés au SLA ayant l'ID « 1 ». Récupérez les données pour une seule période jusqu'à « 1761861599 » (30 oct. 2025 23:59:59 GMT+0200). Étant donné que la période de rapport du SLA est quotidienne, les données SLI sont récupérées de « 1761775200 » (30 oct. 2025 00:00:00 GMT+0200) à « 1761861600 » (31 oct. 2025 00:00:00 GMT+0200).

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "sla.getsli",
    "params": {
        "slaid": "1",
        "serviceids": [
            1,
            4
        ],
        "periods": 1,
        "period_to": 1761861599
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "periods": [
            {
                "period_from": 1761775200,
                "period_to": 1761861600
            }
        ],
        "serviceids": [
            1,
            4
        ],
        "sli": [
            [
                {
                    "uptime": 43843,
                    "downtime": 0,
                    "sli": 100,
                    "error_budget": 0,
                    "excluded_downtimes": [
                        {
                            "name": "Maintenance OCT",
                            "period_from": 1761825600,
                            "period_to": 1761829200
                        }
                    ]
                },
                {
                    "uptime": 32225,
                    "downtime": 0,
                    "sli": 100,
                    "error_budget": 0,
                    "excluded_downtimes": []
                }
            ]
        ]
    },
    "id": 1
}
```

Calcul du SLI pour un SLA mensuel

Récupérez les données SLI des services ayant les ID ”50”, ”60” et ”70” qui sont liés au SLA ayant l’ID ”5”. Récupérez les données pour trois périodes à partir de ”1635724800” (01 nov. 2021 00:00:00 UTC). Comme la période de rapport du SLA est mensuelle, les données SLI sont récupérées pour les trois mois suivants :

- De ”1635724800” (01 nov. 2021 00:00:00 UTC) à ”1638316800” (01 déc. 2021 00:00:00 UTC)

- De ”1638316800” (01 déc. 2021 00:00:00 UTC) à ”1640995200” (01 janv. 2022 00:00:00 UTC)
- De ”1640995200” (01 janv. 2022 00:00:00 UTC) à ”1643673600” (01 févr. 2022 00:00:00 UTC)

Requête :


```
{
    "jsonrpc": "2.0",
    "method": "sla.getsli",
    "params": {
        "slaid": "5",
        "serviceids": [
            50,
            60,
            70
        ],
        "periods": 3,
        "period_from": 1635724800
    },
    "id": 1
}
```


Réponse :


```
{
    "jsonrpc": "2.0",
    "result": {
        "periods": [
            {
                "period_from": 1635724800,
                "period_to": 1638316800
            },
            {
                "period_from": 1638316800,
                "period_to": 1640995200
            },
            {
                "period_from": 1640995200,
                "period_to": 1643673600
            }
        ],
        "serviceids": [
            50,
            60,
            70
        ],
        "sli": [
            [
                {
                    "uptime": 1186212,
                    "downtime": 0,
                    "sli": 100,
                    "error_budget": 0,
                    "excluded_downtimes": [
                        {
                            "name": "Maintenance Nov 25 - Dec 01",
                            "period_from": 1637836212,
                            "period_to": 1638316800
                        }
                    ]
                },
                {
                    "uptime": 1186212,
                    "downtime": 0,
                    "sli": 100,
                    "error_budget": 0,
```

```
                "excluded_downtimes": [
                    {
                        "name": "Maintenance Nov 25 - Dec 01",
                        "period_from": 1637836212,
                        "period_to": 1638316800
                    }
                ]
            },
            {
                "uptime": 1186212,
                "downtime": 0,
                "sli": 100,
                "error_budget": 0,
                "excluded_downtimes": [
                    {
                        "name": "Maintenance Nov 25 - Dec 01",
                        "period_from": 1637836212,
                        "period_to": 1638316800
                    }
                ]
            }
        ],
        [
            {
                "uptime": 1147548,
                "downtime": 0,
                "sli": 100,
                "error_budget": 0,
                "excluded_downtimes": [
                    {
                        "name": "Maintenance Dec 02 - Dec 10",
                        "period_from": 1638439200,
                        "period_to": 1639109652
                    }
                ]
            },
            {
                "uptime": 1147548,
                "downtime": 0,
                "sli": 100,
                "error_budget": 0,
                "excluded_downtimes": [
                    {
                        "name": "Maintenance Dec 02 - Dec 10",
                        "period_from": 1638439200,
                        "period_to": 1639109652
                    }
                ]
            },
            {
                "uptime": 1147548,
                "downtime": 0,
                "sli": 100,
                "error_budget": 0,
                "excluded_downtimes": [
                    {
                        "name": "Maintenance Dec 02 - Dec 10",
                        "period_from": 1638439200,
                        "period_to": 1639109652
                    }
                ]
            }
```

```
            ],
            [
                {
                    "uptime": 1674000,
                    "downtime": 0,
                    "sli": 100,
                    "error_budget": 0,
                    "excluded_downtimes": []
                },
                {
                    "uptime": 1674000,
                    "downtime": 0,
                    "sli": 100,
                    "error_budget": 0,
                    "excluded_downtimes": []
                },
                {
                    "uptime": 1674000,
                    "downtime": 0,
                    "sli": 100,
                    "error_budget": 0,
                    "excluded_downtimes": []
                }
            ]
        ]
    },
    "id": 1
}
```


Source

CSla::getSli() dans *ui/include/classes/api/services/CSla.php*

sla.update

Description

`object sla.update(object/array slaids)`

Cette méthode vous permet de mettre à jour des entrées SLA existantes.

> **Note:**
> Cette méthode est disponible uniquement pour les types d'utilisateur *Admin* et *Super admin*. Les autorisations d'appel de cette méthode peuvent être révoquées dans les paramètres des rôles utilisateur. Consultez Rôles utilisateur pour plus d'informations.

Paramètres

`(object/array)` Propriétés de SLA à mettre à jour.

La propriété `slaid` doit être définie pour chaque SLA, toutes les autres propriétés sont facultatives. Seules les propriétés transmises seront mises à jour, toutes les autres resteront inchangées.

En plus des propriétés SLA standard, la méthode accepte les paramètres suivants.

Paramètre	Type	Description
service_tags	array	Balises de service SLA pour remplacer les balises de service SLA actuelles.
schedule	array	Planification SLA pour remplacer la planification actuelle. Spécifier le paramètre comme vide sera interprété comme une planification 24x7.
excluded_downtimes	array	Périodes d'indisponibilité exclues de SLA pour remplacer celles actuellement définies.

Valeurs de retour

`(object)` Renvoie un objet contenant les identifiants des SLA mis à jour dans la propriété `slaids`.

Exemples

Mise à jour des tags de service

Faites en sorte que le SLA avec l'ID « 5 » soit calculé à des intervalles mensuels pour les services liés à NoSQL, sans modifier sa planification ni les temps d'arrêt exclus ; définissez le SLO à 95 %.

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "sla.update",
    "params": [
        {
            "slaid": "5",
            "name": "NoSQL Database engines",
            "slo": "95",
            "period": 2,
            "service_tags": [
                {
                    "tag": "database",
                    "operator": "0",
                    "value": "redis"
                },
                {
                    "tag": "database",
                    "operator": "0",
                    "value": "mongodb"
                }
            ]
        }
    ],
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "slaids": [
            "5"
        ]
    },
    "id": 1
}
```

Modification de la planification d'un SLA

Basculez le SLA avec l'ID « 5 » vers une planification 24x7.

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "sla.update",
    "params": {
        "slaid": "5",
        "schedule": []
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "slaids": [
            "5"
        ]
    },
    "id": 1
}
```

Modification des périodes d'indisponibilité exclues pour un SLA

Ajoutez une période d'indisponibilité planifiée de 4 heures pour une mise à niveau de la RAM le 6 avril 2022, tout en conservant (elle doit être redéfinie) une mise à niveau logicielle planifiée existante auparavant le 4 juillet pour le SLA avec l'ID « 5 ».

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "sla.update",
    "params": {
        "slaid": "5",
        "excluded_downtimes": [
            {
                "name": "Software version upgrade rollout",
                "period_from": "1648760400",
                "period_to": "1648764900"
            },
            {
                "name": "RAM upgrade",
                "period_from": "1649192400",
                "period_to": "1649206800"
            }
        ]
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "slaids": [
            "5"
        ]
    },
    "id": 1
}
```

Source

CSla::update() dans *ui/include/classes/api/services/CSla.php*.

Span APM

Cette classe est conçue pour fonctionner avec les spans collectés pour la surveillance des performances des applications (APM).

Références d'objet :

- Span APM

Méthodes disponibles :

- apmspan.get - récupérer des spans

Objet span APM

Les objets suivants sont directement liés à l'API `apmspan`.

> **Note:**
> Chaque propriété correspond à une colonne de la table `otel_traces` de la source de données de télémétrie. Pour obtenir la liste des colonnes, consultez Colonnes de données de télémétrie prises en charge.

Span APM

L'objet span APM représente un span (une opération unique au sein d'une trace, telle qu'une requête HTTP, une requête de base de données ou un appel de fonction).

Les spans forment une hiérarchie qui commence par le span racine, représentant la première opération de la requête. Chaque autre span est un span enfant du span pendant lequel son opération a été démarrée.

L'objet span APM possède les propriétés suivantes :

Property	Type	Description
timestamp	integer	Heure Unix (en nanosecondes) à laquelle le span a démarré. Comportement de la propriété : - *lecture seule*
traceid	string	ID de la trace à laquelle appartient le span. Comportement de la propriété : - *lecture seule*
spanid	string	ID du span. Comportement de la propriété : - *lecture seule*
parent_spanid	string	ID du span parent. Vide pour le span racine. Comportement de la propriété : - *lecture seule*
trace_state	string	Données supplémentaires, le cas échéant, que les outils de traçage associent à la trace (voir W3C Trace Context). Comportement de la propriété : - *lecture seule*
span_name	string	Nom de l'opération représentée par le span (par exemple, `mysqli_query`). Comportement de la propriété : - *lecture seule*
span_kind	string	Rôle du span. Valeurs possibles : `Unspecified` - le type de span n'est pas spécifié ; `Internal` - opération effectuée au sein d'une application ; `Server` - traitement d'une requête entrante ; `Client` - requête sortante ; `Producer` - création d'une opération asynchrone (par exemple, l'envoi d'un message) ; `Consumer` - traitement d'une opération asynchrone. Comportement de la propriété : - *lecture seule*

Property	Type	Description
service_name	string	Nom du service ayant produit le span (par exemple, `zabbix-frontend`), extrait de l'attribut de ressource OpenTelemetry `service.name`. Comportement de la propriété : - *lecture seule*
resource_attributes	object	Attributs de ressource clé-valeur du span (par exemple, `host.name: eabf8c2a5ddc`). Comportement de la propriété : - *lecture seule*
scope_name	string	Nom de la portée d'instrumentation ayant produit le span. Comportement de la propriété : - *lecture seule*
scope_version	string	Version de la portée d'instrumentation. Comportement de la propriété : - *lecture seule*
span_attributes	object	Attributs du span clé-valeur du span (par exemple, `db.system: mysql`). Comportement de la propriété : - *lecture seule*
duration	integer	Durée du span en nanosecondes. Comportement de la propriété : - *lecture seule*
status_code	string	État de l'opération représentée par le span. Valeurs possibles : `Unset` - aucun état explicite n'a été attribué au span ; `Ok` - l'opération s'est terminée avec succès ; `Error` - l'opération s'est terminée avec une erreur. Comportement de la propriété : - *lecture seule*
status_message	string	Informations supplémentaires, le cas échéant, concernant l'état du span. Comportement de la propriété : - *lecture seule*
events	array	Tableau d'objets événement du span, c'est-à-dire les moments importants survenus pendant l'opération (par exemple, des exceptions). Comportement de la propriété : - *lecture seule*
links	array	Tableau d'objets lien du span, qui pointent vers des spans associés dans la même trace ou dans une autre trace. Comportement de la propriété : - *lecture seule*

apmspan.get

Description

`integer/array apmspan.get(object parameters)`

Cette méthode vous permet de récupérer des spans selon les paramètres indiqués.

Note:
Cette méthode est disponible pour les utilisateurs de tout type. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres des rôles utilisateur. Consultez la page Rôles utilisateur pour plus d'informations.

Paramètres

`(object)` Paramètres définissant le résultat souhaité.

La méthode prend en charge les paramètres suivants.

Paramètre	Type	Description
time_from	timestamp	Retourne uniquement les spans ayant commencé à l'heure indiquée ou après. Comportement du paramètre : - *requis*
time_till	timestamp	Retourne uniquement les spans ayant commencé avant l'heure indiquée. Comportement du paramètre : - *requis*
traceids	string/array	Retourne uniquement les spans appartenant aux traces indiquées.
spanids	string/array	Retourne uniquement les spans possédant les ID indiqués.
parent_spanids	string/array	Retourne uniquement les spans possédant les spans parents indiqués.
resource_attributes	object/array	Retourne uniquement les spans possédant les attributs de ressource indiqués. Format : `[{"key": "<key>", "value": "<value>", "operator": "<operator>"}, ...]`. Un tableau vide retourne tous les spans. Valeurs possibles de l'opérateur : 0 - *(par défaut)* Contient ; 1 - Est égal à ; 2 - Ne contient pas ; 3 - N'est pas égal à ; 4 - Existe ; 5 - N'existe pas.
resource_attributes_evaltype	integer	Méthode d'évaluation des conditions dans `resource_attributes`. Valeurs possibles : 0 - *(par défaut)* Et/Ou ; 2 - Ou.
span_attributes	object/array	Retourne uniquement les spans possédant les attributs de span indiqués. Format : `[{"key": "<key>", "value": "<value>", "operator": "<operator>"}, ...]`. Un tableau vide retourne tous les spans. Valeurs possibles de l'opérateur : 0 - *(par défaut)* Contient ; 1 - Est égal à ; 2 - Ne contient pas ; 3 - N'est pas égal à ; 4 - Existe ; 5 - N'existe pas.
span_attributes_evaltype	integer	Méthode d'évaluation des conditions dans `span_attributes`. Valeurs possibles : 0 - *(par défaut)* Et/Ou ; 2 - Ou.
min_duration	integer	Retourne uniquement les spans dont la durée est supérieure ou égale à la valeur indiquée, en nanosecondes.

Paramètre	Type	Description
max_duration	integer	Retourne uniquement les spans dont la durée est inférieure à la valeur indiquée, en nanosecondes.
filter	object	Retourne uniquement les résultats correspondant exactement au filtre indiqué. Accepte un objet dont les clés sont des noms de propriétés et dont les valeurs sont soit une valeur unique, soit un tableau de valeurs à comparer. Propriétés prises en charge : `traceid`, `spanid`, `parent_spanid`, `trace_state`, `span_name`, `span_kind`, `service_name`, `scope_name`, `scope_version`, `status_code`, `status_message`.
search	object	Retourne les résultats correspondant au modèle indiqué (sans distinction entre majuscules et minuscules). Accepte un objet dont les clés sont des noms de propriétés et dont les valeurs sont des chaînes à rechercher. Si aucune option supplémentaire n'est indiquée, une recherche `LIKE "%…%"` est effectuée. Propriétés prises en charge : `trace_state`, `span_name`, `service_name`, `scope_name`, `scope_version`, `status_message`.
sortfield	string/array	Trie le résultat selon les propriétés indiquées. Valeurs possibles : `timestamp`, `traceid`, `spanid`, `parent_spanid`, `trace_state`, `span_name`, `span_kind`, `service_name`, `scope_name`, `scope_version`, `duration`, `status_code`, `status_message`.
offset	integer	Nombre d'enregistrements à ignorer avant de retourner le résultat.
countOutput	boolean	Ces paramètres sont décrits dans les informations de référence.
excludeSearch	boolean	
limit	integer	
output	query	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Valeurs de retour

`(integer/array)` Renvoie l'un des éléments suivants :

- un tableau d'objets ;
- le nombre d'objets récupérés, si le paramètre `countOutput` a été utilisé.

Exemples

Récupération de la hiérarchie des spans d'une trace

Récupérez les quatre premiers spans de la trace "dee625f33cb0994e86c5cd4640e3b09c", triés par heure de début. La propriété `parent_spanid` indique la relation entre les spans : le span racine n'a pas de parent, et chaque autre span fait référence au span qui l'a démarré.

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "apmspan.get",
    "params": {
        "output": ["spanid", "parent_spanid", "service_name", "span_name", "duration", "status_code"],
        "time_from": 1790848800,
        "time_till": 1790852400,
        "traceids": "dee625f33cb0994e86c5cd4640e3b09c",
        "sortfield": "timestamp",
```

```
        "sortorder": "ASC",
        "limit": 4
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "spanid": "7c1a9e2f4b6d8a03",
            "parent_spanid": "",
            "service_name": "zabbix-nginx",
            "span_name": "POST /zabbix.php",
            "duration": "1104826531",
            "status_code": "Unset"
        },
        {
            "spanid": "3f8b2c6d1e9a7054",
            "parent_spanid": "7c1a9e2f4b6d8a03",
            "service_name": "zabbix-frontend",
            "span_name": "POST /zabbix.php",
            "duration": "1098230418",
            "status_code": "Unset"
        },
        {
            "spanid": "b3390ad41931ed08",
            "parent_spanid": "3f8b2c6d1e9a7054",
            "service_name": "zabbix-frontend",
            "span_name": "mysqli::real_connect",
            "duration": "36671360",
            "status_code": "Ok"
        },
        {
            "spanid": "e41f7a2c9d5b3086",
            "parent_spanid": "3f8b2c6d1e9a7054",
            "service_name": "zabbix-frontend",
            "span_name": "mysqli_query",
            "duration": "19704118",
            "status_code": "Ok"
        }
    ],
    "id": 1
}
```

Voir aussi

- Trace APM
- Journal APM
- Métrique APM

Source

CApmSpan::get() dans *ui/include/classes/api/services/CApmSpan.php*.

Tableau de bord

Cette classe est conçue pour fonctionner avec les tableaux de bord.

Références d'objet :

- Tableau de bord
- Page de tableau de bord
    - Widget de tableau de bord

        * Champ de widget de tableau de bord
- Groupe d'utilisateurs du tableau de bord
- Utilisateur du tableau de bord

Méthodes disponibles :

- dashboard.create - créer de nouveaux tableaux de bord
- dashboard.delete - supprimer des tableaux de bord
- dashboard.get - récupérer des tableaux de bord
- dashboard.update - mettre à jour des tableaux de bord

Objet dashboard

Les objets suivants sont directement liés à l'API `dashboard`.

Tableau de bord

L'objet tableau de bord possède les propriétés suivantes.

Property	Type	Description
dashboardid	ID	ID du tableau de bord. Comportement de la propriété: - *lecture seule* - *obligatoire* pour les opérations de mise à jour
name	string	Nom du tableau de bord. Comportement de la propriété: - *obligatoire* pour les opérations de création
userid	ID	ID de l'utilisateur qui est le propriétaire du tableau de bord.
private	integer	Type de partage du tableau de bord. Valeurs possibles: 0 - tableau de bord public; 1 - *(par défaut)* tableau de bord privé.
display_period	integer	Intervalle par défaut du diaporama (en secondes). Valeurs possibles: 10, 30, 60, 120, 600, 1800, 3600. Par défaut: 30.
auto_start	integer	Démarrage automatique du diaporama. Valeurs possibles: 0 - ne pas démarrer automatiquement le diaporama; 1 - *(par défaut)* démarrer automatiquement le diaporama.

Page du tableau de bord

L'objet page du tableau de bord possède les propriétés suivantes.

Property	Type	Description
dashboard_pageid	ID	ID de la page du tableau de bord. Property behavior: - *read-only*
name	string	Nom de la page du tableau de bord. Default: empty string.
display_period	integer	Intervalle du diaporama du tableau de bord (en secondes). Possible values: 0, 10, 30, 60, 120, 600, 1800, 3600. Default: 0 (will use the default slideshow interval).

Property	Type	Description
widgets	array	Tableau des objets widget du tableau de bord.

Widget de tableau de bord

L'objet widget de tableau de bord possède les propriétés suivantes.

Propriété	Type	Description
widgetid	ID	ID du widget de tableau de bord. Comportement de la propriété: - *lecture seule*
type	string	Type du widget de tableau de bord. Valeurs possibles : actionlog - Journal des actions ; clock - Horloge ; discovery - Statut de découverte ; favgraphs - Graphiques favoris ; favmaps - Cartes favorites ; gauge - Jauge ; geomap - Geomap ; graph - Graphique (classique) ; graphprototype - Prototype de graphique ; honeycomb - Nid d'abeilles ; hostavail - Disponibilité des hôtes ; hostcard - Carte d'hôte ; hostnavigator - Navigateur d'hôtes ; itemcard - Carte d'élément ; itemhistory - Historique des éléments ; itemnavigator - Navigateur d'éléments ; item - Valeur de l'élément ; map - Carte ; navtree - Arborescence de navigation des cartes ; piechart - Diagramme circulaire ; problemhosts - Hôtes avec problèmes ; problems - Problèmes ; problemsbysv - Problèmes par sévérité ; scatterplot - Nuage de points ; slareport - Rapport SLA ; svggraph - Graphique ; systeminfo - Informations système ; tophosts - Principaux hôtes ; topitems - Principaux éléments ; toptriggers - Principaux déclencheurs ; trigover - Aperçu des déclencheurs ; url - URL ; web - Supervision web. Comportement de la propriété: - *obligatoire*
name	string	Nom personnalisé du widget.
x	integer	Position horizontale à partir du côté gauche du tableau de bord. Les valeurs possibles vont de 0 à 71.
y	integer	Position verticale à partir du haut du tableau de bord. Les valeurs possibles vont de 0 à 63.
width	integer	Largeur du widget. Les valeurs possibles vont de 1 à 72.

Propriété	Type	Description
height	integer	Hauteur du widget. Les valeurs possibles vont de 1 à 64.
view_mode	integer	Mode d'affichage du widget. Valeurs possibles : 0 - *(par défaut)* affichage par défaut du widget ; 1 - avec en-tête masqué ;
fields	array	Tableau d'objets champ de widget de tableau de bord. Comportement de la propriété: - voir les widgets individuels dans Champs des widgets de tableau de bord

Champ de widget de tableau de bord

L'objet champ de widget de tableau de bord possède les propriétés suivantes.

Propriété	Type	Description
type	integer	Type du champ de widget. Valeurs possibles : 0 - Entier ; 1 - Chaîne ; 2 - Groupe d'hôtes ; 3 - Hôte ; 4 - Élément ; 5 - Prototype d'élément ; 6 - Graphique ; 7 - Prototype de graphique ; 8 - Carte ; 9 - Service ; 10 - SLA ; 11 - Utilisateur ; 12 - Action ; 13 - Type de média. Comportement de la propriété : - *obligatoire*
name	string	Nom du champ de widget. Valeurs possibles : voir Champs de widget de tableau de bord. Comportement de la propriété : - *obligatoire*
value	mixed	Valeur du champ de widget selon le type. Valeurs possibles : voir Champs de widget de tableau de bord. Comportement de la propriété : - *obligatoire*

Groupe d'utilisateurs du tableau de bord

Liste des autorisations du tableau de bord basées sur les groupes d'utilisateurs. Il possède les propriétés suivantes.

Propriété	Type	Description
usrgrpid	ID	ID du groupe d'utilisateurs. Comportement de la propriété : - *obligatoire*
permission	integer	Type de niveau d'autorisation. Valeurs possibles : 2 - lecture seule ; 3 - lecture-écriture. Comportement de la propriété : - *obligatoire*

Utilisateur du tableau de bord

Liste des autorisations du tableau de bord en fonction des utilisateurs. Elle possède les propriétés suivantes.

Propriété	Type	Description
userid	ID	ID de l'utilisateur. Comportement de la propriété: - *obligatoire*
permission	integer	Type de niveau d'autorisation. Valeurs possibles : 2 - lecture seule ; 3 - lecture-écriture. Comportement de la propriété: - *obligatoire*

dashboard.create

Description

`object dashboard.create(object/array dashboards)`

Cette méthode vous permet de créer de nouveaux tableaux de bord.

Note:
Cette méthode est disponible pour les utilisateurs de tout type. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres des rôles utilisateur. Consultez Rôles utilisateur pour plus d'informations.

Paramètres

`(object/array)` Tableaux de bord à créer.

En plus des propriétés standard du tableau de bord, la méthode accepte les paramètres suivants.

Paramètre	Type	Description
pages	array	Pages du tableau de bord à créer pour le tableau de bord. Les pages du tableau de bord seront ordonnées dans le même ordre que celui spécifié. Comportement des paramètres: - *obligatoire*
users	array	Partages d'utilisateur du tableau de bord à créer sur le tableau de bord.
userGroups	array	Partages de groupe d'utilisateurs du tableau de bord à créer sur le tableau de bord.

Valeurs retournées

(object) Retourne un objet contenant les ID des tableaux de bord créés dans la propriété dashboardids. L'ordre des ID retournés correspond à l'ordre des tableaux de bord transmis.

Exemples

Création d'un tableau de bord

Créez un tableau de bord nommé "My dashboard" avec un widget Problems comportant des tags et en utilisant deux types de partage (groupe d'utilisateurs et utilisateur) sur une seule page de tableau de bord.

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": 30,
        "auto_start": 1,
        "pages": [
            {
                "widgets": [
                    {
                        "type": "problems",
                        "x": 0,
                        "y": 0,
                        "width": 36,
                        "height": 5,
                        "view_mode": 0,
                        "fields": [
                            {
                                "type": 1,
                                "name": "tags.0.tag",
                                "value": "service"
                            },
                            {
                                "type": 0,
                                "name": "tags.0.operator",
                                "value": 1
                            },
                            {
                                "type": 1,
                                "name": "tags.0.value",
                                "value": "zabbix_server"
                            }
                        ]
                    }
                ]
            }
        ],
        "userGroups": [
            {
                "usrgrpid": "7",
                "permission": 2
            }
        ],
        "users": [
            {
                "userid": "4",
                "permission": 3
            }
        ]
    },
    "id": 1
```

```
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "2"
        ]
    },
    "id": 1
}
```

Voir aussi

- Page de tableau de bord
- Widget de tableau de bord
- Champ de widget de tableau de bord
- Utilisateur du tableau de bord
- Groupe d'utilisateurs du tableau de bord

Source

CDashboard::create() dans *ui/include/classes/api/services/CDashboard.php*.

dashboard.delete

Description

`object dashboard.delete(array dashboardids)`

Cette méthode vous permet de supprimer des tableaux de bord.

> **Note:**
> Cette méthode est disponible pour les utilisateurs de tout type. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres des rôles utilisateur. Voir Rôles utilisateur pour plus d'informations.

Paramètres

`(array)` IDs des tableaux de bord à supprimer.

Valeurs retournées

`(object)` Retourne un objet contenant les IDs des tableaux de bord supprimés dans la propriété `dashboardids`.

Exemples

Suppression de plusieurs tableaux de bord

Supprimez deux tableaux de bord.

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.delete",
    "params": [
        "2",
        "3"
    ],
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
```

```
            "2",
            "3"
        ]
    },
    "id": 1
}
```

Source

CDashboard::delete() dans *ui/include/classes/api/services/CDashboard.php*.

dashboard.get

Description

`integer/array dashboard.get(object parameters)`

Cette méthode permet de récupérer les tableaux de bord selon les paramètres fournis.

> **Note:**
> Cette méthode est disponible pour les utilisateurs de tout type. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres des rôles utilisateur. Consultez Rôles utilisateur pour plus d'informations.

Paramètres

`(object)` Paramètres définissant la sortie souhaitée.

La méthode prend en charge les paramètres suivants.

Paramètre	Type	Description
dashboardids	ID/array	Retourner uniquement les tableaux de bord avec les ID donnés.
selectPages	query	Retourner une propriété `pages` avec les pages du tableau de bord, correctement ordonnées.
selectUsers	query	Retourner une propriété `users` avec les utilisateurs avec lesquels le tableau de bord est partagé.
selectUserGroups	query	Retourner une propriété `userGroups` avec les groupes d'utilisateurs avec lesquels le tableau de bord est partagé.
sortfield	string/array	Trier le résultat selon les propriétés données. Valeurs possibles : `dashboardid`.
countOutput	boolean	Ces paramètres sont décrits dans le commentaire de référence.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Valeurs retournées

`(integer/array)` Retourne soit :

- un tableau d'objets ;
- le nombre d'objets récupérés, si le paramètre `countOutput` a été utilisé.

Exemples

Récupération d'un tableau de bord par ID

Récupérez toutes les données des tableaux de bord « 1 » et « 2 ».

Requête :


```
{
    "jsonrpc": "2.0",
    "method": "dashboard.get",
    "params": {
        "output": "extend",
        "selectPages": "extend",
        "selectUsers": "extend",
        "selectUserGroups": "extend",
        "dashboardids": [
            "1",
            "2"
        ]
    },
    "id": 1
}
```


Réponse :


```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "dashboardid": "1",
            "name": "Dashboard",
            "userid": "1",
            "private": "0",
            "display_period": "30",
            "auto_start": "1",
            "users": [],
            "userGroups": [],
            "pages": [
                {
                    "dashboard_pageid": "1",
                    "name": "",
                    "display_period": "0",
                    "widgets": [
                        {
                            "widgetid": "9",
                            "type": "systeminfo",
                            "name": "",
                            "x": "12",
                            "y": "8",
                            "width": "12",
                            "height": "5",
                            "view_mode": "0",
                            "fields": []
                        },
                        {
                            "widgetid": "8",
                            "type": "problemsbysv",
                            "name": "",
                            "x": "12",
                            "y": "4",
                            "width": "12",
                            "height": "4",
                            "view_mode": "0",
                            "fields": []
                        },
                        {
                            "widgetid": "7",
                            "type": "problemhosts",
                            "name": "",
```

```
                "x": "12",
                "y": "0",
                "width": "12",
                "height": "4",
                "view_mode": "0",
                "fields": []
            },
            {
                "widgetid": "6",
                "type": "discovery",
                "name": "",
                "x": "6",
                "y": "9",
                "width": "18",
                "height": "4",
                "view_mode": "0",
                "fields": []
            },
            {
                "widgetid": "5",
                "type": "web",
                "name": "",
                "x": "0",
                "y": "9",
                "width": "18",
                "height": "4",
                "view_mode": "0",
                "fields": []
            },
            {
                "widgetid": "4",
                "type": "problems",
                "name": "",
                "x": "0",
                "y": "3",
                "width": "12",
                "height": "6",
                "view_mode": "0",
                "fields": []
            },
            {
                "widgetid": "3",
                "type": "favmaps",
                "name": "",
                "x": "8",
                "y": "0",
                "width": "12",
                "height": "3",
                "view_mode": "0",
                "fields": []
            },
            {
                "widgetid": "1",
                "type": "favgraphs",
                "name": "",
                "x": "0",
                "y": "0",
                "width": "12",
                "height": "3",
                "view_mode": "0",
                "fields": []
            }
```

```
                ]
            },
            {
                "dashboard_pageid": "2",
                "name": "",
                "display_period": "0",
                "widgets": []
            },
            {
                "dashboard_pageid": "3",
                "name": "Custom page name",
                "display_period": "60",
                "widgets": []
            }
        ]
    },
    {
        "dashboardid": "2",
        "name": "My dashboard",
        "userid": "1",
        "private": "1",
        "display_period": "60",
        "auto_start": "1",
        "users": [
            {
                "userid": "4",
                "permission": "3"
            }
        ],
        "userGroups": [
            {
                "usrgrpid": "7",
                "permission": "2"
            }
        ],
        "pages": [
            {
                "dashboard_pageid": "4",
                "name": "",
                "display_period": "0",
                "widgets": [
                    {
                        "widgetid": "10",
                        "type": "problems",
                        "name": "",
                        "x": "0",
                        "y": "0",
                        "width": "12",
                        "height": "5",
                        "view_mode": "0",
                        "fields": [
                            {
                                "type": "2",
                                "name": "groupids",
                                "value": "4"
                            }
                        ]
                    }
                ]
            }
        ]
    }
```

```
    ],
    "id": 1
}
```

Voir aussi

- Page de tableau de bord
- Widget de tableau de bord
- Champ de widget de tableau de bord
- Utilisateur du tableau de bord
- Groupe d'utilisateurs du tableau de bord

Source

CDashboard::get() dans *ui/include/classes/api/services/CDashboard.php*.

dashboard.update

Description

`object dashboard.update(object/array dashboards)`

Cette méthode vous permet de mettre à jour des tableaux de bord existants.

> **Note:**
> Cette méthode est disponible pour les utilisateurs de tout type. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres des rôles utilisateur. Consultez Rôles utilisateur pour plus d'informations.

Paramètres

`(object/array)` Propriétés du tableau de bord à mettre à jour.

La propriété `dashboardid` doit être définie pour chaque tableau de bord, toutes les autres propriétés sont facultatives. Seules les propriétés transmises seront mises à jour, toutes les autres resteront inchangées.

En plus des propriétés standard du tableau de bord, la méthode accepte les paramètres suivants.

Paramètre	Type	Description
pages	array	Pages du tableau de bord pour remplacer les pages existantes du tableau de bord. Les pages du tableau de bord sont mises à jour à l'aide de la propriété `dashboard_pageid`. De nouvelles pages du tableau de bord seront créées pour les objets sans propriété `dashboard_pageid` et les pages existantes du tableau de bord seront supprimées si elles ne sont pas réutilisées. Les pages du tableau de bord seront ordonnées dans le même ordre que celui spécifié. Seules les propriétés spécifiées des pages du tableau de bord seront mises à jour.
users	array	Partages d'utilisateur du tableau de bord pour remplacer les éléments existants.
userGroups	array	Partages de groupe d'utilisateurs du tableau de bord pour remplacer les éléments existants.

Valeurs retournées

`(object)` Retourne un objet contenant les identifiants des tableaux de bord mis à jour dans la propriété `dashboardids`.

Exemples

Renommer un tableau de bord

Renommez un tableau de bord en « SQL server status ».

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.update",
    "params": {
        "dashboardid": "2",
        "name": "SQL server status"
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "2"
        ]
    },
    "id": 1
}
```

Mise à jour des pages du tableau de bord

Renommez la première page du tableau de bord, remplacez les widgets de la deuxième page du tableau de bord et ajoutez une nouvelle page en troisième position. Supprimez toutes les autres pages du tableau de bord.

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.update",
    "params": {
        "dashboardid": "2",
        "pages": [
            {
                "dashboard_pageid": 1,
                "name": "Page renommée"
            },
            {
                "dashboard_pageid": 2,
                "widgets": [
                    {
                        "type": "clock",
                        "x": 0,
                        "y": 0,
                        "width": 12,
                        "height": 3
                    }
                ]
            },
            {
                "display_period": 60
            }
        ]
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "2"
```

```
        ]
    },
    "id": 1
}
```

Changer le propriétaire du tableau de bord

Disponible uniquement pour les admins et les super admins.

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.update",
    "params": {
        "dashboardid": "2",
        "userid": "1"
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "2"
        ]
    },
    "id": 1
}
```

Voir aussi

- Page de tableau de bord
- Widget de tableau de bord
- Champ de widget de tableau de bord
- Utilisateur du tableau de bord
- Groupe d'utilisateurs du tableau de bord

Source

CDashboard::update() dans *ui/include/classes/api/services/CDashboard.php*.

Champs du widget de tableau de bord

Cette page contient des liens de navigation vers les paramètres des widgets de tableau de bord et les valeurs de propriété possibles pour les objets dashboard widget field respectifs.

Pour voir les paramètres et les valeurs de propriété de chaque widget, veuillez utiliser la barre latérale pour accéder à la page du widget correspondante.

######1 Journal des actions {#manual-api-reference-dashboard-widget_fields-action_log}

Description

Ces paramètres et les valeurs de propriété possibles pour les objets de champ respectifs du widget de tableau de bord vous permettent de configurer le widget *Journal des actions* dans les méthodes `dashboard.create` et `dashboard.update`.

> **Attention:**
> Les propriétés `fields` du widget ne sont pas validées lors de la création ou de la mise à jour d'un tableau de bord. Cela permet aux utilisateurs de modifier les widgets intégrés et de créer des widgets personnalisés, mais introduit également le risque de créer ou de mettre à jour des widgets de manière incorrecte. Pour garantir la création ou la mise à jour réussie du widget *Journal des actions*, veuillez vous reporter au comportement des paramètres décrit dans les tableaux ci-dessous.

Paramètres

Les paramètres suivants sont pris en charge pour le widget *Journal des actions*.

Paramètre	type	name	value
Intervalle d'actualisation	0	rf_rate	0 - Aucune actualisation ; 10 - 10 secondes ; 30 - 30 secondes ; 60 - *(par défaut)* 1 minute ; 120 - 2 minutes ; 600 - 10 minutes ; 900 - 15 minutes.
Destinataires	11	userids.0	ID de l'utilisateur. Remarque : pour configurer plusieurs utilisateurs, créez un objet de champ de widget de tableau de bord pour chaque utilisateur, avec un numéro incrémenté dans le nom de la propriété.
Actions	12	actionids.0	ID de l'action. Remarque : pour configurer plusieurs actions, créez un objet de champ de widget de tableau de bord pour chaque action, avec un numéro incrémenté dans le nom de la propriété.
Types de média	13	mediatypeids.0	ID du type de média. Remarque : pour configurer plusieurs types de média, créez un objet de champ de widget de tableau de bord pour chaque type de média, avec un numéro incrémenté dans le nom de la propriété.
État	0	statuses.0	0 - En cours ; 1 - Envoyé/Exécuté ; 2 - Échec. Remarque : pour configurer plusieurs valeurs, créez un objet de champ de widget de tableau de bord pour chaque valeur, avec un numéro incrémenté dans le nom de la propriété.
Chaîne de recherche	1	message	Toute valeur de chaîne.
Période	1	time_period._reference	`DASHBOARD._timeperiod` - définir le sélecteur de période du tableau de bord comme source de données ; `ABCDE._timeperiod` - définir un widget compatible (avec son paramètre *Référence* défini sur "`ABCDE`") comme source de données. Par défaut : `DASHBOARD._timeperiod` Vous pouvez également définir la période uniquement dans les paramètres *Du* et *Au*.
Du	1	time_period.from	Chaîne temporelle valide au format absolu (`YYYY-MM-DD hh:mm:ss`) ou au format temporel relatif (`now`, `now/d`, `now/w-1w`, etc.). Comportement du paramètre : - *pris en charge* si *Période* n'est pas défini - *requis* si `time_period.to` est défini
Au	1	time_period.to	Chaîne temporelle valide au format absolu (`YYYY-MM-DD hh:mm:ss`) ou au format temporel relatif (`now`, `now/d`, `now/w-1w`, etc.). Comportement du paramètre : - *pris en charge* si *Période* n'est pas défini - *requis* si `time_period.from` est défini

Paramètre	type	name	value
Trier les entrées par	0	sort_triggers	3 - Heure (croissante) ; 4 - *(par défaut)* Heure (décroissante) ; 5 - Type (croissant) ; 6 - Type (décroissant) ; 7 - État (croissant) ; 8 - État (décroissant) ; 11 - Destinataire (croissant) ; 12 - Destinataire (décroissant).
Afficher les lignes	0	show_lines	Les valeurs possibles vont de 1 à 100. Par défaut : 25.

Exemples

Les exemples suivants visent à décrire uniquement la configuration des objets de champ du widget de tableau de bord pour le widget *Action log*. Pour plus d'informations sur la configuration d'un tableau de bord, consultez `dashboard.create`.

Configuration d'un widget *Journal des actions*

Configurez un widget *Journal des actions* qui affiche 10 entrées des détails des opérations d'action, triées par heure (dans l'ordre croissant). En outre, affichez uniquement les détails des opérations d'action qui ont tenté d'envoyer un e-mail à l'utilisateur « 1 », mais sans succès.

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": 30,
        "auto_start": 1,
        "pages": [
            {
                "widgets": [
                    {
                        "type": "actionlog",
                        "name": "Action log",
                        "x": 0,
                        "y": 0,
                        "width": 36,
                        "height": 5,
                        "view_mode": 0,
                        "fields": [
                            {
                                "type": 0,
                                "name": "show_lines",
                                "value": 10
                            },
                            {
                                "type": 0,
                                "name": "sort_triggers",
                                "value": 3
                            },
                            {
                                "type": 11,
                                "name": "userids.0",
                                "value": 1
                            },
                            {
                                "type": 13,
                                "name": "mediatypeids.0",
                                "value": 1
```

```
                            },
                            {
                                "type": 0,
                                "name": "statuses.0",
                                "value": 2
                            }
                        ]
                    }
                ]
            }
        ],
        "userGroups": [
            {
                "usrgrpid": 7,
                "permission": 2
            }
        ],
        "users": [
            {
                "userid": 1,
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Voir également

- Champ du widget du tableau de bord
- `dashboard.create`
- `dashboard.update`

2 Horloge

Description

Ces paramètres et les valeurs de propriété possibles pour les objets de champ respectifs du widget de tableau de bord vous permettent de configurer le widget *Horloge* dans les méthodes `dashboard.create` et `dashboard.update`.

Attention:
Les propriétés `fields` du widget ne sont pas validées lors de la création ou de la mise à jour d'un tableau de bord. Cela permet aux utilisateurs de modifier les widgets intégrés et de créer des widgets personnalisés, mais introduit également le risque de créer ou de mettre à jour des widgets de manière incorrecte. Pour garantir la création ou la mise à jour réussie du widget *Horloge*, veuillez vous reporter au comportement des paramètres décrit dans les tableaux ci-dessous.

Paramètres

Les paramètres suivants sont pris en charge pour le widget *Horloge*.

Paramètre	type	name	value
Intervalle d'actualisation	0	rf_rate	0 - Pas d'actualisation ; 10 - 10 secondes ; 30 - 30 secondes ; 60 - 1 minute ; 120 - 2 minutes ; 600 - 10 minutes ; 900 - *(par défaut)* 15 minutes.
Type d'heure	0	time_type	0 - *(par défaut)* Heure locale ; 1 - Heure du serveur ; 2 - Heure de l'hôte.
Type d'horloge	0	clock_type	0 - *(par défaut)* Analogique ; 1 - Numérique.

Les paramètres suivants sont pris en charge si *Type d'heure* est défini sur "Heure de l'hôte".

Paramètre	type	name	value
Élément	4	itemid.0	ID de l'élément. Comportement du paramètre : - *obligatoire*

Les paramètres suivants sont pris en charge si *Type d'horloge* est défini sur "Numérique".

Paramètre	type	name	value
Afficher	0	show.0	1 - Date ; 2 - *(par défaut)* Heure ; 3 - Fuseau horaire. Remarque : pour configurer plusieurs valeurs, créez un objet de champ de widget de tableau de bord pour chaque valeur avec un numéro incrémenté dans le nom de la propriété.

Configuration avancée

Les paramètres de configuration avancée suivants sont pris en charge si le *Type d'horloge* est défini sur « Numérique ».

Paramètre	type	name	value
Couleur d'arrière-plan	1	bg_color	Code couleur hexadécimal (par ex. FF0000). Par défaut : "" (vide).

Date

Les paramètres de configuration avancés suivants sont pris en charge si *Clock type* est défini sur « Digital » et *Show* sur « Date ».

Parameter	type	name	value
Gras	0	date_bold	0 - *(par défaut)* Désactivé ; 1 - Activé.
Couleur	1	date_color	Code couleur hexadécimal (par exemple, FF0000). Par défaut : "" (vide).

Heure

Les paramètres de configuration avancés suivants sont pris en charge si *Type d'horloge* est défini sur « Digital » et *Afficher* sur « Heure ».

Paramètre	type	name	value
Gras	0	time_bold	0 - *(par défaut)* Désactivé; 1 - Activé.
Couleur	1	time_color	Code couleur hexadécimal (par ex. `FF0000`). Par défaut : `""` (vide).
Secondes	0	time_sec	0 - Désactivé; 1 - *(par défaut)* Activé.
Format	0	time_format	0 - *(par défaut)* 24 heures; 1 - 12 heures.

Fuseau horaire

Les paramètres de configuration avancée suivants sont pris en charge si *Type d'horloge* est défini sur « Digital » et *Afficher* est défini sur « Fuseau horaire ».

Paramètre	type	name	value
Gras	0	tzone_bold	0 - *(par défaut)* Désactivé; 1 - Activé.
Couleur	1	tzone_color	Code couleur hexadécimal (par ex. `FF0000`). Par défaut : `""` (vide).
Fuseau horaire	1	tzone_timezone	Chaîne de fuseau horaire valide (par ex. `Europe/Riga`, `system`, `UTC`, etc.). Pour la liste complète des fuseaux horaires pris en charge, veuillez consulter la documentation PHP. Par défaut : `local`. Comportement du paramètre : - *pris en charge* si *Type d'heure* est défini sur « Heure locale » ou « Heure du serveur »
Format	0	tzone_format	0 - *(par défaut)* Court; 1 - Complet. Comportement du paramètre : - *pris en charge* si *Type d'heure* est défini sur « Heure locale » ou « Heure du serveur »

Exemples

Les exemples suivants visent à décrire uniquement la configuration des objets de champ du widget de tableau de bord pour le widget *Horloge*. Pour plus d'informations sur la configuration d'un tableau de bord, consultez `dashboard.create`.

Configuration d'un widget *Clock*

Configurez un widget *Clock* qui affiche la date locale, l'heure et le fuseau horaire dans une horloge numérique personnalisée.

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": 30,
        "auto_start": 1,
        "pages": [
            {
                "widgets": [
                    {
                        "type": "clock",
                        "name": "Clock",
                        "x": 0,
```

```
                    "y": 0,
                    "width": 12,
                    "height": 3,
                    "view_mode": 0,
                    "fields": [
                        {
                            "type": 0,
                            "name": "clock_type",
                            "value": 1
                        },
                        {
                            "type": 0,
                            "name": "show.0",
                            "value": 1
                        },
                        {
                            "type": 0,
                            "name": "show.1",
                            "value": 2
                        },
                        {
                            "type": 0,
                            "name": "show.2",
                            "value": 3
                        },
                        {
                            "type": 1,
                            "name": "date_color",
                            "value": "E1E1E1"
                        },
                        {
                            "type": 0,
                            "name": "time_bold",
                            "value": 1
                        },
                        {
                            "type": 1,
                            "name": "tzone_color",
                            "value": "E1E1E1"
                        },
                        {
                            "type": 1,
                            "name": "tzone_timezone",
                            "value": "Europe/Riga"
                        },
                        {
                            "type": 0,
                            "name": "tzone_format",
                            "value": 1
                        }
                    ]
                }
            ]
        }
    ],
    "userGroups": [
        {
            "usrgrpid": 7,
            "permission": 2
        }
    ],
    "users": [
```

```
            {
                "userid": 1,
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Voir également

- Champ du widget du tableau de bord
- `dashboard.create`
- `dashboard.update`

3 État de la découverte

Description

Ces paramètres et les valeurs de propriété possibles pour les objets de champ respectifs du widget de tableau de bord vous permettent de configurer le widget *État de la découverte* dans les méthodes `dashboard.create` et `dashboard.update`.

Paramètres

Les paramètres suivants sont pris en charge pour le widget *Statut de découverte*.

Paramètre	type	name	value
Intervalle de rafraîchissement	0	rf_rate	0 - Aucun rafraîchissement; 10 - 10 secondes; 30 - 30 secondes; 60 - *(par défaut)* 1 minute; 120 - 2 minutes; 600 - 10 minutes; 900 - 15 minutes.

Exemples

Les exemples suivants visent uniquement à décrire la configuration des objets de champ du widget de tableau de bord pour le widget *État de la découverte*. Pour plus d'informations sur la configuration d'un tableau de bord, consultez `dashboard.create`.

Configuration du widget *État de la découverte*

Configurez un widget *État de la découverte* avec l'intervalle d'actualisation défini sur 15 minutes.

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": 30,
        "auto_start": 1,
```

```
        "pages": [
            {
                "widgets": [
                    {
                        "type": "discovery",
                        "name": "Discovery status",
                        "x": 0,
                        "y": 0,
                        "width": 18,
                        "height": 3,
                        "view_mode": 0,
                        "fields": [
                            {
                                "type": 0,
                                "name": "rf_rate",
                                "value": 900
                            }
                        ]
                    }
                ]
            }
        ],
        "userGroups": [
            {
                "usrgrpid": 7,
                "permission": 2
            }
        ],
        "users": [
            {
                "userid": 1,
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Voir aussi

- Champ de widget de tableau de bord
- `dashboard.create`
- `dashboard.update`

4 Graphiques favoris

Description

Ces paramètres et les valeurs de propriété possibles pour les objets de champ du widget de tableau de bord correspondant vous permettent de configurer le widget *Graphiques favoris* dans les méthodes `dashboard.create` et `dashboard.update`.

Paramètres

Les paramètres suivants sont pris en charge pour le widget *Graphiques favoris*.

Paramètre	type	name	value
Intervalle de rafraîchissement	0	rf_rate	0 - Aucun rafraîchissement; 10 - 10 secondes; 30 - 30 secondes; 60 - 1 minute; 120 - 2 minutes; 600 - 10 minutes; 900 - *(par défaut)* 15 minutes.

Exemples

Les exemples suivants visent uniquement à décrire la configuration des objets de champ du widget de tableau de bord pour le widget *Favorite graphs*. Pour plus d'informations sur la configuration d'un tableau de bord, consultez `dashboard.create`.

Configuration d'un widget *Graphiques favoris*

Configurez un widget *Graphiques favoris* avec l'intervalle d'actualisation défini sur 10 minutes.

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": 30,
        "auto_start": 1,
        "pages": [
            {
                "widgets": [
                    {
                        "type": "favgraphs",
                        "name": "Favorite graphs",
                        "x": 0,
                        "y": 0,
                        "width": 12,
                        "height": 3,
                        "view_mode": 0,
                        "fields": [
                            {
                                "type": 0,
                                "name": "rf_rate",
                                "value": 600
                            }
                        ]
                    }
                ]
            }
        ],
        "userGroups": [
            {
                "usrgrpid": 7,
                "permission": 2
            }
        ],
        "users": [
            {
                "userid": 1,
                "permission": 3
            }
        ]
    },
```

```
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Voir aussi

- Champ de widget de tableau de bord
- `dashboard.create`
- `dashboard.update`

5 Cartes favorites

Description

Ces paramètres et les valeurs de propriété possibles pour les objets de champ correspondants du widget de tableau de bord vous permettent de configurer le widget *Favorite maps* dans les méthodes `dashboard.create` et `dashboard.update`.

Paramètres

Les paramètres suivants sont pris en charge pour le widget *Cartes favorites*.

Paramètre	type	name	value
Intervalle de rafraîchissement	0	rf_rate	0 - Aucun rafraîchissement; 10 - 10 secondes; 30 - 30 secondes; 60 - 1 minute; 120 - 2 minutes; 600 - 10 minutes; 900 - *(par défaut)* 15 minutes.

Exemples

Les exemples suivants visent à décrire uniquement la configuration des objets de champ du widget de tableau de bord pour le widget *Cartes préférées*. Pour plus d'informations sur la configuration d'un tableau de bord, consultez `dashboard.create`.

Configuration d'un widget *Cartes favorites*

Configurez un widget *Cartes favorites* avec l'intervalle d'actualisation défini sur 10 minutes.

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": 30,
        "auto_start": 1,
        "pages": [
            {
                "widgets": [
                    {
                        "type": "favmaps",
                        "name": "Favorite maps",
```

```
                        "x": 0,
                        "y": 0,
                        "width": 12,
                        "height": 3,
                        "view_mode": 0,
                        "fields": [
                            {
                                "type": 0,
                                "name": "rf_rate",
                                "value": 600
                            }
                        ]
                    }
                ]
            }
        ],
        "userGroups": [
            {
                "usrgrpid": 7,
                "permission": 2
            }
        ],
        "users": [
            {
                "userid": 1,
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Voir aussi

- Champ de widget de tableau de bord
- `dashboard.create`
- `dashboard.update`

6 Jauge

Description

Ces paramètres et les valeurs de propriété possibles pour les objets de champ respectifs du widget de tableau de bord vous permettent de configurer le widget *Jauge* dans les méthodes `dashboard.create` et `dashboard.update`.

Attention:
Les propriétés `fields` du widget ne sont pas validées lors de la création ou de la mise à jour d'un tableau de bord. Cela permet aux utilisateurs de modifier les widgets intégrés et de créer des widgets personnalisés, mais introduit également le risque de créer ou de mettre à jour des widgets de manière incorrecte. Pour garantir la réussite de la création ou de la mise à jour du widget *Jauge*, veuillez vous reporter au comportement des paramètres décrit dans les tableaux ci-dessous.

Paramètres

Les paramètres suivants sont pris en charge pour le widget *Jauge*.

Paramètre	type	name	value
Intervalle d'actualisation	0	rf_rate	0 - Aucune actualisation; 10 - 10 secondes; 30 - 30 secondes; 60 - *(par défaut)* 1 minute; 120 - 2 minutes; 600 - 10 minutes; 900 - 15 minutes.
Élément	4	itemid.0	ID de l'élément. Comportement du paramètre : - *obligatoire* si *Élément (widget)* n'est pas défini
Élément (widget)	1	itemid._reference	Au lieu de l'ID de l'élément : `ABCDE._itemid` - définir un widget compatible (avec son paramètre *Référence* défini sur "ABCDE") comme source de données pour les éléments. Comportement du paramètre : - *obligatoire* si *Élément* n'est pas défini
Min	1	min	Toute valeur numérique. Les suffixes (par ex. "1d", "2w", "4K", "8G") sont pris en charge. Par défaut : "0".
Max	1	max	Toute valeur numérique. Les suffixes (par ex. "1d", "2w", "4K", "8G") sont pris en charge. Par défaut : "100".
Arc de valeur	1	value_arc_color	Code couleur hexadécimal (par ex. `FF0000`). Par défaut : `""` (vide).
Arrière-plan de l'arc	1	empty_color	Code couleur hexadécimal (par ex. `FF0000`). Par défaut : `""` (vide).
Arrière-plan	1	bg_color	Code couleur hexadécimal (par ex. `FF0000`). Par défaut : `""` (vide).

Paramètre	type	name	value
Afficher	0	show.0	1 - Description; 2 - Valeur; 3 - Aiguille; 4 - Échelle; 5 - Arc de valeur. Remarque : pour configurer plusieurs valeurs, créez un objet de champ de widget de tableau de bord pour chaque valeur avec un numéro incrémenté dans le nom de propriété. Par défaut : 1, 2, 4, 5. Les valeurs ”Aiguille” et ”Échelle” ne sont pas prises en charge si les deux conditions suivantes sont remplies : - aucun objet de champ de widget de tableau de bord pour *Afficher* avec la valeur ”Arc de valeur” n’est défini ; - le paramètre de configuration avancée *Afficher l’arc* est défini sur ”Désactivé”. Les paramètres de configuration avancée pour les options de *Afficher* ne sont pas pris en charge si aucun objet de champ de widget de tableau de bord avec les valeurs correspondantes n’est défini.
Remplacer l’hôte	1	override_hostid._reference	`ABCDE._hostid` - définir un widget compatible (avec son paramètre *Référence* défini sur ”`ABCDE`”) comme source de données pour les hôtes ; `DASHBOARD._hostid` - définir le sélecteur d’hôte du tableau de bord comme source de données pour les hôtes. Ce paramètre n’est pas pris en charge lors de la configuration du widget sur un tableau de bord de modèle.

Configuration avancée

Les paramètres de configuration avancée suivants sont pris en charge pour le widget *Jauge*.

Note:
Le nombre dans le nom de la propriété *Seuils* (par exemple, thresholds.0.color) fait référence à la position du seuil dans une liste triée par ordre croissant. Cependant, si les seuils sont configurés dans un ordre différent, les valeurs seront triées par ordre croissant après la mise à jour de la configuration du widget dans l’interface Zabbix (par exemple, `"thresholds.0.threshold":"5"` → `"thresholds.0.threshold":"1"` ; `"thresholds.1.threshold":"1"` → `"thresholds.1.threshold": "5"`).

Paramètre	type	name	value
Angle	0	angle	Valeurs possibles : 180 *(par défaut)* ou 270.
Description			
Description	1	description	Toute valeur de type chaîne, y compris les macros. Macros prises en charge : `{HOST.*}`, `{ITEM.*}`, `{INVENTORY.*}`, macros utilisateur. Par défaut : `{ITEM.NAME}`.
Taille	0	desc_size	Les valeurs possibles vont de 1 à 100. Par défaut : 15.
Position verticale	0	desc_v_pos	0 - En haut ; 1 - *(par défaut)* En bas.
Gras	0	desc_bold	0 - *(par défaut)* Désactivé ; 1 - Activé.

Paramètre	type	name	value
Couleur	1	desc_color	Code couleur hexadécimal (par exemple, FF0000). Par défaut : "" (vide).
Valeur			
Décimales	0	decimal_places	Les valeurs possibles vont de 1 à 10. Par défaut : 2.
Taille	0	value_size	Les valeurs possibles vont de 1 à 100. Par défaut : 25.
Gras	0	value_bold	0 - *(par défaut)* Désactivé ; 1 - Activé.
Couleur	1	value_color	Code couleur hexadécimal (par exemple, FF0000). Par défaut : "" (vide).
Unités			
Unités (case à cocher)	0	units_show	0 - Désactivées ; 1 - *(par défaut)* Activées.
Unités (valeur)	1	units	Toute valeur de type chaîne. Comportement du paramètre : - *pris en charge* si *Unités* (case à cocher) est défini sur « Activées »
Taille	0	units_size	Les valeurs possibles vont de 1 à 100. Par défaut : 25. Comportement du paramètre : - *pris en charge* si *Unités* (case à cocher) est défini sur « Activées »
Gras	0	units_bold	0 - *(par défaut)* Désactivé ; 1 - Activé. Comportement du paramètre : - *pris en charge* si *Unités* (case à cocher) est défini sur « Activées »
Position	0	units_pos	0 - Avant la valeur ; 1 - Au-dessus de la valeur ; 2 - *(par défaut)* Après la valeur ; 3 - En dessous de la valeur. Comportement du paramètre : - *pris en charge* si *Unités* (case à cocher) est défini sur « Activées » Ce paramètre est ignoré s'il est défini sur l'une des unités liées au temps suivantes : unixtime, uptime, s.
Couleur	1	units_color	Code couleur hexadécimal (par exemple, FF0000). Par défaut : "" (vide).
Arc de valeur			
Taille de l'arc	0	value_arc_size	Les valeurs possibles vont de 1 à 100. Par défaut : 20.
Aiguille			
Couleur	1	needle_color	Code couleur hexadécimal (par exemple, FF0000). Par défaut : "" (vide). Comportement du paramètre : - *pris en charge* si un objet de champ de widget de tableau de bord *Afficher* avec la valeur « Arc de valeur » est défini, ou si *Afficher l'arc* est défini sur « Activé »
Échelle			

Paramètre	type	name	value
Afficher les unités	0	scale_show_units	0 - Désactivées ; 1 - *(par défaut)* Activées. Comportement du paramètre : - *pris en charge* si *Unités* (case à cocher) est défini sur « Activées » et si un objet de champ de widget de tableau de bord *Afficher* avec la valeur « Arc de valeur » est défini ou si *Afficher l'arc* est défini sur « Activé »
Taille	0	scale_size	Les valeurs possibles vont de 1 à 100. Par défaut : 15. Comportement du paramètre : - *pris en charge* si un objet de champ de widget de tableau de bord *Afficher* avec la valeur « Arc de valeur » est défini, ou si *Afficher l'arc* est défini sur « Activé »
Décimales	0	scale_decimal_places	Les valeurs possibles vont de 1 à 10. Par défaut : 0. Comportement du paramètre : - *pris en charge* si un objet de champ de widget de tableau de bord *Afficher* avec la valeur « Arc de valeur » est défini, ou si *Afficher l'arc* est défini sur « Activé »
Seuils			
Couleur	1	thresholds.0.color	Code couleur hexadécimal (par exemple, FF0000).
Seuil	1	thresholds.0.threshold	Toute valeur numérique. Les suffixes (par exemple, « 1d », « 2w », « 4K », « 8G ») sont pris en charge.
Afficher les libellés	0	th_show_labels	0 - *(par défaut)* Désactivé ; 1 - Activé. Comportement du paramètre : - *pris en charge* si des *Seuils* sont définis et si un objet de champ de widget de tableau de bord *Afficher* avec la valeur « Arc de valeur » est défini ou si *Afficher l'arc* est défini sur « Activé »
Afficher l'arc	0	th_show_arc	0 - *(par défaut)* Désactivé ; 1 - Activé. Comportement du paramètre : - *pris en charge* si des *Seuils* sont définis
Taille de l'arc	0	th_arc_size	Les valeurs possibles vont de 1 à 100. Par défaut : 5. Comportement du paramètre : - *pris en charge* si *Afficher l'arc* est défini sur « Activé »

Exemples

Les exemples suivants visent uniquement à décrire la configuration des objets de champ du widget de tableau de bord pour le widget *Gauge*. Pour plus d'informations sur la configuration d'un tableau de bord, consultez `dashboard.create`.

Configuration d'un widget *Gauge*

Configurez un widget *Gauge* qui affiche la valeur d'élément pour l'élément « 44474 » (Interface enp0s3: Bits envoyés). De plus, ajustez finement l'apparence du widget à l'aide de plusieurs options avancées, y compris des seuils.

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
```

```
        "display_period": 30,
        "auto_start": 1,
        "pages": [
            {
                "widgets": [
                    {
                        "type": "gauge",
                        "name": "Gauge",
                        "x": 0,
                        "y": 0,
                        "width": 18,
                        "height": 5,
                        "view_mode": 0,
                        "fields": [
                            {
                                "type": 4,
                                "name": "itemid.0",
                                "value": 44474
                            },
                            {
                                "type": 1,
                                "name": "min",
                                "value": "100000"
                            },
                            {
                                "type": 1,
                                "name": "max",
                                "value": "1000000"
                            },
                            {
                                "type": 0,
                                "name": "show.0",
                                "value": 1
                            },
                            {
                                "type": 0,
                                "name": "show.1",
                                "value": 2
                            },
                            {
                                "type": 0,
                                "name": "show.2",
                                "value": 3
                            },
                            {
                                "type": 0,
                                "name": "show.4",
                                "value": 4
                            },
                            {
                                "type": 0,
                                "name": "show.5",
                                "value": 5
                            },
                            {
                                "type": 0,
                                "name": "angle",
                                "value": 270
                            },
                            {
                                "type": 0,
                                "name": "desc_size",
```

```
            "value": 10
        },
        {
            "type": 0,
            "name": "desc_bold",
            "value": 1
        },
        {
            "type": 0,
            "name": "decimal_places",
            "value": 0
        },
        {
            "type": 0,
            "name": "value_bold",
            "value": 1
        },
        {
            "type": 0,
            "name": "units_size",
            "value": 15
        },
        {
            "type": 0,
            "name": "units_pos",
            "value": 3
        },
        {
            "type": 1,
            "name": "needle_color",
            "value": "3C3C3C"
        },
        {
            "type": 1,
            "name": "thresholds.0.color",
            "value": "FF465C"
        },
        {
            "type": 1,
            "name": "thresholds.0.threshold",
            "value": "700000"
        },
        {
            "type": 1,
            "name": "thresholds.1.color",
            "value": "FFD54F"
        },
        {
            "type": 1,
            "name": "thresholds.1.threshold",
            "value": "500000"
        },
        {
            "type": 1,
            "name": "thresholds.2.color",
            "value": "0EC9AC"
        },
        {
            "type": 1,
            "name": "thresholds.2.threshold",
            "value": "100000"
        },
```

```
                            {
                                "type": 0,
                                "name": "th_show_labels",
                                "value": 1
                            },
                            {
                                "type": 0,
                                "name": "th_show_arc",
                                "value": 1
                            },
                            {
                                "type": 0,
                                "name": "th_arc_size",
                                "value": 15
                            }
                        ]
                    }
                ]
            }
        ],
        "userGroups": [
            {
                "usrgrpid": 7,
                "permission": 2
            }
        ],
        "users": [
            {
                "userid": 1,
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Voir aussi

- Champ de widget de tableau de bord
- `dashboard.create`
- `dashboard.update`

7 Géocarte

Description

Ces paramètres et les valeurs de propriété possibles pour les objets de champ respectifs du widget de tableau de bord vous permettent de configurer le widget *Geomap* dans les méthodes `dashboard.create` et `dashboard.update`.

Attention:
Les propriétés `fields` du widget ne sont pas validées lors de la création ou de la mise à jour d'un tableau de bord. Cela permet aux utilisateurs de modifier les widgets intégrés et de créer des widgets personnalisés, mais introduit également le risque de créer ou de mettre à jour des widgets de manière incorrecte. Pour garantir la création ou la mise à jour réussie du widget *Geomap*, veuillez vous reporter au comportement des paramètres décrit dans les tableaux ci-dessous.

Paramètres

Les paramètres suivants sont pris en charge pour le widget *Geomap*.

Parameter	type	name	value
Intervalle d'actualisation	0	rf_rate	0 - Aucune actualisation ; 10 - 10 secondes ; 30 - 30 secondes ; 60 - *(par défaut)* 1 minute ; 120 - 2 minutes ; 600 - 10 minutes ; 900 - 15 minutes.
Groupes d'hôtes	2	groupids.0	ID du groupe d'hôtes. Remarque : pour configurer plusieurs groupes d'hôtes, créez un objet de champ de widget de tableau de bord pour chaque groupe d'hôtes, avec un numéro incrémenté dans le nom de la propriété. Ce paramètre n'est pas pris en charge lors de la configuration du widget sur un tableau de bord de modèle.
Groupes d'hôtes (widget)	1	groupids._reference	Au lieu de l'ID du groupe d'hôtes : `ABCDE._hostgroupids` - définissez un widget compatible (avec son paramètre *Reference* défini sur "`ABCDE`") comme source de données des groupes d'hôtes. Ce paramètre n'est pas pris en charge lors de la configuration du widget sur un tableau de bord de modèle.
Hôtes	3	hostids.0	ID de l'hôte. Remarque : pour configurer plusieurs hôtes, créez un objet de champ de widget de tableau de bord pour chaque hôte, avec un numéro incrémenté dans le nom de la propriété. Pour plusieurs hôtes, le paramètre *Groupes d'hôtes* doit soit ne pas être configuré, soit être configuré avec au moins un groupe d'hôtes auquel appartiennent les hôtes configurés. Ce paramètre n'est pas pris en charge lors de la configuration du widget sur un tableau de bord de modèle.
Hôtes (widget/tableau de bord)	1	hostids._reference	Au lieu de l'ID de l'hôte : `DASHBOARD.hostids` - définissez le sélecteur d'hôtes du tableau de bord comme source de données des hôtes ; `ABCDE._hostids` - définissez un widget compatible (avec son paramètre *Reference* défini sur "`ABCDE`") comme source de données des hôtes. Ce paramètre n'est pas pris en charge lors de la configuration du widget sur un tableau de bord de modèle.
Étiquettes			
Type d'évaluation	0	evaltype	0 - *(par défaut)* Et/Ou ; 2 - Ou. Ce paramètre n'est pas pris en charge lors de la configuration du widget sur un tableau de bord de modèle.

Parameter	type	name	value
Nom de l'étiquette	1	tags.0.tag	Toute valeur de chaîne. Remarque : le numéro dans le nom de la propriété fait référence à l'ordre de l'étiquette dans la liste d'évaluation des étiquettes. Comportement du paramètre : - *obligatoire* lors de la configuration des *Étiquettes* Ce paramètre n'est pas pris en charge lors de la configuration du widget sur un tableau de bord de modèle.
Opérateur	0	tags.0.operator	0 - Contient ; 1 - Est égal à ; 2 - Ne contient pas ; 3 - N'est pas égal à ; 4 - Existe ; 5 - N'existe pas. Remarque : le numéro dans le nom de la propriété fait référence à l'ordre de l'étiquette dans la liste d'évaluation des étiquettes. Comportement du paramètre : - *obligatoire* lors de la configuration des *Étiquettes* Ce paramètre n'est pas pris en charge lors de la configuration du widget sur un tableau de bord de modèle.
Valeur de l'étiquette	1	tags.0.value	Toute valeur de chaîne. Remarque : le numéro dans le nom de la propriété fait référence à l'ordre de l'étiquette dans la liste d'évaluation des étiquettes. Comportement du paramètre : - *obligatoire* lors de la configuration des *Étiquettes* Ce paramètre n'est pas pris en charge lors de la configuration du widget sur un tableau de bord de modèle.
Vue initiale Regroupement	1	default_view	*latitude*, *longitude*, *niveau de zoom* séparés par des virgules (*facultatif*, les valeurs possibles vont de 0 à 30). Exemple : `40.6892494,-74.0466891,10`.
Mode de regroupement	0	clustering_mode	0 - *(par défaut)* Automatique ; 1 - Niveau de zoom.
Niveau de zoom du regroupement	0	clustering_zoom_level	Les valeurs possibles vont de 0 à 30. Valeur par défaut : 0. Comportement du paramètre : - *pris en charge* si le *Mode de regroupement* est défini sur « Niveau de zoom »
Référence	1	reference	Toute valeur de chaîne composée de 5 caractères (par exemple, `ABCDE` ou `JBPNL`). Cette valeur doit être unique au sein du tableau de bord auquel appartient le widget. Comportement du paramètre : - *obligatoire*

Exemples

Les exemples suivants visent uniquement à décrire la configuration des objets de champ du widget de tableau de bord pour le widget *Geomap*. Pour plus d'informations sur la configuration d'un tableau de bord, voir `dashboard.create`.

Configuration d'un widget *Geomap*

Configurez un widget *Geomap* qui affiche les hôtes des groupes d'hôtes « 2 » et « 22 » en fonction de la configuration de tags suivante : le tag nommé « component » contient la valeur « node », et le tag nommé « location » est égal à la valeur « New York ». De plus, définissez la vue initiale de la carte sur les coordonnées « 40.6892494 » (latitude), « -74.0466891 » (longitude) avec le niveau de zoom « 10 ».

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": 30,
        "auto_start": 1,
        "pages": [
            {
                "widgets": [
                    {
                        "type": "geomap",
                        "name": "Geomap",
                        "x": 0,
                        "y": 0,
                        "width": 36,
                        "height": 5,
                        "view_mode": 0,
                        "fields": [
                            {
                                "type": 2,
                                "name": "groupids.0",
                                "value": 22
                            },
                            {
                                "type": 2,
                                "name": "groupids.1",
                                "value": 2
                            },
                            {
                                "type": 1,
                                "name": "default_view",
                                "value": "40.6892494,-74.0466891,10"
                            },
                            {
                                "type": 0,
                                "name": "evaltype",
                                "value": 2
                            },
                            {
                                "type": 1,
                                "name": "tags.0.tag",
                                "value": "component"
                            },
                            {
                                "type": 0,
                                "name": "tags.0.operator",
                                "value": 0
                            },
                            {
                                "type": 1,
                                "name": "tags.0.value",
                                "value": "node"
                            },
                            {
                                "type": 1,
                                "name": "tags.1.tag",
```

```
                                "value": "location"
                            },
                            {
                                "type": 0,
                                "name": "tags.1.operator",
                                "value": 1
                            },
                            {
                                "type": 1,
                                "name": "tags.1.value",
                                "value": "New York"
                            }
                        ]
                    }
                ]
            }
        ],
        "userGroups": [
            {
                "usrgrpid": 7,
                "permission": 2
            }
        ],
        "users": [
            {
                "userid": 1,
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Voir également

- Champ du widget du tableau de bord
- `dashboard.create`
- `dashboard.update`

8 Graphique

Description

Ces paramètres et les valeurs de propriété possibles pour les objets de champ respectifs du widget du tableau de bord vous permettent de configurer le widget *Graphique* dans les méthodes `dashboard.create` et `dashboard.update`.

Attention:
Les propriétés `fields` du widget ne sont pas validées lors de la création ou de la mise à jour d'un tableau de bord. Cela permet aux utilisateurs de modifier les widgets intégrés et de créer des widgets personnalisés, mais introduit également le risque de créer ou de mettre à jour des widgets de manière incorrecte. Pour garantir la création ou la mise à jour réussie du widget *Graphique*, veuillez vous reporter au comportement des paramètres décrit dans les tableaux ci-dessous.

Paramètres

Les paramètres suivants sont supportés pour le widget *Graph*

Paramètre	type	nom	valeur
Interval de mise à jour	0	rf_rate	0 - Pas de mise à jour; 10 - 10 secondes; 30 - 30 secondes; 60 - *(defaul)* 1 minute; 120 - 2 minutes; 600 - 10 minutes; 900 - 15 minutes.
Réference	1	reference	Toute valeur de chaîne composée de 5 caractères (exemple : `ABCDE` ou `JBPNL`). Cette valeur doit être unique au sein du tableau de bord auquel appartient le widget. Comportement des paramètres: - *requis*

Ensemble de données

Les paramètres suivants sont pris en charge pour configurer un *Ensemble de données*.

Note:
Le premier nombre dans le nom de la propriété (par exemple, ds.0.hosts.0, ds.0.items.0) représente l'ensemble de données concerné, tandis que le deuxième nombre, s'il est présent, représente le hôte ou l'élément configuré.

Paramètre	type	name	value
Type d'ensemble de données	0	ds.0.dataset_type	0 - Liste d'éléments ; 1 - *(par défaut)* Modèles d'éléments.
Éléments	4	ds.0.itemids.0	ID de l'élément. Lors de la configuration du widget sur un tableau de bord de modèle, seuls les éléments configurés sur le modèle doivent être définis. Remarque : Pour configurer plusieurs éléments, créez un objet de champ de widget de tableau de bord pour chaque élément, avec un nombre incrémenté dans le nom de la propriété. Comportement du paramètre : - *obligatoire* si le *Type d'ensemble de données* est défini sur « Liste d'éléments » et si *Éléments (widget)* n'est pas défini

Paramètre	type	name	value
Éléments (widget)	1	ds.0.itemids.0._reference	À la place de l'ID de l'élément : `ABCDE._itemid` - définissez un widget compatible (avec son paramètre *Référence* défini sur « `ABCDE` ») comme source de données des éléments. Remarque : Pour configurer plusieurs widgets, créez un objet de champ de widget de tableau de bord pour chaque widget, avec un nombre incrémenté dans le nom de la propriété. Comportement du paramètre : - *obligatoire* si le *Type d'ensemble de données* est défini sur « Liste d'éléments » et si *Éléments* n'est pas défini
Couleur	1	ds.0.color.0	Code couleur hexadécimal (par exemple, `FF0000`). Comportement du paramètre : - *obligatoire* si le *Type d'ensemble de données* est défini sur « Liste d'éléments »
Modèles de hôtes	1	ds.0.hosts.0	Nom ou modèle de hôte (par exemple, « Zabbix* »). Comportement du paramètre : - *obligatoire* si le *Type d'ensemble de données* est défini sur « Modèles d'éléments » Ce paramètre n'est pas pris en charge si le widget est configuré sur un tableau de bord de modèle.
Modèles d'éléments	1	ds.0.items.0	Nom ou modèle de l'élément (par exemple, « * : Nombre de *valeurs traitées par seconde »). Lors de la configuration du widget sur un tableau de bord de modèle, seuls les modèles correspondant aux éléments configurés sur le modèle doivent être définis. Comportement du paramètre : - *obligatoire* si le *Type d'ensemble de données* est défini sur « Modèles d'éléments »
Couleur	1	ds.0.color	Code couleur hexadécimal (par exemple, `FF0000`). Comportement du paramètre : - *pris en charge* si le *Type d'ensemble de données* est défini sur « Modèles d'éléments » et si *Palette de couleurs* n'est pas définie
Palette de couleurs	0	ds.0.color_palette	Index de la palette de couleurs. Les valeurs possibles vont de 0 à 11. Valeur par défaut : 0. Comportement du paramètre : - *pris en charge* si le *Type d'ensemble de données* est défini sur « Modèles d'éléments » et si *Couleur* n'est pas définie
Tracé	0	ds.0.type	0 - *(par défaut)* Ligne ; 1 - Points ; 2 - Escalier ; 3 - Barres.
Empilé	0	ds.0.stacked	0 - *(par défaut)* Désactivé ; 1 - Activé. Comportement du paramètre : - *pris en charge* si *Tracé* est défini sur « Ligne », « Escalier » ou « Barres »

Paramètre	type	name	value
Largeur	0	ds.0.width	Les valeurs possibles vont de 1 à 10. Valeur par défaut : 1. Comportement du paramètre : - *pris en charge* si *Tracé* est défini sur « Ligne » ou « Escalier »
Taille des points	0	ds.0.pointsize	Les valeurs possibles vont de 1 à 10. Valeur par défaut : 3. Comportement du paramètre : - *pris en charge* si *Tracé* est défini sur « Points »
Transparence	0	ds.0.transparency	Les valeurs possibles vont de 1 à 10. Valeur par défaut : 5.
Remplissage	0	ds.0.fill	Les valeurs possibles vont de 1 à 10. Valeur par défaut : 3. Comportement du paramètre : - *pris en charge* si *Tracé* est défini sur « Ligne » ou « Escalier »
Données manquantes	0	ds.0.missingdatafunc	0 - *(par défaut)* Aucune ; 1 - Connectées ; 2 - Traitées comme 0 ; 3 - Dernière valeur connue. Comportement du paramètre : - *pris en charge* si *Tracé* est défini sur « Ligne » ou « Escalier »
Remplacer le hôte	1	ds.0.override_hostid.ABCDE_reference_hostid	`ABCDE` - définissez un widget compatible (avec son paramètre « Référence » défini sur « `ABCDE` ») comme source de données des hôtes ; `DASHBOARD._hostid` - définissez le sélecteur de hôtes du tableau de bord comme source de données des hôtes. Ce paramètre n'est pas pris en charge si le widget est configuré sur un tableau de bord de modèle.
Axe Y	0	ds.0.axisy	0 - *(par défaut)* Gauche ; 1 - Droite.
Inverser les valeurs	0	ds.0.invert_values	0 - (par défaut) Désactivé ; 1 - Activé.
Décalage temporel	1	ds.0.timeshift	Chaîne temporelle valide (par exemple, 3600, `1h`, etc.). Vous pouvez utiliser les suffixes temporels. Les valeurs négatives sont également autorisées. Valeur par défaut : `""` (vide).
Fonction d'agrégation	0	ds.0.aggregate_function	0 - *(par défaut)* non utilisée ; 1 - min ; 2 - max ; 3 - avg ; 4 - count ; 5 - sum ; 6 - first ; 7 - last.
Intervalle d'agrégation	1	ds.0.aggregate_interval	Chaîne temporelle valide (par exemple, 3600, `1h`, etc.). Vous pouvez utiliser les suffixes temporels. Valeur par défaut : `1h`.

Paramètre	type	name	value
Agrégat	0	ds.0.aggregate_grouping	0 - *(par défaut)* Chaque élément ; 1 - Ensemble de données. Comportement du paramètre : - *pris en charge* si la *Fonction d'agrégation* est définie sur « min », « max », « avg », « count », « sum », « first » ou « last »
Approximation	0	ds.0.approximation	1 - min ; 2 - *(par défaut)* avg ; 4 - max ; 7 - toutes.
Libellé de l'ensemble de données	1	ds.0.data_set_label	Toute valeur de chaîne. Valeur par défaut : "" (vide).
Étiquettes des éléments			
Type d'évaluation	0	ds.0.item_tags_evaltype	0 - *(par défaut)* Et/Ou ; 2 - Ou.
Nom de l'étiquette	1	ds.0.item_tags.0.tag	Toute valeur de chaîne. Remarque : Le nombre dans le nom de la propriété fait référence à l'ordre de l'étiquette dans la liste d'évaluation des étiquettes. Comportement du paramètre : - *obligatoire* lors de la configuration des *Étiquettes des éléments*
Opérateur	0	ds.0.item_tags.0.operator	0 - Contient ; 1 - Est égal à ; 2 - Ne contient pas ; 3 - N'est pas égal à ; 4 - Existe ; 5 - N'existe pas. Remarque : Le nombre dans le nom de la propriété fait référence à l'ordre de l'étiquette dans la liste d'évaluation des étiquettes. Comportement du paramètre : - *obligatoire* lors de la configuration des *Étiquettes des éléments*
Valeur de l'étiquette	1	ds.0.item_tags.0.value	Toute valeur de chaîne. Remarque : Le nombre dans le nom de la propriété fait référence à l'ordre de l'étiquette dans la liste d'évaluation des étiquettes. Comportement du paramètre : - *obligatoire* lors de la configuration des *Étiquettes des éléments*

Options d'affichage

Les paramètres suivants sont pris en charge pour configurer les *Options d'affichage*.

Parameter	type	name	value
Sélection des données d'historique	0	source	0 - *(par défaut)* Automatique ; 1 - Historique ; 2 - Tendances.

Parameter	type	name	value
Déclencheurs simples	0	simple_triggers	0 - *(par défaut)* Désactivé ; 1 - Activé.
Temps de fonctionnement	0	working_time	0 - *(par défaut)* Désactivé ; 1 - Activé.
Noms des hôtes dans les libellés	0	show_hostnames	0 - *(par défaut)* Automatique ; 1 - Afficher ; 2 - Masquer. Ce paramètre n'est pas pris en charge lors de la configuration du widget sur un tableau de bord de modèle.
Ligne de percentile (gauche)			
État	0	percentile_left	0 - *(par défaut)* Désactivé ; 1 - Activé. Comportement du paramètre : - *pris en charge* si l'*axe Y* (dans la configuration de l'*ensemble de données*) est défini sur « Gauche »
Valeur	0	percentile_left_value	Les valeurs possibles sont comprises entre 1 et 100. Comportement du paramètre : - *pris en charge* si l'*axe Y* (dans la configuration de l'*ensemble de données*) est défini sur « Gauche »
Ligne de percentile (droite)			
État	0	percentile_right	0 - *(par défaut)* Désactivé ; 1 - Activé. Comportement du paramètre : - *pris en charge* si l'*axe Y* (dans la configuration de l'*ensemble de données*) est défini sur « Droite »
Valeur	0	percentile_right_value	Les valeurs possibles sont comprises entre 1 et 100. Comportement du paramètre : - *pris en charge* si l'*axe Y* (dans la configuration de l'*ensemble de données*) est défini sur « Droite »

Période de temps

Les paramètres suivants sont pris en charge pour configurer la *Période de temps*.

Paramètre	type	name	value
Période de temps	1	time_period._reference	`DASHBOARD._timeperiod` - définit le sélecteur de période de temps du tableau de bord comme source de données ; `ABCDE._timeperiod` - définit un widget compatible (avec son paramètre *Reference* défini sur "`ABCDE`") comme source de données. Par défaut : `DASHBOARD._timeperiod` Vous pouvez également définir la période de temps uniquement dans les paramètres *From* et *To*.
From	1	time_period.from	Chaîne de temps valide en syntaxe temporelle absolue (`YYYY-MM-DD hh:mm:ss`) ou relative (`now`, `now/d`, `now/w-1w`, etc.). Comportement du paramètre : - *pris en charge* si *Période de temps* n'est pas défini - *obligatoire* si `time_period.to` est défini
To	1	time_period.to	Chaîne de temps valide en syntaxe temporelle absolue (`YYYY-MM-DD hh:mm:ss`) ou relative (`now`, `now/d`, `now/w-1w`, etc.). Comportement du paramètre : - *pris en charge* si *Période de temps* n'est pas défini - *obligatoire* si `time_period.from` est défini

Axes

Les paramètres suivants sont pris en charge pour configurer les *Axes*.

Paramètre	type	name	value
Y gauche	0	lefty	0 - Désactivé; 1 - *(par défaut)* Activé. Comportement du paramètre : - *pris en charge* si *Axe Y* (dans la configuration de *Jeu de données*) est défini sur « Gauche »
Y droite	0	righty	0 - *(par défaut)* Désactivé; 1 - Activé. Comportement du paramètre : - *pris en charge* si *Axe Y* (dans la configuration de *Jeu de données*) est défini sur « Droite »
Échelle	0	lefty_scale righty_scale	0 - *(par défaut)* Linéaire; 1 - Logarithmique.
Min	1	lefty_min righty_min	Toute valeur numérique. Par défaut : "" (vide).
Max	1	lefty_max righty_max	Toute valeur numérique. Par défaut : "" (vide).
Unités (type)	0	lefty_units righty_units	0 - *(par défaut)* Auto; 1 - Statique.
Unités (valeur)	1	lefty_static_units righty_static_units	Toute valeur de chaîne. Par défaut : "" (vide).

Paramètre	type	name	value
Axe X	0	xaxis	0 - Désactivé; 1 - *(par défaut)* Activé.

Légende

Les paramètres suivants sont pris en charge pour configurer la *Légende*.

Paramètre	type	name	value
Afficher la légende	0	legend	0 - Désactivé ; 1 - *(par défaut)* Activé.
Afficher le min/moy/max	0	legend_statistic	0 - *(par défaut)* Désactivé ; 1 - Activé. Comportement du paramètre : - *pris en charge* si *Afficher la légende* est défini sur « Activé »
Afficher la fonction d'agrégation	0	legend_aggregation	0 - *(par défaut)* Désactivé ; 1 - Activé. Comportement du paramètre : - *pris en charge* si *Afficher la légende* est défini sur « Activé »
Lignes	0	legend_lines_mode	0 - *(par défaut)* Fixe ; 1 - Variable. Comportement du paramètre : - *pris en charge* si *Afficher la légende* est défini sur « Activé »
Nombre de lignes/ Nombre maximal de lignes	0	legend_lines	Les valeurs possibles vont de 1 à 10. Valeur par défaut : 1. Comportement du paramètre : - *pris en charge* si *Afficher la légende* est défini sur « Activé »
Nombre de colonnes	0	legend_columns	Les valeurs possibles vont de 1 à 4. Valeur par défaut : 4. Comportement du paramètre : - *pris en charge* si *Afficher la légende* est défini sur « Activé » et *Afficher le min/moy/max* est défini sur « Désactivé »

Problèmes

Les paramètres suivants sont pris en charge pour configurer les *Problèmes*.

Paramètre	type	name	value
Afficher les problèmes	0	show_problems	0 - *(par défaut)* Désactivé ; 1 - Activé.
Éléments sélectionnés uniquement	0	graph_item_problems	0 - Désactivé ; 1 - *(par défaut)* Activé.

Paramètre	type	name	value
Hôtes des prob-lèmes	1	problemhosts.0	Nom de l'hôte. Remarque : Le nombre dans le nom de propriété fait référence à l'hôte configuré. Pour configurer plusieurs hôtes, créez un objet de champ de widget de tableau de bord pour chaque hôte avec un nombre incrémenté dans le nom de propriété. Ce paramètre n'est pas pris en charge lors de la configuration du widget sur un tableau de bord de modèle.
Sévérité	0	severities.0	0 - Non classé ; 1 - Information ; 2 - Avertissement ; 3 - Moyen ; 4 - Élevé ; 5 - Désastre. Par défaut : vide (tous activés). Remarque : Pour configurer plusieurs valeurs, créez un objet de champ de widget de tableau de bord pour chaque valeur avec un nombre incrémenté dans le nom de propriété.
Problème	1	problem_name	Nom d'événement du problème (insensible à la casse, nom complet ou partie de celui-ci).
Balises de prob-lème			
Type d'évaluation	0	evaltype	0 - *(par défaut)* Et/Ou ; 2 - Ou.
Nom de balise	1	tags.0.tag	Toute valeur de chaîne. Remarque : Le nombre dans le nom de propriété fait référence à l'ordre des balises dans la liste d'évaluation des balises. Comportement du paramètre : - *obligatoire* lors de la configuration des *Balises de problème*
Opérateur	0	tags.0.operator	0 - Contient ; 1 - Égal à ; 2 - Ne contient pas ; 3 - N'est pas égal à ; 4 - Existe ; 5 - N'existe pas. Remarque : Le nombre dans le nom de propriété fait référence à l'ordre des balises dans la liste d'évaluation des balises. Comportement du paramètre : - *obligatoire* lors de la configuration des *Balises de problème*
Valeur de balise	1	tags.0.value	Toute valeur de chaîne. Remarque : Le nombre dans le nom de propriété fait référence à l'ordre des balises dans la liste d'évaluation des balises. Comportement du paramètre : - *obligatoire* lors de la configuration des *Balises de problème*

Remplacements

Les paramètres suivants sont pris en charge pour configurer les *Remplacements*.

Note:
Le premier nombre dans le nom de la propriété (par exemple, or.0.hosts.0, or.0.items.0) représente l'ensemble de données concerné, tandis que le deuxième nombre, s'il est présent, représente le hôte ou l'élément configuré.

Paramètre	type	name	value
Motifs de hôtes	1	or.0.hosts.0	Nom ou motif du hôte (par exemple, `Zabbix*`). Ce paramètre n'est pas pris en charge lors de la configuration du widget sur un tableau de bord de modèle. Comportement du paramètre : - *obligatoire* lors de la configuration des *Remplacements*
Motifs d'éléments	1	or.0.items.0	Nom ou motif de l'élément (par exemple, `*: Number of processed *values per second`). Lors de la configuration du widget sur un tableau de bord de modèle, seuls les motifs des éléments configurés sur le modèle doivent être définis. Comportement du paramètre : - *obligatoire* lors de la configuration des *Remplacements*
Couleur de base	1	or.0.color	Code couleur hexadécimal (par exemple, FF0000).
Largeur	0	or.0.width	Les valeurs possibles vont de 1 à 10.
Tracé	0	or.0.type	0 - Ligne ; 1 - Points ; 2 - Escalier ; 3 - Barres.
Transparence	0	or.0.transparency	Les valeurs possibles vont de 1 à 10.
Remplissage	0	or.0.fill	Les valeurs possibles vont de 1 à 10.
Taille des points	0	or.0.pointsize	Les valeurs possibles vont de 1 à 10.
Données manquantes	0	or.0.missingdatafunc	0 - Aucune ; 1 - Connectées ; 2 - Traitées comme 0 ; 3 - Dernière valeur connue.
Axe Y	0	or.0.axisy	0 - Gauche ; 1 - Droite.
Inverser les valeurs	0	or.0.invert_values	0 - (par défaut) Désactivé ; 1 - Activé.
Décalage temporel	1	or.0.timeshift	Chaîne temporelle valide (par exemple, 3600, 1h, etc.). Vous pouvez utiliser les suffixes temporels. Les valeurs négatives sont autorisées.

Exemples

Les exemples suivants visent à décrire uniquement la configuration des objets de champ du widget de tableau de bord pour le widget *Graphique*. Pour plus d'informations sur la configuration d'un tableau de bord, consultez `dashboard.create`.

Configuration d'un widget *Graph*

Configurez un widget *Graph* de la manière suivante :

- 2 ensembles de données pour un total de 9 éléments sur 1 hôte.
- Le premier ensemble de données est de type « Item list » et se compose de 3 éléments représentés par des lignes de couleur différente, mais avec la même largeur, la même transparence et le même remplissage.
- Le deuxième ensemble de données est de type « Item patterns », se compose de 6 éléments, possède une agrégation configurée et est représenté par une ligne avec une couleur, une largeur, une transparence et un remplissage personnalisés.
- Le deuxième ensemble de données possède également un libellé personnalisé.
- Les données du graphique sont affichées pour une période correspondant aux 3 dernières heures.
- Les problèmes dans le graphique sont affichés uniquement pour les éléments configurés.
- Le graphique possède deux axes Y, dont l'axe Y droit affiche des valeurs uniquement pour le deuxième ensemble de données.
- La légende du graphique affiche les éléments configurés sur 4 lignes, ainsi que les valeurs minimale, maximale et moyenne des ensembles de données.

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": 30,
        "auto_start": 1,
        "pages": [
            {
                "widgets": [
                    {
                        "type": "svggraph",
                        "name": "Graph",
                        "x": 0,
                        "y": 0,
                        "width": 36,
                        "height": 5,
                        "view_mode": 0,
                        "fields": [
                            {
                                "type": 0,
                                "name": "ds.0.dataset_type",
                                "value": 0
                            },
                            {
                                "type": 4,
                                "name": "ds.0.itemids.0",
                                "value": 23264
                            },
                            {
                                "type": 1,
                                "name": "ds.0.color.0",
                                "value": "FF0000"
                            },
                            {
                                "type": 4,
                                "name": "ds.0.itemids.1",
                                "value": 23269
                            },
                            {
                                "type": 1,
                                "name": "ds.0.color.1",
                                "value": "BF00FF"
                            },
                            {
                                "type": 4,
                                "name": "ds.0.itemids.2",
                                "value": 23257
                            },
                            {
                                "type": 1,
                                "name": "ds.0.color.2",
                                "value": "0040FF"
                            },
                            {
                                "type": 0,
                                "name": "ds.0.width",
                                "value": 3
                            },
                            {
                                "type": 0,
                                "name": "ds.0.transparency",
```

```
            "value": 3
        },
        {
            "type": 0,
            "name": "ds.0.fill",
            "value": 1
        },
        {
            "type": 1,
            "name": "ds.1.hosts.0",
            "value": "Zabbix server"
        },
        {
            "type": 1,
            "name": "ds.1.items.0",
            "value": "*: Number of processed *values per second"
        },
        {
            "type": 1,
            "name": "ds.1.color",
            "value": "000000"
        },
        {
            "type": 0,
            "name": "ds.1.transparency",
            "value": 0
        },
        {
            "type": 0,
            "name": "ds.1.fill",
            "value": 0
        },
        {
            "type": 0,
            "name": "ds.1.axisy",
            "value": 1
        },
        {
            "type": 0,
            "name": "ds.1.aggregate_function",
            "value": 3
        },
        {
            "type": 1,
            "name": "ds.1.aggregate_interval",
            "value": "1m"
        },
        {
            "type": 0,
            "name": "ds.1.aggregate_grouping",
            "value": 1
        },
        {
            "type": 1,
            "name": "ds.1.data_set_label",
            "value": "Number of processed values per second"
        },
        {
            "type": 0,
            "name": "graph_time",
            "value": 1
        },
```

```
                            {
                                "type": 1,
                                "name": "time_period.from",
                                "value": "now-3h"
                            },
                            {
                                "type": 1,
                                "name": "time_period.to",
                                "value": "now"
                            },
                            {
                                "type": 0,
                                "name": "legend_statistic",
                                "value": 1
                            },
                            {
                                "type": 0,
                                "name": "legend_lines",
                                "value": 4
                            },
                            {
                                "type": 0,
                                "name": "show_problems",
                                "value": 1
                            },
                            {
                                "type": 1,
                                "name": "reference",
                                "value": "YZABC"
                            }
                        ]
                    }
                ]
            }
        ],
        "userGroups": [
            {
                "usrgrpid": 7,
                "permission": 2
            }
        ],
        "users": [
            {
                "userid": 1,
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Voir également

- Champ du widget du tableau de bord
- `dashboard.create`
- `dashboard.update`

9 Graphique (classique)

Description

Ces paramètres et les valeurs de propriété possibles pour les objets de champ respectifs du widget de tableau de bord vous permettent de configurer le widget *Graphique (classique)* dans les méthodes `dashboard.create` et `dashboard.update`.

Attention:
Les propriétés `fields` du widget ne sont pas validées lors de la création ou de la mise à jour d'un tableau de bord. Cela permet aux utilisateurs de modifier les widgets intégrés et de créer des widgets personnalisés, mais introduit également le risque de créer ou de mettre à jour des widgets de manière incorrecte. Pour garantir la réussite de la création ou de la mise à jour du widget *Graphique (classique)*, veuillez vous reporter au comportement des paramètres décrit dans les tableaux ci-dessous.

Paramètres

Les paramètres suivants sont pris en charge pour le widget *Graphique (classique)*.

Paramètre	type	name	value
Intervalle d'actualisation	0	rf_rate	0 - Aucune actualisation; 10 - 10 secondes; 30 - 30 secondes; 60 - *(par défaut)* 1 minute; 120 - 2 minutes; 600 - 10 minutes; 900 - 15 minutes.
Source	0	source_type	0 - *(par défaut)* Graphique; 1 - Graphique simple.
Graphique	6	graphid.0	ID de graphique. Comportement du paramètre: - *obligatoire* si *Source* est défini sur "Graphique"
Graphique (widget)	1	graphid._reference	Au lieu de l'ID de graphique : `ABCDE._graphid` - définir un widget compatible (avec son paramètre *Référence* défini sur "`ABCDE`") comme source de données pour les graphiques. Comportement du paramètre: - *obligatoire* si *Source* est défini sur "Graphique simple" et que *Graphique* n'est pas défini
Élément	4	itemid.0	ID d'élément. Comportement du paramètre: - *obligatoire* si *Source* est défini sur "Graphique simple" et que *Élément (widget)* n'est pas défini
Élément (widget)	1	itemid._reference	Au lieu de l'ID d'élément : `ABCDE._itemid` - définir un widget compatible (avec son paramètre *Référence* défini sur "`ABCDE`") comme source de données pour les éléments. Comportement du paramètre: - *obligatoire* si *Source* est défini sur "Graphique simple" et que *Élément* n'est pas défini

Paramètre	type	name	value
Période	1	time_period._referenceDASHBOARD._timeperiod	DASHBOARD._timeperiod - définir le sélecteur de période du tableau de bord comme source de données; ABCDE._timeperiod - définir un widget compatible (avec son paramètre *Référence* défini sur "ABCDE") comme source de données. Par défaut : DASHBOARD._timeperiod Vous pouvez également définir la période uniquement dans les paramètres *De* et *À*.
De	1	time_period.from	Chaîne temporelle valide en syntaxe temporelle absolue (YYYY-MM-DD hh:mm:ss) ou relative (now, now/d, now/w-1w, etc.). Comportement du paramètre: - *pris en charge* si *Période* n'est pas défini - *obligatoire* si time_period.to est défini
À	1	time_period.to	Chaîne temporelle valide en syntaxe temporelle absolue (YYYY-MM-DD hh:mm:ss) ou relative (now, now/d, now/w-1w, etc.). Comportement du paramètre: - *pris en charge* si *Période* n'est pas défini - *obligatoire* si time_period.from est défini
Afficher la légende	0	show_legend	0 - Désactivé; 1 - *(par défaut)* Activé.
Remplacer l'hôte	1	override_hostid._referenceABCDE._hostid	ABCDE._hostid - définir un widget compatible (avec son paramètre *Référence* défini sur "ABCDE") comme source de données pour les hôtes; DASHBOARD._hostid - définir le sélecteur d'hôte du tableau de bord comme source de données pour les hôtes. Ce paramètre n'est pas pris en charge lors de la configuration du widget sur un tableau de bord de modèle.
Référence	1	reference	Toute valeur de chaîne composée de 5 caractères (par exemple, ABCDE ou JBPNL). Cette valeur doit être unique dans le tableau de bord auquel le widget appartient. Comportement du paramètre: - *obligatoire*

Exemples

Les exemples suivants ont pour objectif de décrire uniquement la configuration des objets champs du widget de tableau de bord pour le widget *Graphique (classique)*. Pour plus d'informations sur la configuration d'un tableau de bord, consultez dashboard.create.

Configuration d'un widget *Graph (classic)*

Configurez un widget *Graph (classic)* qui affiche un graphique simple pour l'élément « 42269 ».

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": 30,
        "auto_start": 1,
        "pages": [
```

```
            {
                "widgets": [
                    {
                        "type": "graph",
                        "name": "Graph (classic)",
                        "x": 0,
                        "y": 0,
                        "width": 36,
                        "height": 5,
                        "view_mode": 0,
                        "fields": [
                            {
                                "type": 0,
                                "name": "source_type",
                                "value": 1
                            },
                            {
                                "type": 4,
                                "name": "itemid.0",
                                "value": 42269
                            },
                            {
                                "type": 1,
                                "name": "reference",
                                "value": "RSTUV"
                            }
                        ]
                    }
                ]
            }
        ],
        "userGroups": [
            {
                "usrgrpid": 7,
                "permission": 2
            }
        ],
        "users": [
            {
                "userid": 1,
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Voir également

- Champ du widget du tableau de bord
- `dashboard.create`
- `dashboard.update`

Description

Ces paramètres et les valeurs de propriété possibles pour les objets de champ respectifs du widget de tableau de bord vous permettent de configurer le widget *Graph prototype* dans les méthodes `dashboard.create` et `dashboard.update`.

Attention:
Les propriétés `fields` du widget ne sont pas validées lors de la création ou de la mise à jour d'un tableau de bord. Cela permet aux utilisateurs de modifier les widgets intégrés et de créer des widgets personnalisés, mais introduit également le risque de créer ou de mettre à jour des widgets de manière incorrecte. Pour garantir la création ou la mise à jour réussie du widget *Graph prototype*, veuillez vous reporter au comportement des paramètres décrit dans les tableaux ci-dessous.

Paramètres

Les paramètres suivants sont pris en charge pour le widget *Prototype de graphique*.

Paramètre	type	name	value
Intervalle d'actualisation	0	rf_rate	0 - Aucune actualisation ; 10 - 10 secondes ; 30 - 30 secondes ; 60 - *(par défaut)* 1 minute ; 120 - 2 minutes ; 600 - 10 minutes ; 900 - 15 minutes.
Source	0	source_type	2 - *(par défaut)* Prototype de graphique ; 3 - Prototype de graphique simple.
Prototype de graphique	7	graphid.0	ID du prototype de graphique. Comportement du paramètre : - *obligatoire* si *Source* est défini sur « Prototype de graphique »
Prototype d'élément	5	itemid.0	ID du prototype d'élément. Comportement du paramètre : - *obligatoire* si *Source* est défini sur « Prototype de graphique simple »
Période	1	time_period._reference	`DASHBOARD._timeperiod` - définir le sélecteur de période du tableau de bord comme source de données ; `ABCDE._timeperiod` - définir un widget compatible (avec son paramètre *Référence* défini sur « `ABCDE` ») comme source de données. Valeur par défaut : `DASHBOARD._timeperiod` Vous pouvez également définir la période uniquement dans les paramètres *Du* et *Au*.
Du	1	time_period.from	Chaîne temporelle valide au format absolu (`YYYY-MM-DD hh:mm:ss`) ou au format temporel relatif (`now`, `now/d`, `now/w-1w`, etc.). Comportement du paramètre : - *pris en charge* si *Période* n'est pas défini - *obligatoire* si `time_period.to` est défini
Au	1	time_period.to	Chaîne temporelle valide au format absolu (`YYYY-MM-DD hh:mm:ss`) ou au format temporel relatif (`now`, `now/d`, `now/w-1w`, etc.). Comportement du paramètre : - *pris en charge* si *Période* n'est pas défini - *obligatoire* si `time_period.from` est défini

Paramètre	type	name	value
Afficher la légende	0	show_legend	0 - Désactivé ; 1 - *(par défaut)* Activé.
Remplacer le hôte	1	override_hostid._reference	ABCDE._hostid - définir un widget compatible (avec son paramètre *Référence* défini sur « ABCDE ») comme source de données pour les hôtes ; DASHBOARD._hostid - définir le sélecteur d'hôte du tableau de bord comme source de données pour les hôtes. Ce paramètre n'est pas pris en charge lors de la configuration du widget sur un tableau de bord de modèle.
Colonnes	0	columns	Les valeurs possibles vont de 1 à 24. Valeur par défaut : 2.
Lignes	0	rows	Les valeurs possibles vont de 1 à 16. Valeur par défaut : 1.
Référence	1	reference	Toute valeur de chaîne composée de 5 caractères (par exemple, ABCDE ou JBPNL). Cette valeur doit être unique au sein du tableau de bord auquel appartient le widget. Comportement du paramètre : - *obligatoire*

Exemples

Les exemples suivants visent à décrire uniquement la configuration des objets de champ du widget de tableau de bord pour le widget *Prototype de graphique*. Pour plus d'informations sur la configuration d'un tableau de bord, consultez `dashboard.create`.

Configuration d'un widget *Prototype de graphique*

Configurez un widget *Prototype de graphique* qui affiche une grille de 3 graphiques (3 colonnes, 1 ligne) créés à partir d'un prototype d'élément (ID : "42316") par découverte de bas niveau.

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": 30,
        "auto_start": 1,
        "pages": [
            {
                "widgets": [
                    {
                        "type": "graphprototype",
                        "name": "Graph prototype",
                        "x": 0,
                        "y": 0,
                        "width": 48,
                        "height": 5,
                        "view_mode": 0,
                        "fields": [
                            {
                                "type": 0,
                                "name": "source_type",
                                "value": 3
                            },
                            {
                                "type": 5,
```

```
                                "name": "itemid.0",
                                "value": 42316
                            },
                            {
                                "type": 0,
                                "name": "columns",
                                "value": 3
                            },
                            {
                                "type": 1,
                                "name": "reference",
                                "value": "OPQWX"
                            }
                        ]
                    }
                ]
            }
        ],
        "userGroups": [
            {
                "usrgrpid": 7,
                "permission": 2
            }
        ],
        "users": [
            {
                "userid": 1,
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Voir également

- Champ du widget du tableau de bord
- `dashboard.create`
- `dashboard.update`

11 Honeycomb

Description

Ces paramètres et les valeurs de propriété possibles pour les objets de champ respectifs du widget de tableau de bord vous permettent de configurer le widget *Honeycomb* dans les méthodes `dashboard.create` et `dashboard.update`.

Attention:
Les propriétés `fields` du widget ne sont pas validées lors de la création ou de la mise à jour d'un tableau de bord. Cela permet aux utilisateurs de modifier les widgets intégrés et de créer des widgets personnalisés, mais introduit également le risque de créer ou de mettre à jour des widgets de manière incorrecte. Pour garantir la réussite de la création ou de la mise à jour du widget *Honeycomb*, veuillez vous reporter au comportement des paramètres décrit dans les tableaux ci-dessous.

Paramètres

Les paramètres suivants sont pris en charge pour le widget *Honeycomb*.

Paramètre	type	name	value
Intervalle d'actualisation	0	rf_rate	0 - Aucune actualisation; 10 - 10 secondes; 30 - 30 secondes; 60 - *(par défaut)* 1 minute; 120 - 2 minutes; 600 - 10 minutes; 900 - 15 minutes.
Groupes d'hôtes	2	groupids.0	ID du groupe d'hôtes. Remarque : Pour configurer plusieurs groupes d'hôtes, créez un objet de champ de widget de tableau de bord pour chaque groupe d'hôtes avec un numéro incrémenté dans le nom de la propriété. Ce paramètre n'est pas pris en charge lors de la configuration du widget sur un tableau de bord de modèle.
Groupes d'hôtes (Widget)	1	groupids._reference	Au lieu de l'ID du groupe d'hôtes : `ABCDE._hostgroupids` - définissez un widget compatible (avec son paramètre *Reference* défini sur "`ABCDE`") comme source de données pour les groupes d'hôtes. Ce paramètre n'est pas pris en charge lors de la configuration du widget sur un tableau de bord de modèle.
Hôtes	3	hostids.0	ID de l'hôte. Remarque : Pour configurer plusieurs hôtes, créez un objet de champ de widget de tableau de bord pour chaque hôte avec un numéro incrémenté dans le nom de la propriété. Pour plusieurs hôtes, le paramètre *Groupes d'hôtes* ne doit soit pas être configuré du tout, soit être configuré avec au moins un groupe d'hôtes auquel appartiennent les hôtes configurés. Ce paramètre n'est pas pris en charge lors de la configuration du widget sur un tableau de bord de modèle.
Hôtes (Widget/Tableau de bord)	1	hostids._reference	Au lieu de l'ID de l'hôte : `DASHBOARD.hostids` - définissez le sélecteur d'hôte du tableau de bord comme source de données pour les hôtes ; `ABCDE._hostids` - définissez un widget compatible (avec son paramètre *Reference* défini sur "`ABCDE`") comme source de données pour les hôtes. Ce paramètre n'est pas pris en charge lors de la configuration du widget sur un tableau de bord de modèle.
Tags d'hôte			
Type d'évaluation	0	evaltype_host	0 - *(par défaut)* Et/Ou; 2 - Ou. Ce paramètre n'est pas pris en charge lors de la configuration du widget sur un tableau de bord de modèle.

Paramètre	type	name	value
Nom du tag	1	host_tags.0.tag	Toute valeur de chaîne. Remarque : Le numéro dans le nom de la propriété fait référence à l'ordre du tag dans la liste d'évaluation des tags. Comportement du paramètre : - *obligatoire* lors de la configuration de *Tags d'hôte* Ce paramètre n'est pas pris en charge lors de la configuration du widget sur un tableau de bord de modèle.
Opérateur	0	host_tags.0.operator	0 - Contient; 1 - Égal à; 2 - Ne contient pas; 3 - Différent de; 4 - Existe; 5 - N'existe pas. Remarque : Le numéro dans le nom de la propriété fait référence à l'ordre du tag dans la liste d'évaluation des tags. Comportement du paramètre : - *obligatoire* lors de la configuration de *Tags d'hôte* Ce paramètre n'est pas pris en charge lors de la configuration du widget sur un tableau de bord de modèle.
Valeur du tag	1	host_tags.0.value	Toute valeur de chaîne. Remarque : Le numéro dans le nom de la propriété fait référence à l'ordre du tag dans la liste d'évaluation des tags. Comportement du paramètre : - *obligatoire* lors de la configuration de *Tags d'hôte* Ce paramètre n'est pas pris en charge lors de la configuration du widget sur un tableau de bord de modèle.
Modèles d'élément	1	items.0	Nom ou modèle de l'élément. Remarque : Pour configurer plusieurs modèles d'élément, créez un objet de champ de widget de tableau de bord pour chaque modèle d'élément avec un numéro incrémenté dans le nom de la propriété. Comportement du paramètre : - *obligatoire*
Tags d'élément			
Type d'évaluation	0	evaltype_item	0 - *(par défaut)* Et/Ou; 2 - Ou.
Nom du tag	1	item_tags.0.tag	Toute valeur de chaîne. Remarque : Le numéro dans le nom de la propriété fait référence à l'ordre du tag dans la liste d'évaluation des tags. Comportement du paramètre : - *obligatoire* lors de la configuration de *Tags d'élément*

Paramètre	type	name	value
Opérateur	0	item_tags.0.operator	0 - Contient; 1 - Égal à; 2 - Ne contient pas; 3 - Différent de; 4 - Existe; 5 - N'existe pas. Remarque : Le numéro dans le nom de la propriété fait référence à l'ordre du tag dans la liste d'évaluation des tags. Comportement du paramètre : - *obligatoire* lors de la configuration de *Tags d'élément*
Valeur du tag	1	item_tags.0.value	Toute valeur de chaîne. Remarque : Le numéro dans le nom de la propriété fait référence à l'ordre du tag dans la liste d'évaluation des tags. Comportement du paramètre : - *obligatoire* lors de la configuration de *Tags d'élément*
Afficher les hôtes en maintenance	0	maintenance	0 - *(par défaut)* Désactivé; 1 - Activé.
Afficher	0	show.0	1 - Étiquette principale; 2 - Étiquette secondaire. Remarque : Pour configurer plusieurs valeurs, créez un objet de champ de widget de tableau de bord pour chaque valeur avec un numéro incrémenté dans le nom de la propriété. Par défaut : 1, 2.
Reference	1	reference	Toute valeur de chaîne composée de 5 caractères (par exemple, `ABCDE` ou `JBPNL`). Cette valeur doit être unique dans le tableau de bord auquel appartient le widget. Comportement du paramètre : - *obligatoire*

Configuration avancée

Les paramètres de configuration avancée suivants sont pris en charge pour le widget *Honeycomb*.

Note:
Le nombre dans le nom de la propriété *Seuils* (par exemple, thresholds.0.color) fait référence à la position du seuil dans une liste triée par ordre croissant. Cependant, si les seuils sont configurés dans un ordre différent, les valeurs seront triées par ordre croissant après la mise à jour de la configuration du widget dans l'interface Zabbix (par exemple, `"thresholds.0.threshold":"5"` → `"thresholds.0.threshold":"1"` ; `"thresholds.1.threshold":"1"` → `"thresholds.1.threshold": "5"`).

Paramètre	type	name	value
Libellé principal			
Type	0	primary_label_type	0 - *(par défaut)* Texte ; 1 - Valeur.

Paramètre	type	name	value
Texte	1	primary_label	Toute valeur de chaîne, y compris les macros. Macros prises en charge : {HOST.*}, {ITEM.*}, {INVENTORY.*}, macros utilisateur. Par défaut : {HOST.NAME} Comportement du paramètre : - *pris en charge* si *Type* est défini sur « Texte »
Décimales	0	primary_label_decimals	Les valeurs possibles vont de 0 à 6. Par défaut : 2. Comportement du paramètre : - *pris en charge* si *Type* est défini sur « Valeur »
Taille (type)	0	primary_label_size_type	0 - *(par défaut)* Automatique ; 1 - Personnalisée.
Taille	0	primary_label_size	Les valeurs possibles vont de 1 à 100. Par défaut : 20. Comportement du paramètre : - *pris en charge* si *Taille* (type) est défini sur « Personnalisée »
Gras	0	primary_label_bold	0 - Désactivé ; 1 - Activé.
Couleur	1	primary_label_color	Code couleur hexadécimal (par exemple, FF0000). Par défaut : basé sur theme de l'objet Settings et de l'objet User : 1F2C33 pour « blue-theme », « blue-classic-theme » ou « hc-light » ; EEEEEE pour « dark-theme », « dark-classic-theme » ou « hc-dark » ; E5F1FF pour « dark-blue-theme ».
Unités (case à cocher)	0	primary_label_units_show	0 - Désactivées ; 1 - *(par défaut)* Activées. Comportement du paramètre : - *pris en charge* si *Type* est défini sur « Valeur »
Unités (valeur)	1	primary_label_units	Toute valeur de chaîne. "" (vide) Comportement du paramètre : - *pris en charge* si *Type* est défini sur « Valeur » et si *Unités* (case à cocher) est défini sur « Activées »
Position	0	primary_label_units_pos	0 - Avant la valeur ; 1 - *(par défaut)* Après la valeur. Comportement du paramètre : - *pris en charge* si *Type* est défini sur « Valeur » et si *Unités* (case à cocher) est défini sur « Activées » Ce paramètre est ignoré s'il est défini sur l'une des unités liées au temps suivantes : unixtime, uptime, s.
Libellé secondaire			
Type	0	secondary_label_type	0 - Texte ; 1 - *(par défaut)* Valeur.

Paramètre	type	name	value
Texte	1	secondary_label	Toute valeur de chaîne, y compris les macros. Macros prises en charge : {HOST.*}, {ITEM.*}, {INVENTORY.*}, macros utilisateur. Par défaut : {{ITEM.LASTVALUE}.fmtnum(2)} Comportement du paramètre : - *pris en charge* si *Type* est défini sur « Texte »
Décimales	0	secondary_label_decimal_places	Les valeurs possibles vont de 0 à 6. Par défaut : 2. Comportement du paramètre : - *pris en charge* si *Type* est défini sur « Valeur »
Taille (type)	0	secondary_label_size_type	0 - *(par défaut)* Automatique ; 1 - Personnalisée.
Taille	0	secondary_label_size	Les valeurs possibles vont de 1 à 100. Par défaut : 30. Comportement du paramètre : - *pris en charge* si *Taille* (type) est défini sur « Personnalisée »
Gras	0	secondary_label_bold	0 - Désactivé ; 1 - *(par défaut)* Activé.
Couleur	1	secondary_label_color	Code couleur hexadécimal (par exemple, FF0000). Par défaut : basé sur theme de l'objet Settings et de l'objet User : 1F2C33 pour « blue-theme », « blue-classic-theme » ou « hc-light » ; EEEEEE pour « dark-theme », « dark-classic-theme » ou « hc-dark » ; E5F1FF pour « dark-blue-theme ».
Unités (case à cocher)	0	secondary_label_units_show	0 - Désactivées ; 1 - *(par défaut)* Activées. Comportement du paramètre : - *pris en charge* si *Type* est défini sur « Valeur »
Unités (valeur)	1	secondary_label_units	Toute valeur de chaîne. "" (vide) Comportement du paramètre : - *pris en charge* si *Type* est défini sur « Valeur » et si *Unités* (case à cocher) est défini sur « Activées »
Position	0	secondary_label_position	0 - Avant la valeur ; 1 - *(par défaut)* Après la valeur. Comportement du paramètre : - *pris en charge* si *Type* est défini sur « Valeur » et si *Unités* (case à cocher) est défini sur « Activées » Ce paramètre est ignoré s'il est défini sur l'une des unités liées au temps suivantes : unixtime, uptime, s.
Couleur d'arrière-plan			
Couleur d'arrière-plan	1	bg_color	Code couleur hexadécimal (par exemple, FF0000). Par défaut : basé sur theme de l'objet Settings et de l'objet User : D9E7ED pour « blue-theme » et « blue-classic-theme » ; 3D5059 pour « dark-theme » et « dark-classic-theme » ; 004FA8 pour « dark-blue-theme » ; AAD7E9 pour « hc-light » ; 335463 pour « hc-dark ».

Paramètre	type	name	value
Seuils			
Interpolation des couleurs	0	interpolation	0 - Désactivée ; 1 - *(par défaut)* Activée.
Couleur	1	thresholds.0.color	Code couleur hexadécimal (par exemple, FF0000).
Seuil	1	thresholds.0.threshold	Toute valeur numérique. Les suffixes (par exemple, « 1d », « 2w », « 4K », « 8G ») sont pris en charge.

Exemples

Les exemples suivants visent uniquement à décrire la configuration des objets de champ de widget du widget *Honeycomb*. Pour plus d'informations sur la configuration d'un tableau de bord, consultez `dashboard.create`.

Configuration d'un widget *Honeycomb*

Configurez un widget *Honeycomb* qui affiche l'utilisation des processus du serveur Zabbix. De plus, modifiez l'étiquette principale des cellules du nid d'abeilles et ajustez visuellement le widget à l'aide de seuils.

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": "30",
        "auto_start": "1",
        "pages": [
            {
                "widgets": [
                    {
                        "type": "honeycomb",
                        "name": "Honeycomb",
                        "x": "0",
                        "y": "0",
                        "width": "24",
                        "height": "5",
                        "view_mode": "0",
                        "fields": [
                            {
                                "type": 2,
                                "name": "groupids.0",
                                "value": 4
                            },
                            {
                                "type": 3,
                                "name": "hostids.0",
                                "value": 10084
                            },
                            {
                                "type": 1,
                                "name": "items.0",
                                "value": "Zabbix server: Utilization*"
                            },
                            {
                                "type": 1,
                                "name": "primary_label",
                                "value": "{ITEM.NAME}"
                            },
                            {
                                "type": 1,
                                "name": "thresholds.0.color",
                                "value": "0EC9AC"
                            },
```

```
                            {
                                "type": 1,
                                "name": "thresholds.0.threshold",
                                "value": "0"
                            },
                            {
                                "type": 1,
                                "name": "thresholds.1.color",
                                "value": "FFD54F"
                            },
                            {
                                "type": 1,
                                "name": "thresholds.1.threshold",
                                "value": "70"
                            },
                            {
                                "type": 1,
                                "name": "thresholds.2.color",
                                "value": "FF465C"
                            },
                            {
                                "type": 1,
                                "name": "thresholds.2.threshold",
                                "value": "90"
                            },
                            {
                                "type": 1,
                                "name": "reference",
                                "value": "KSTMQ"
                            }
                        ]
                    }
                ]
            }
        ],
        "userGroups": [
            {
                "usrgrpid": 7,
                "permission": 2
            }
        ],
        "users": [
            {
                "userid": 1,
                "permission": 3
            }
        ]
    },
    "id": 1
}
```


Réponse :


```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```


Voir aussi

- Champ de widget de tableau de bord
- `dashboard.create`
- `dashboard.update`

12 Disponibilité de l'hôte

Description

Ces paramètres et les valeurs de propriété possibles pour les objets de champ correspondants du widget de tableau de bord vous permettent de configurer le widget *Disponibilité des hôtes* dans les méthodes `dashboard.create` et `dashboard.update`.

Paramètres

Les paramètres suivants sont pris en charge pour le widget *Disponibilité des hôtes*.

Paramètre	type	name	value
Intervalle d'actualisation	0	rf_rate	0 - Aucune actualisation; 10 - 10 secondes; 30 - 30 secondes; 60 - 1 minute; 120 - 2 minutes; 600 - 10 minutes; 900 - *(par défaut)* 15 minutes.
Groupes d'hôtes	2	groupids.0	ID du groupe d'hôtes. Remarque : pour configurer plusieurs groupes d'hôtes, créez un objet de champ de widget de tableau de bord pour chaque groupe d'hôtes avec un numéro incrémenté dans le nom de la propriété. Ce paramètre n'est pas pris en charge lors de la configuration du widget sur un tableau de bord de modèle.
Groupes d'hôtes (widget)	1	groupids._reference	Au lieu de l'ID du groupe d'hôtes : `ABCDE._hostgroupids` - définissez un widget compatible (avec son paramètre *Référence* défini sur "`ABCDE`") comme source de données pour les groupes d'hôtes. Ce paramètre n'est pas pris en charge lors de la configuration du widget sur un tableau de bord de modèle.
Type d'interface	0	interface_type.0	0 - Aucun; 1 - Agent Zabbix (contrôles passifs); 2 - SNMP; 3 - IPMI; 4 - JMX; 5 - Agent Zabbix (contrôles actifs). Par défaut : 1, 2, 3, 4, 5 (tous activés). Remarque : pour configurer plusieurs valeurs, créez un objet de champ de widget de tableau de bord pour chaque valeur avec un numéro incrémenté dans le nom de la propriété.
Disposition	0	layout	0 - *(par défaut)* Horizontale; 1 - Verticale.
Afficher les hôtes en maintenance	0	maintenance	0 - *(par défaut)* Désactivé; 1 - Activé.

Paramètre	type	name	value
Afficher uniquement les totaux	0	only_totals	0 - *(par défaut)* Désactivé; 1 - Activé.

Exemples

Les exemples suivants visent à décrire uniquement la configuration des objets de champ du widget de tableau de bord pour le widget *Disponibilité de l'hôte*. Pour plus d'informations sur la configuration d'un tableau de bord, consultez `dashboard.create`.

Configuration d'un widget *Host availability*

Configurez un widget *Host availability* qui affiche des informations de disponibilité (dans une disposition verticale) pour les hôtes du groupe d'hôtes « 4 » avec des interfaces « Zabbix agent » et « SNMP » configurées.

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": 30,
        "auto_start": 1,
        "pages": [
            {
                "widgets": [
                    {
                        "type": "hostavail",
                        "name": "Host availability",
                        "x": 0,
                        "y": 0,
                        "width": 18,
                        "height": 3,
                        "view_mode": 0,
                        "fields": [
                            {
                                "type": 2,
                                "name": "groupids.0",
                                "value": 4
                            },
                            {
                                "type": 0,
                                "name": "interface_type",
                                "value": 1
                            },
                            {
                                "type": 0,
                                "name": "interface_type",
                                "value": 2
                            },
                            {
                                "type": 0,
                                "name": "layout",
                                "value": 1
                            }
                        ]
                    }
                ]
            }
        ],
```

```
        "userGroups": [
            {
                "usrgrpid": 7,
                "permission": 2
            }
        ],
        "users": [
            {
                "userid": 1,
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Voir également

- Champ du widget du tableau de bord
- `dashboard.create`
- `dashboard.update`

13 Carte d'hôte

Description

Ces paramètres et les valeurs de propriété possibles pour les objets de champ respectifs du widget de tableau de bord vous permettent de configurer le widget *Carte de l'hôte* dans les méthodes `dashboard.create` et `dashboard.update`.

Attention:
Les propriétés `fields` du widget ne sont pas validées lors de la création ou de la mise à jour d'un tableau de bord. Cela permet aux utilisateurs de modifier les widgets intégrés et de créer des widgets personnalisés, mais introduit également le risque de créer ou de mettre à jour des widgets de manière incorrecte. Pour garantir la réussite de la création ou de la mise à jour du widget *Carte de l'hôte*, veuillez vous reporter au comportement des paramètres décrit dans les tableaux ci-dessous.

Paramètres

Les paramètres suivants sont pris en charge pour le widget *Carte d'hôte*.

Paramètre	type	name	value
Intervalle d'actualisation	0	rf_rate	0 - Pas d'actualisation ; 10 - 10 secondes ; 30 - 30 secondes ; 60 - *(par défaut)* 1 minute ; 120 - 2 minutes ; 600 - 10 minutes ; 900 - 15 minutes.

Paramètre	type	name	value
Hôte	3	hostid.0	ID de l'hôte. Comportement du paramètre : - *obligatoire* si *Hôte (Widget/Tableau de bord)* n'est pas défini Ce paramètre n'est pas pris en charge lors de la configuration du widget sur un tableau de bord de modèle.
Hôte (Widget/Tableau de bord)	1	hostid._reference	Au lieu de l'ID de l'hôte : `DASHBOARD.hostid` - définit le sélecteur d'hôte du tableau de bord comme source de données pour l'hôte ; `ABCDE._hostid` - définit un widget compatible (avec son paramètre *Référence* défini sur "ABCDE") comme source de données pour l'hôte. Comportement du paramètre : - *obligatoire* si *Hôte* n'est pas défini Ce paramètre n'est pas pris en charge lors de la configuration du widget sur un tableau de bord de modèle.
Afficher les problèmes supprimés	0	show_suppressed	0 - *(par défaut)* Désactivé ; 1 - Activé.
Afficher	0	sections.0	0 - Groupes d'hôtes ; 1 - Description ; 2 - Surveillance ; 3 - Disponibilité ; 4 - Surveillé par ; 5 - Modèles ; 6 - Inventaire ; 7 - Tags. Remarque : Le nombre dans le nom de propriété fait référence à l'ordre des sections dans la liste des sections. Pour configurer plusieurs sections, créez un objet de champ de widget de tableau de bord pour chaque section avec un nombre incrémenté dans le nom de propriété.

Les paramètres suivants sont pris en charge si *Afficher* est défini sur "Inventaire".

Paramètre	type	name	value
Champs d'inventaire	0	inventory.0	ID de l'inventaire. Remarque : Pour configurer plusieurs champs d'inventaire, créez un objet de champ de widget de tableau de bord pour chaque champ d'inventaire avec un nombre incrémenté dans le nom de propriété.

Exemples

Les exemples suivants visent uniquement à décrire la configuration des objets de champ du widget de tableau de bord pour le widget *Host card*. Pour plus d'informations sur la configuration d'un tableau de bord, consultez `dashboard.create`.

Configuration d'un widget *Carte d'hôte*

Configurez un widget *Carte d'hôte* qui affiche les sections suivantes : « Monitoring », « Availability », « Monitored by », « Inventory » et « Tags ».

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": 30,
        "auto_start": 1,
        "pages": [
            {
                "widgets": [
                    {
                        "type": "hostcard",
                        "name": "Host card",
                        "x": 0,
                        "y": 0,
                        "width": 14,
                        "height": 7,
                        "view_mode": 0,
                        "fields": [
                            {
                                "type": 3,
                                "name": "hostid.0",
                                "value": 10084
                            },
                            {
                                "type": 0,
                                "name": "show_suppressed",
                                "value": 1
                            },
                            {
                                "type": 0,
                                "name": "sections.0",
                                "value": 2
                            },
                            {
                                "type": 0,
                                "name": "sections.1",
                                "value": 3
                            },
                            {
                                "type": 0,
                                "name": "sections.2",
                                "value": 4
                            },
                            {
                                "type": 0,
                                "name": "sections.3",
                                "value": 6
                            },
                            {
                                "type": 0,
                                "name": "sections.4",
                                "value": 7
                            },
                            {
                                "type": 0,
                                "name": "inventory.0",
                                "value": 25
                            },
                            {
                                "type": 0,
                                "name": "inventory.1",
```

```
                                "value": 26
                            }
                        ]
                    }
                ]
            }
        ],
        "userGroups": [
            {
                "usrgrpid": 7,
                "permission": 2
            }
        ],
        "users": [
            {
                "userid": 1,
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Voir aussi

- Champ de widget de tableau de bord
- `dashboard.create`
- `dashboard.update`

######14 Navigateur d'hôte {#manual-api-reference-dashboard-widget_fields-host_navigator}

Description

Ces paramètres et les valeurs de propriété possibles pour les objets de champ correspondants du widget de tableau de bord vous permettent de configurer le widget *Navigateur d'hôtes* dans les méthodes `dashboard.create` et `dashboard.update`.

Attention:
Les propriétés `fields` du widget ne sont pas validées lors de la création ou de la mise à jour d'un tableau de bord. Cela permet aux utilisateurs de modifier les widgets intégrés et de créer des widgets personnalisés, mais introduit également le risque de créer ou de mettre à jour des widgets de manière incorrecte. Pour garantir la création ou la mise à jour réussie du widget *Navigateur d'hôtes*, veuillez vous reporter au comportement des paramètres décrit dans les tableaux ci-dessous.

Paramètres

Les paramètres suivants sont pris en charge pour le widget *Navigateur d'hôtes*.

Paramètre	type	name	value
Intervalle d'actualisation	0	rf_rate	0 - Aucune actualisation ; 10 - 10 secondes ; 30 - 30 secondes ; 60 - *(par défaut)* 1 minute ; 120 - 2 minutes ; 600 - 10 minutes ; 900 - 15 minutes.
Groupes d'hôtes	2	groupids.0	ID du groupe d'hôtes. Remarque : pour configurer plusieurs groupes d'hôtes, créez un objet de champ de widget de tableau de bord pour chaque groupe d'hôtes, avec un numéro incrémenté dans le nom de la propriété. Ce paramètre n'est pas pris en charge lors de la configuration du widget sur un tableau de bord de modèle.
Groupes d'hôtes (widget)	1	groupids._reference	Au lieu de l'ID du groupe d'hôtes : `ABCDE._hostgroupids` - définissez un widget compatible (avec son paramètre *Référence* défini sur « `ABCDE` ») comme source de données pour les groupes d'hôtes. Ce paramètre n'est pas pris en charge lors de la configuration du widget sur un tableau de bord de modèle.
Modèles d'hôtes	1	hosts.0	Nom ou modèle de hôte. Remarque : pour configurer plusieurs modèles d'hôtes, créez un objet de champ de widget de tableau de bord pour chaque modèle d'hôte, avec un numéro incrémenté dans le nom de la propriété. Pour plusieurs modèles d'hôtes, le paramètre *Groupes d'hôtes* doit être soit entièrement non configuré, soit configuré avec au moins un groupe d'hôtes auquel appartiennent les hôtes correspondant aux modèles d'hôtes configurés. Ce paramètre n'est pas pris en charge lors de la configuration du widget sur un tableau de bord de modèle.
État de l'hôte	0	status	-1 - *(par défaut)* Tous ; 0 - Activé ; 1 - Désactivé. Ce paramètre n'est pas pris en charge lors de la configuration du widget sur un tableau de bord de modèle.
Étiquettes d'hôte			
Type d'évaluation	0	host_tags_evaltype	0 - *(par défaut)* Et/Ou ; 2 - Ou. Ce paramètre n'est pas pris en charge lors de la configuration du widget sur un tableau de bord de modèle.
Nom de l'étiquette	1	host_tags.0.tag	Toute valeur de chaîne. Remarque : le numéro dans le nom de la propriété fait référence à l'ordre de l'étiquette dans la liste d'évaluation des étiquettes. Comportement du paramètre : - *obligatoire* lors de la configuration des *Étiquettes d'hôte* Ce paramètre n'est pas pris en charge lors de la configuration du widget sur un tableau de bord de modèle.

Paramètre	type	name	value
Opérateur	0	host_tags.0.operator	0 - Contient ; 1 - Est égal à ; 2 - Ne contient pas ; 3 - N'est pas égal à ; 4 - Existe ; 5 - N'existe pas. Remarque : le numéro dans le nom de la propriété fait référence à l'ordre de l'étiquette dans la liste d'évaluation des étiquettes. Comportement du paramètre : - *obligatoire* lors de la configuration des *Étiquettes d'hôte* Ce paramètre n'est pas pris en charge lors de la configuration du widget sur un tableau de bord de modèle.
Valeur de l'étiquette	1	host_tags.0.value	Toute valeur de chaîne. Remarque : le numéro dans le nom de la propriété fait référence à l'ordre de l'étiquette dans la liste d'évaluation des étiquettes. Comportement du paramètre : - *obligatoire* lors de la configuration des *Étiquettes d'hôte* Ce paramètre n'est pas pris en charge lors de la configuration du widget sur un tableau de bord de modèle.
Gravité	0	severities.0	0 - Non classifié ; 1 - Information ; 2 - Avertissement ; 3 - Moyenne ; 4 - Élevée ; 5 - Désastre. Par défaut : vide (toutes activées). Remarque : pour configurer plusieurs valeurs, créez un objet de champ de widget de tableau de bord pour chaque valeur, avec un numéro incrémenté dans le nom de la propriété.
Afficher les hôtes en maintenance	0	maintenance	0 - *(par défaut)* Désactivé ; 1 - Activé.
Afficher les problèmes	0	show_problems	0 - Tous ; 1 - *(par défaut)* Non supprimés ; 2 - Aucun.
Regrouper par *Attribut*	0	group_by.0.attribute	0 - Groupe d'hôtes ; 1 - Valeur de l'étiquette ; 2 - Gravité. Remarque : le numéro dans le nom de la propriété fait référence à l'ordre de l'attribut dans la liste des attributs de regroupement. Comportement du paramètre : - *obligatoire* lors de la configuration de *Regrouper par*

Paramètre	type	name	value
Valeur	1	group_by.0.tag_name	Toute valeur de chaîne. Remarque : le numéro dans le nom de la propriété fait référence à l'attribut de regroupement défini dans le paramètre *Attribut*. Comportement du paramètre : - *obligatoire* lors de la configuration de *Regrouper par* et lorsque *Attribut* est défini sur « Valeur de l'étiquette »
Limite d'hôtes	0	show_lines	Les valeurs possibles vont de 1 à 9999. Par défaut : 100. Ce paramètre n'est pas pris en charge lors de la configuration du widget sur un tableau de bord de modèle.
Référence	1	reference	Toute valeur de chaîne composée de 5 caractères (par exemple, `ABCDE` ou `JBPNL`). Cette valeur doit être unique au sein du tableau de bord auquel appartient le widget. Comportement du paramètre : - *obligatoire*

Exemples

Les exemples suivants ont pour objectif de décrire uniquement la configuration des objets champs du widget du tableau de bord pour le widget *Navigateurd'hôte*. Pour plus d'informations sur la configuration d'un tableau de bord, consultez `dashboard.create`.

Configuration d'un widget *Navigateur d'hôtes*

Configurez un widget *Navigateur d'hôtes* qui affiche les hôtes regroupés par leur groupe d'hôtes puis par la valeur du tag « city ».

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": "30",
        "auto_start": "1",
        "pages": [
            {
                "widgets": [
                    {
                        "type": "hostnavigator",
                        "name": "Host navigator",
                        "x": "0",
                        "y": "0",
                        "width": "12",
                        "height": "5",
                        "view_mode": "0",
                        "fields": [
                            {
                                "type": 2,
                                "name": "groupids.0",
                                "value": 2
                            },
                            {
                                "type": 2,
                                "name": "groupids.1",
                                "value": 4
                            },
                            {
```

```
                                "type": 0,
                                "name": "group_by.0.attribute",
                                "value": 0
                            },
                            {
                                "type": 0,
                                "name": "group_by.1.attribute",
                                "value": 1
                            },
                            {
                                "type": 1,
                                "name": "group_by.1.tag_name",
                                "value": "city"
                            },
                            {
                                "type": 1,
                                "name": "reference",
                                "value": "SWKLB"
                            }
                        ]
                    }
                ]
            }
        ],
        "userGroups": [
            {
                "usrgrpid": 7,
                "permission": 2
            }
        ],
        "users": [
            {
                "userid": 1,
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Voir également

- Champ du widget du tableau de bord
- `dashboard.create`
- `dashboard.update`

15 Carte d'élément

Description

Ces paramètres et les valeurs de propriété possibles pour les objets de champ respectifs du widget de tableau de bord vous

permettent de configurer le widget *Carte d'élément* dans les méthodes `dashboard.create` et `dashboard.update`.

Attention:
Les propriétés `fields` du widget ne sont pas validées lors de la création ou de la mise à jour d'un tableau de bord. Cela permet aux utilisateurs de modifier les widgets intégrés et de créer des widgets personnalisés, mais introduit également le risque de créer ou de mettre à jour des widgets de manière incorrecte. Pour garantir la création ou la mise à jour réussie du widget *Carte d'élément*, veuillez vous reporter au comportement des paramètres décrit dans les tableaux ci-dessous.

Paramètres

Les paramètres suivants sont pris en charge pour le widget *Carte d'élément*.

Paramètre	type	name	value
Intervalle d'actualisation	0	rf_rate	0 - Aucune actualisation; 10 - 10 secondes; 30 - 30 secondes; 60 - *(par défaut)* 1 minute; 120 - 2 minutes; 600 - 10 minutes; 900 - 15 minutes.
Élément	4	itemid.0	ID de l'élément. Comportement du paramètre : - *obligatoire* si *Élément (widget)* n'est pas défini
Élément (widget)	1	itemid._reference	Au lieu de l'ID de l'élément : `ABCDE._itemid` - définir un widget compatible (avec son paramètre *Référence* défini sur "`ABCDE`") comme source de données pour l'élément. Comportement du paramètre : - *obligatoire* si *Élément* n'est pas défini
Afficher	0	sections.0	0 - Description; 1 - Texte d'erreur; 2 - Métriques; 3 - Dernières données; 4 - Type d'information; 5 - Déclencheurs; 6 - Interface de l'hôte; 7 - Type; 8 - Inventaire de l'hôte; 9 - Tags. Remarque : Le numéro dans le nom de propriété fait référence à l'ordre des sections dans la liste des sections. Pour configurer plusieurs sections, créez un objet de champ de widget de tableau de bord pour chaque section avec un numéro incrémenté dans le nom de propriété.
Remplacer l'hôte	1	override_hostid._reference	`ABCDE._hostid` - définir un widget compatible (avec son paramètre *Référence* défini sur "`ABCDE`") comme source de données pour les hôtes ; `DASHBOARD._hostid` - définir le sélecteur d'hôte du tableau de bord comme source de données pour les hôtes. Par défaut : `""` (vide) Ce paramètre n'est pas pris en charge lors de la configuration du widget sur un tableau de bord de modèle.

Sparkline

Les paramètres suivants sont pris en charge si *Afficher* est défini sur « Dernières données ».

Paramètre	type	name	value
Largeur	0	sparkline.width	Les valeurs possibles vont de 0 à 10. Valeur par défaut : 1.
Remplissage	0	sparkline.fill	Les valeurs possibles vont de 0 à 10. Valeur par défaut : 3.
Couleur	1	sparkline.color	Code couleur hexadécimal (par exemple, `FF0000`). Valeur par défaut : `42A5F5`.
Période	1	sparkline.time_period.from	`DASHBOARD._timeperiod` - définir le sélecteur de période du tableau de bord comme source de données ; `ABCDE._timeperiod` - définir un widget compatible (avec son paramètre `reference` défini sur `ABCDE`) comme source de données. Valeur par défaut : `""` (vide) Vous pouvez également définir la période uniquement dans les paramètres *Du* et *Au*.
Du	1	sparkline.time_period.from	Chaîne temporelle valide au format absolu (`YYYY-MM-DD hh:mm:ss`) ou au format temporel relatif (`now`, `now/d`, `now/w-1w`, etc.). Valeur par défaut : now-1h. Comportement du paramètre : - *pris en charge* si *Période* n'est pas défini - *obligatoire* si `sparkline.time_period.to` est défini
Au	1	sparkline.time_period.to	Chaîne temporelle valide au format absolu (`YYYY-MM-DD hh:mm:ss`) ou au format temporel relatif (`now`, `now/d`, `now/w-1w`, etc.). Valeur par défaut : now. Comportement du paramètre : - *pris en charge* si *Période* n'est pas défini - *obligatoire* si `time_period.to` est défini
Données d'historique	0	sparkline.history	0 - *(par défaut)* Automatique ; 1 - Historique ; 2 - Tendances.

Exemples

Les exemples suivants visent uniquement à décrire la configuration des objets de champ du widget de tableau de bord pour le widget *Item card*. Pour plus d'informations sur la configuration d'un tableau de bord, consultez `dashboard.create`.

Configuration d'un widget *Carte d'élément*

Configurez un widget *Carte d'élément* qui affiche les sections suivantes : « Description », « Dernières données », « Déclencheurs » et « Étiquettes ».

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": 30,
        "auto_start": 1,
        "pages": [
            {
                "widgets": [
```

```
                    {
                        "type": "itemcard",
                        "name": "Item card",
                        "x": 0,
                        "y": 0,
                        "width": 14,
                        "height": 7,
                        "view_mode": 0,
                        "fields": [
                            {
                                "type": 4,
                                "name": "itemid.0",
                                "value": 42257
                            },
                            {
                                "type": 0,
                                "name": "sections.0",
                                "value": 0
                            },
                            {
                                "type": 0,
                                "name": "sections.1",
                                "value": 3
                            },
                            {
                                "type": 0,
                                "name": "sections.2",
                                "value": 5
                            },
                            {
                                "type": 0,
                                "name": "sections.3",
                                "value": 9
                            }
                        ]
                    }
                ]
            }
        ],
        "userGroups": [
            {
                "usrgrpid": 7,
                "permission": 2
            }
        ],
        "users": [
            {
                "userid": 1,
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
```

```
    },
    "id": 1
}
```

Voir aussi

- Champ de widget de tableau de bord
- `dashboard.create`
- `dashboard.update`

16 Historique des éléments

Description

Ces paramètres et les valeurs de propriété possibles pour les objets de champ respectifs du widget de tableau de bord vous permettent de configurer le widget *Historique des éléments* dans les méthodes `dashboard.create` et `dashboard.update`.

Attention:
Les propriétés `fields` du widget ne sont pas validées lors de la création ou de la mise à jour d'un tableau de bord. Cela permet aux utilisateurs de modifier les widgets intégrés et de créer des widgets personnalisés, mais introduit également le risque de créer ou de mettre à jour des widgets de manière incorrecte. Pour garantir la création ou la mise à jour réussie du widget *Historique des éléments*, veuillez vous reporter au comportement des paramètres décrit dans les tableaux ci-dessous.

Paramètres

Les paramètres suivants sont pris en charge pour le widget *Historique des éléments*.

Paramètre	type	name	value
Intervalle d'actualisation	0	rf_rate	0 - Aucune actualisation ; 10 - 10 secondes ; 30 - 30 secondes ; 60 - *(par défaut)* 1 minute ; 120 - 2 minutes ; 600 - 10 minutes ; 900 - 15 minutes.
Disposition	0	layout	0 - *(par défaut)* Horizontale ; 1 - Verticale.
Colonnes (voir ci-dessous)			
Afficher les lignes	0	show_lines	Les valeurs possibles vont de 1 à 100. Valeur par défaut : 25.
Remplacer le hôte	1	override_hostid._reference	`ABCDE._hostid` - définir un widget compatible (avec son paramètre *Référence* défini sur "`ABCDE`") comme source de données pour les hôtes ; `DASHBOARD._hostid` - définir le sélecteur d'hôtes du tableau de bord comme source de données pour les hôtes. Ce paramètre n'est pas pris en charge lors de la configuration du widget sur un tableau de bord de modèle.
Configuration avancée (voir ci-dessous)			
Référence	1	reference	Toute valeur de chaîne composée de 5 caractères (par exemple, `ABCDE` ou `JBPNL`). Cette valeur doit être unique au sein du tableau de bord auquel appartient le widget. Comportement du paramètre : *- requis*

Colonnes

Les colonnes ont des paramètres communs et des paramètres supplémentaires selon la configuration du paramètre *Élément*.

Note:
Pour tous les paramètres liés aux colonnes, le numéro dans le nom de la propriété (par ex. columns.0.name) fait référence à une colonne pour laquelle le paramètre est configuré.

Les paramètres suivants sont pris en charge pour toutes les colonnes.

Parameter	type	name	value
Nom	1	columns.0.name	Toute valeur de chaîne. Comportement du paramètre: - *obligatoire*
Élément	4	columns.0.itemid	ID de l'élément. Lors de la configuration du widget sur un tableau de bord de modèle, seuls les éléments configurés sur le modèle doivent être définis. Comportement du paramètre: - *obligatoire*
Couleur de base	1	columns.0.base_color	Code couleur hexadécimal (par ex. FF0000). Par défaut : "" (vide).

Les paramètres de colonne suivants sont pris en charge si l'*élément* configuré est un élément de type numérique.

Paramètre	type	name	value
Affichage	0	columns.0.display	1 - *(par défaut)* Tel quel ; 2 - Barre ; 3 - Indicateurs.
Min	1	columns.0.min	Toute valeur numérique. Comportement du paramètre : - *pris en charge* si *Affichage* est défini sur « Barre » ou « Indicateurs »
Max	1	columns.0.max	Toute valeur numérique. Comportement du paramètre : - *pris en charge* si *Affichage* est défini sur « Barre » ou « Indicateurs »
Seuils			
Couleur	1	columns.0.thresholds.color	Code couleur hexadécimal (par ex. FF0000).
Seuil	1	columns.0.thresholds.threshold	Toute valeur numérique. Les suffixes (par ex. « 1d », « 2w », « 4K », « 8G ») sont pris en charge.
Données d'historique	0	columns.0.history	0 - *(par défaut)* Auto ; 1 - Historique ; 2 - Tendances.

Les paramètres de colonne suivants sont pris en charge si l'*élément* configuré est un élément de type caractère, texte ou journal.

Parameter	type	name	value
Mise en évidence			
Mise en évidence	1	columns.0.highlights.color	Code couleur hexadécimal (par exemple, FF0000).
Seuil	1	columns.0.highlights.pattern	Toute expression régulière.
Affichage	0	columns.0.display	1 - *(par défaut)* Tel quel ; 4 - HTML ; 5 - Une seule ligne.

Parameter	type	name	value
Une seule ligne	0	columns.0.max_length	Les valeurs possibles vont de 1 à 500. Valeur par défaut : 100. Comportement du paramètre : - *pris en charge* si *Affichage* est défini sur « Une seule ligne »
Utiliser une police à chasse fixe	0	columns.0.monospace_font	0 - *(par défaut)* Utiliser la police par défaut ; 1 - Utiliser une police à chasse fixe.
Afficher l'heure locale	0	columns.0.local_time	0 - *(par défaut)* Afficher l'horodatage ; 1 - Afficher l'heure locale. Comportement du paramètre : - *pris en charge* si *Élément* est défini sur un élément de type journal et si *Afficher l'horodatage* est défini sur « Activé »

Les paramètres de colonne suivants sont pris en charge si l'*élément* configuré est un élément de type binaire.

Paramètre	type	name	value
Afficher la miniature	1	columns.0.show_thumbnail	0 - *(par défaut)* Désactivé; 1 - Activé.

Configuration avancée

Les paramètres de configuration avancée suivants sont pris en charge pour le widget *Historique des éléments*.

Paramètre	type	name	value
Nouvelles valeurs	0	sortorder	0 - *(par défaut)* En haut ; 1 - En bas.
Afficher l'horodatage	0	show_timestamp	0 - *(par défaut)* Désactivé ; 1 - Activé.
Afficher l'en-tête de colonne	0	show_column_header	0 - Désactivé ; 1 - Horizontal ; 2 - *(par défaut)* Vertical.
Période	1	time_period._reference	`DASHBOARD._timeperiod` - définir le sélecteur de période du tableau de bord comme source de données ; `ABCDE._timeperiod` - définir un widget compatible (avec son paramètre *Reference* défini sur "`ABCDE`") comme source de données. Par défaut : `DASHBOARD._timeperiod` Vous pouvez également définir la période uniquement dans les paramètres *From* et *To*.
De	1	time_period.from	Chaîne temporelle valide en syntaxe absolue (`YYYY-MM-DD hh:mm:ss`) ou relative (`now`, `now/d`, `now/w-1w`, etc.). Comportement du paramètre : - *pris en charge* si *Période* n'est pas défini - *obligatoire* si `time_period.to` est défini

Paramètre	type	name	value
À	1	time_period.to	Chaîne temporelle valide en syntaxe absolue (`YYYY-MM-DD hh:mm:ss`) ou relative (`now`, `now/d`, `now/w-1w`, etc.). Comportement du paramètre : - *pris en charge* si *Période* n'est pas défini - *obligatoire* si `time_period.from` est défini

Exemples

Les exemples suivants visent uniquement à décrire la configuration des objets de champ du widget de tableau de bord pour le widget *Historique des éléments*. Pour plus d'informations sur la configuration d'un tableau de bord, consultez `dashboard.create`.

Configuration d'un widget *Historique des éléments*

Configurez un widget *Historique des éléments* qui affiche les dernières données de deux éléments numériques « 42269 » et « 42270 ». En outre, configurez les colonnes des éléments pour qu'elles soient affichées verticalement, avec les noms de colonnes affichés horizontalement ; limitez l'affichage à 15 lignes de données et incluez une colonne d'horodatage distincte.

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": 30,
        "auto_start": 1,
        "pages": [
            {
                "widgets": [
                    {
                        "type": "itemhistory",
                        "name": "Item history",
                        "x": "0",
                        "y": "0",
                        "width": "18",
                        "height": "6",
                        "view_mode": "0",
                        "fields": [
                            {
                                "type": "0",
                                "name": "layout",
                                "value": "1"
                            },
                            {
                                "type": "1",
                                "name": "columns.0.name",
                                "value": "CPU utilization"
                            },
                            {
                                "type": "4",
                                "name": "columns.0.itemid",
                                "value": "42269"
                            },
                            {
                                "type": "1",
                                "name": "columns.1.name",
                                "value": "Memory utilization"
                            },
                            {
                                "type": "4",
                                "name": "columns.1.itemid",
                                "value": "42270"
```

```
                            },
                            {
                                "type": "0",
                                "name": "show_lines",
                                "value": "15"
                            },
                            {
                                "type": "0",
                                "name": "show_timestamp",
                                "value": "1"
                            },
                            {
                                "type": "0",
                                "name": "show_column_header",
                                "value": "1"
                            },
                            {
                                "type": "1",
                                "name": "reference",
                                "value": "KIVKD"
                            }
                        ]
                    }
                ]
            }
        ],
        "userGroups": [
            {
                "usrgrpid": 7,
                "permission": 2
            }
        ],
        "users": [
            {
                "userid": 1,
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Voir aussi

- Champ de widget de tableau de bord
- dashboard.create
- dashboard.update

17 Navigateur d'éléments

Description

Ces paramètres et les valeurs de propriété possibles pour les objets de champ respectifs du widget de tableau de bord vous permettent de configurer le widget *Navigateur d'éléments* dans les méthodes `dashboard.create` et `dashboard.update`.

Attention:
Les propriétés `fields` du widget ne sont pas validées lors de la création ou de la mise à jour d'un tableau de bord. Cela permet aux utilisateurs de modifier les widgets intégrés et de créer des widgets personnalisés, mais introduit également le risque de créer ou de mettre à jour des widgets de manière incorrecte. Pour garantir la réussite de la création ou de la mise à jour du widget *Navigateur d'éléments*, veuillez vous reporter au comportement des paramètres décrit dans les tableaux ci-dessous.

Paramètres

Les paramètres suivants sont pris en charge pour le widget *Navigateur d'éléments*.

Paramètre	type	name	value
Intervalle d'actualisation	0	rf_rate	0 - Aucune actualisation ; 10 - 10 secondes ; 30 - 30 secondes ; 60 - *(par défaut)* 1 minute ; 120 - 2 minutes ; 600 - 10 minutes ; 900 - 15 minutes.
Groupes de hôtes	2	groupids.0	ID du groupe de hôtes. Remarque : pour configurer plusieurs groupes de hôtes, créez un objet de champ de widget de tableau de bord pour chaque groupe de hôtes en incrémentant le numéro dans le nom de la propriété. Ce paramètre n'est pas pris en charge lors de la configuration du widget sur un tableau de bord de modèle.
Groupes de hôtes (widget)	1	groupids._reference	Au lieu de l'ID du groupe de hôtes : `ABCDE._hostgroupids` - définissez un widget compatible (avec son paramètre *Référence* défini sur "`ABCDE`") comme source de données pour les groupes de hôtes. Ce paramètre n'est pas pris en charge lors de la configuration du widget sur un tableau de bord de modèle.
Hôtes	3	hostids.0	ID de l'hôte. Remarque : pour configurer plusieurs hôtes, créez un objet de champ de widget de tableau de bord pour chaque hôte en incrémentant le numéro dans le nom de la propriété. Pour plusieurs hôtes, le paramètre *Groupes de hôtes* doit soit ne pas être configuré, soit être configuré avec au moins un groupe de hôtes auquel appartiennent les hôtes configurés. Ce paramètre n'est pas pris en charge lors de la configuration du widget sur un tableau de bord de modèle.
Hôtes (widget/tableau de bord)	1	hostids._reference	Au lieu de l'ID de l'hôte : `DASHBOARD.hostid` - définissez le sélecteur d'hôte du tableau de bord comme source de données pour les hôtes ; `ABCDE._hostid` - définissez un widget compatible (avec son paramètre *Référence* défini sur "`ABCDE`") comme source de données pour les hôtes. Ce paramètre n'est pas pris en charge lors de la configuration du widget sur un tableau de bord de modèle.
Étiquettes d'hôte			

Paramètre	type	name	value
Type d'évaluation	0	host_tags_evaltype	0 - *(par défaut)* Et/Ou ; 2 - Ou. Ce paramètre n'est pas pris en charge lors de la configuration du widget sur un tableau de bord de modèle.
Nom de l'étiquette	1	host_tags.0.tag	Toute valeur de chaîne. Remarque : le numéro dans le nom de la propriété fait référence à l'ordre de l'étiquette dans la liste d'évaluation des étiquettes. Comportement du paramètre : - *obligatoire* lors de la configuration des *Étiquettes d'hôte* Ce paramètre n'est pas pris en charge lors de la configuration du widget sur un tableau de bord de modèle.
Opérateur	0	host_tags.0.operator	0 - Contient ; 1 - Est égal à ; 2 - Ne contient pas ; 3 - N'est pas égal à ; 4 - Existe ; 5 - N'existe pas. Remarque : le numéro dans le nom de la propriété fait référence à l'ordre de l'étiquette dans la liste d'évaluation des étiquettes. Comportement du paramètre : - *obligatoire* lors de la configuration des *Étiquettes d'hôte* Ce paramètre n'est pas pris en charge lors de la configuration du widget sur un tableau de bord de modèle.
Valeur de l'étiquette	1	host_tags.0.value	Toute valeur de chaîne. Remarque : le numéro dans le nom de la propriété fait référence à l'ordre de l'étiquette dans la liste d'évaluation des étiquettes. Comportement du paramètre : - *obligatoire* lors de la configuration des *Étiquettes d'hôte* Ce paramètre n'est pas pris en charge lors de la configuration du widget sur un tableau de bord de modèle.
Motifs d'éléments	1	items.0	Nom ou motif d'élément. Remarque : pour configurer plusieurs motifs d'éléments, créez un objet de champ de widget de tableau de bord pour chaque motif d'élément en incrémentant le numéro dans le nom de la propriété.
Étiquettes d'élément			
Type d'évaluation	0	item_tags_evaltype	0 - *(par défaut)* Et/Ou ; 2 - Ou.
Nom de l'étiquette	1	item_tags.0.tag	Toute valeur de chaîne. Remarque : le numéro dans le nom de la propriété fait référence à l'ordre de l'étiquette dans la liste d'évaluation des étiquettes. Comportement du paramètre : - *obligatoire* lors de la configuration des *Étiquettes d'élément*

Paramètre	type	name	value
Opérateur	0	item_tags.0.operator	0 - Contient ; 1 - Est égal à ; 2 - Ne contient pas ; 3 - N'est pas égal à ; 4 - Existe ; 5 - N'existe pas. Remarque : le numéro dans le nom de la propriété fait référence à l'ordre de l'étiquette dans la liste d'évaluation des étiquettes. Comportement du paramètre : - *obligatoire* lors de la configuration des *Étiquettes d'élément*
Valeur de l'étiquette	1	item_tags.0.value	Toute valeur de chaîne. Remarque : le numéro dans le nom de la propriété fait référence à l'ordre de l'étiquette dans la liste d'évaluation des étiquettes. Comportement du paramètre : - *obligatoire* lors de la configuration des *Étiquettes d'élément*
État	0	state	-1 - *(par défaut)* Tous ; 0 - Normal ; 1 - Non pris en charge.
Afficher les problèmes	0	show_problems	0 - Tous ; 1 - *(par défaut)* Non supprimés ; 2 - Aucun.
Regrouper par			
Attribut	0	group_by.0.attribute	0 - Groupe de hôtes ; 1 - Nom d'hôte ; 2 - Valeur de l'étiquette d'hôte ; 3 - Valeur de l'étiquette d'élément. Remarque : le numéro dans le nom de la propriété fait référence à l'ordre de l'attribut dans la liste des attributs de regroupement. Comportement du paramètre : - *obligatoire* lors de la configuration de *Regrouper par*
Valeur	1	group_by.0.tag_name	Toute valeur de chaîne. Remarque : le numéro dans le nom de la propriété fait référence à l'attribut de regroupement défini dans le paramètre *Attribut*. Comportement du paramètre : - *obligatoire* lors de la configuration de *Regrouper par* lorsque *Attribut* est défini sur « Valeur de l'étiquette d'hôte » ou « Valeur de l'étiquette d'élément »
Limite d'éléments	0	show_lines	Les valeurs possibles sont comprises entre 1 et 9999. Valeur par défaut : 100.
Référence	1	reference	Toute valeur de chaîne composée de 5 caractères (par exemple, `ABCDE` ou `JBPNL`). Cette valeur doit être unique au sein du tableau de bord auquel appartient le widget. Comportement du paramètre : - *obligatoire*

Exemples

Les exemples suivants visent uniquement à décrire la configuration des objets de champ de widget de tableau de bord pour le widget *Item navigator*. Pour plus d'informations sur la configuration d'un tableau de bord, consultez `dashboard.create`.

Configuration d'un widget *Navigateur d'éléments*

Configurez un widget *Navigateur d'éléments* qui affiche jusqu'à 1000 éléments regroupés par leur hôte puis par la valeur de tag d'élément « component ».

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": "30",
        "auto_start": "1",
        "pages": [
            {
                "widgets": [
                    {
                        "type": "itemnavigator",
                        "name": "Item navigator",
                        "x": "0",
                        "y": "0",
                        "width": "12",
                        "height": "5",
                        "view_mode": "0",
                        "fields": [
                            {
                                "type": 0,
                                "name": "group_by.0.attribute",
                                "value": 0
                            },
                            {
                                "type": 0,
                                "name": "group_by.1.attribute",
                                "value": 3
                            },
                            {
                                "type": 1,
                                "name": "group_by.1.tag_name",
                                "value": "component"
                            },
                            {
                                "type": 0,
                                "name": "show_lines",
                                "value": 1000
                            },
                            {
                                "type": 1,
                                "name": "reference",
                                "value": "DFNLK"
                            }
                        ]
                    }
                ]
            }
        ],
        "userGroups": [
            {
                "usrgrpid": 7,
                "permission": 2
            }
        ],
        "users": [
            {
```

```
                "userid": 1,
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Voir aussi

- Champ de widget de tableau de bord
- `dashboard.create`
- `dashboard.update`

18 Valeur d'élément

Description

Ces paramètres et les valeurs de propriété possibles pour les objets de champ respectifs du widget de tableau de bord vous permettent de configurer le widget *Valeur de l'élément* dans les méthodes `dashboard.create` et `dashboard.update`.

Attention:
Les propriétés `fields` du widget ne sont pas validées lors de la création ou de la mise à jour d'un tableau de bord. Cela permet aux utilisateurs de modifier les widgets intégrés et de créer des widgets personnalisés, mais introduit également le risque de créer ou de mettre à jour des widgets de manière incorrecte. Pour garantir la réussite de la création ou de la mise à jour du widget *Valeur de l'élément*, veuillez vous reporter au comportement des paramètres décrit dans les tableaux ci-dessous.

Paramètres

Les paramètres suivants sont pris en charge pour le widget *Valeur de l'élément*.

Paramètre	type	name	value
Intervalle d'actualisation	0	rf_rate	0 - Pas d'actualisation; 10 - 10 secondes; 30 - 30 secondes; 60 - *(par défaut)* 1 minute; 120 - 2 minutes; 600 - 10 minutes; 900 - 15 minutes.
Élément	4	itemid.0	ID de l'élément. Comportement du paramètre : - *obligatoire* si *Élément (widget)* n'est pas défini
Élément (widget)	1	itemid._reference	Au lieu de l'ID de l'élément : `ABCDE._itemid` - définir un widget compatible (avec son paramètre *Référence* défini sur "ABCDE") comme source de données pour les éléments. Comportement du paramètre : - *obligatoire* si *Élément* n'est pas défini

Paramètre	type	name	value
Afficher	0	show.0	1 - Description; 2 - Valeur; 3 - Heure; 4 - Indicateur de changement; 5 - Sparkline. Par défaut : 1, 2, 3, 4. Remarque : pour configurer plusieurs valeurs, créez un objet de champ de widget de tableau de bord pour chaque valeur avec un numéro incrémenté dans le nom de la propriété.
Remplacer l'hôte	1	override_hostid._reference	`ABCDE._hostid` - définir un widget compatible (avec son paramètre *Référence* défini sur "ABCDE") comme source de données pour les hôtes ; `DASHBOARD._hostid` - définir le sélecteur d'hôte du tableau de bord comme source de données pour les hôtes. Ce paramètre n'est pas pris en charge lors de la configuration du widget sur un tableau de bord de modèle.

Configuration avancée

Les paramètres de configuration avancée suivants sont pris en charge pour le widget *Valeur d'élément*.

Note:
Le nombre dans le nom de propriété *Thresholds* (par ex. thresholds.0.color) fait référence à la position du seuil dans une liste, triée par ordre croissant. Cependant, si les seuils sont configurés dans un ordre différent, les valeurs seront triées par ordre croissant après la mise à jour de la configuration du widget dans l'interface Zabbix (par ex. `"thresholds.0.threshold":"5"` → `"thresholds.0.threshold":"1"` ; `"thresholds.1.threshold":"1"` → `"thresholds.1.threshold": "5"`).

Paramètre	type	name	value
Couleur d'arrière-plan	1	bg_color	Code couleur hexadécimal (par ex. `FF0000`). Par défaut : `""` (vide).
Seuils			
Couleur	1	thresholds.0.color	Code couleur hexadécimal (par ex. `FF0000`).
Seuil	1	thresholds.0.threshold	Toute valeur de chaîne.
Fonction d'agrégation	0	aggregate_function	0 - *(par défaut)* non utilisée ; 1 - min ; 2 - max ; 3 - avg ; 4 - count ; 5 - sum ; 6 - first ; 7 - last.
Période	1	time_period._reference	`DASHBOARD._timeperiod` - définir le sélecteur de période du tableau de bord comme source de données ; `ABCDE._timeperiod` - définir un widget compatible (avec son paramètre `reference` égal à `ABCDE`) comme source de données. Par défaut : `DASHBOARD._timeperiod` Vous pouvez également définir la période uniquement dans les paramètres *From* et *To*. Comportement du paramètre : - *pris en charge* si *Fonction d'agrégation* est définie sur "min", "max", "avg", "count", "sum", "first", "last"

Paramètre	type	name	value
From	1	time_period.from	Chaîne temporelle valide en syntaxe absolue (YYYY-MM-DD hh:mm:ss) ou relative (now, now/d, now/w-1w, etc.). Comportement du paramètre : - *pris en charge* si *Période* n'est pas définie et si *Fonction d'agrégation* est définie sur "min", "max", "avg", "count", "sum", "first", "last" - *obligatoire* si time_period.to est défini
To	1	time_period.to	Chaîne temporelle valide en syntaxe absolue (YYYY-MM-DD hh:mm:ss) ou relative (now, now/d, now/w-1w, etc.). Comportement du paramètre : - *pris en charge* si *Période* n'est pas définie et si *Fonction d'agrégation* est définie sur "min", "max", "avg", "count", "sum", "first", "last" - *obligatoire* si time_period.from est défini
Données d'historique	0	history	0 - *(par défaut)* Auto ; 1 - Historique ; 2 - Tendances.

Description

Les paramètres de configuration avancés suivants sont pris en charge si *Afficher* est défini sur « Description ».

Paramètre	type	name	value
Description	1	description	Toute valeur de chaîne, y compris les macros. Macros prises en charge : {HOST.*}, {ITEM.*}, {INVENTORY.*}, macros utilisateur. Valeur par défaut : {ITEM.NAME}.
Position horizontale	0	desc_h_pos	0 - Gauche ; 1 - *(par défaut)* Centre ; 2 - Droite.
Position verticale	0	desc_v_pos	Deux éléments ou plus (Description, Valeur, Heure) ne peuvent pas partager la même *Position horizontale* et la même *Position verticale*. 0 - Haut ; 1 - Milieu ; 2 - *(par défaut)* Bas.
Taille	0	desc_size	Deux éléments ou plus (Description, Valeur, Heure) ne peuvent pas partager la même *Position horizontale* et la même *Position verticale*. Les valeurs possibles vont de 1 à 100. Valeur par défaut : 15.
Gras	0	desc_bold	0 - *(par défaut)* Désactivé ; 1 - Activé.
Couleur	1	desc_color	Code couleur hexadécimal (par exemple, FF0000). Valeur par défaut : "" (vide).

Valeur

Les paramètres de configuration avancés suivants sont pris en charge si *Afficher* est défini sur « Valeur ».

Parameter	type	name	value
Nombre de déci-males			
Nombre de décimales	0	decimal_places	Les valeurs possibles vont de 1 à 10. Valeur par défaut : 2.
Taille	0	decimal_size	Les valeurs possibles vont de 1 à 100. Valeur par défaut : 35.
Position			
Position horizontale	0	value_h_pos	0 - Gauche ; 1 - *(par défaut)* Centre ; 2 - Droite. Deux éléments ou plus (Description, Valeur, Heure) ne peuvent pas partager la même *Position horizontale* et la même *Position verticale*.
Position verticale	0	value_v_pos	0 - Haut ; 1 - *(par défaut)* Milieu ; 2 - Bas. Deux éléments ou plus (Description, Valeur, Heure) ne peuvent pas partager la même *Position horizontale* et la même *Position verticale*.
Taille	0	value_size	Les valeurs possibles vont de 1 à 100. Valeur par défaut : 45.
Gras	0	value_bold	0 - Désactivé ; 1 - *(par défaut)* Activé.
Couleur	1	value_color	Code couleur hexadécimal (par exemple, FF0000). Valeur par défaut : "" (vide).
Unités			
Unités (case à cocher)	0	units_show	0 - Désactivé ; 1 - *(par défaut)* Activé.
Unités (valeur)	1	units	Toute valeur de type chaîne.
Position	0	units_pos	0 - Avant la valeur ; 1 - Au-dessus de la valeur ; 2 - *(par défaut)* Après la valeur ; 3 - En dessous de la valeur.
Taille	0	units_size	Les valeurs possibles vont de 1 à 100. Valeur par défaut : 35.
Gras	0	units_bold	0 - Désactivé ; 1 - *(par défaut)* Activé.
Couleur	1	units_color	Code couleur hexadécimal (par exemple, FF0000). Valeur par défaut : "" (vide).

Heure

Les paramètres de configuration avancés suivants sont pris en charge si *Afficher* est défini sur « Heure ».

Parameter	type	name	value
Position horizontale	0	time_h_pos	0 - À gauche ; 1 - *(par défaut)* Au centre ; 2 - À droite. Deux éléments ou plus (Description, Valeur, Heure) ne peuvent pas partager la même *Position horizontale* et la même *Position verticale*.

Parameter	type	name	value
Position verticale	0	time_v_pos	0 - *(par défaut)* En haut ; 1 - Au milieu ; 2 - En bas. Deux éléments ou plus (Description, Valeur, Heure) ne peuvent pas partager la même *Position horizontale* et la même *Position verticale*.
Taille	0	time_size	Les valeurs possibles vont de 1 à 100. Valeur par défaut : 15.
Gras	0	time_bold	0 - *(par défaut)* Désactivé ; 1 - Activé.
Couleur	1	time_color	Code couleur hexadécimal (par exemple, FF0000). Valeur par défaut : "" (vide).

Indicateur de changement

Les paramètres de configuration avancée suivants sont pris en charge si *Afficher* est défini sur « Indicateur de changement ».

Paramètre	type	name	value
Couleur de l'indicateur de changement ↑	1	up_color	Code couleur hexadécimal (par exemple FF0000). Par défaut : "" (vide).
Couleur de l'indicateur de changement ↓	1	down_color	Code couleur hexadécimal (par exemple FF0000). Par défaut : "" (vide).
Couleur de l'indicateur de changement ↕	1	updown_color	Code couleur hexadécimal (par exemple FF0000). Par défaut : "" (vide).

Sparkline

Les paramètres de configuration avancés suivants sont pris en charge si *Afficher* est défini sur « Sparkline ».

Paramètre	type	name	value
Largeur	0	sparkline.width	Les valeurs possibles vont de 0 à 10. Valeur par défaut : 1.
Remplissage	0	sparkline.fill	Les valeurs possibles vont de 0 à 10. Valeur par défaut : 3.
Couleur	1	sparkline.color	Code couleur hexadécimal (par exemple, FF0000). Valeur par défaut : 42A5F5.
Période	1	sparkline.time_period.reference	DASHBOARD._timeperiod - définir le sélecteur de période du tableau de bord comme source de données ; ABCDE._timeperiod - définir un widget compatible (avec son paramètre reference défini sur ABCDE) comme source de données. Valeur par défaut : DASHBOARD._timeperiod Vous pouvez également définir la période uniquement dans les paramètres *Du* et *Au*.

Paramètre	type	name	value
Du	1	sparkline.time_period.from	Chaîne temporelle valide au format absolu (`YYYY-MM-DD hh:mm:ss`) ou au format temporel relatif (`now`, `now/d`, `now/w-1w`, etc.). Comportement du paramètre : - *obligatoire* si `sparkline.time_period.to` est défini
Au	1	sparkline.time_period.to	Chaîne temporelle valide au format absolu (`YYYY-MM-DD hh:mm:ss`) ou au format temporel relatif (`now`, `now/d`, `now/w-1w`, etc.). Comportement du paramètre : - *obligatoire* si `sparkline.time_period.from` est défini
Données d'historique	0	sparkline.history	0 - *(par défaut)* Automatique ; 1 - Historique ; 2 - Tendances.

Exemples

Les exemples suivants visent à décrire uniquement la configuration des objets de champ du widget de tableau de bord pour le widget *Valeur d'élément*. Pour plus d'informations sur la configuration d'un tableau de bord, consultez `dashboard.create`.

Configuration d'un widget *Valeur d'élément*

Configurez un widget *Valeur d'élément* qui affiche la valeur de l'élément pour l'élément « 42266 » (disponibilité de l'agent Zabbix). De plus, ajustez finement l'apparence du widget à l'aide de plusieurs options avancées, notamment une couleur d'arrière-plan dynamique qui change en fonction de l'état de disponibilité de l'agent Zabbix.

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": 30,
        "auto_start": 1,
        "pages": [
            {
                "widgets": [
                    {
                        "type": "item",
                        "name": "Item value",
                        "x": 0,
                        "y": 0,
                        "width": 12,
                        "height": 3,
                        "view_mode": 0,
                        "fields": [
                            {
                                "type": 4,
                                "name": "itemid.0",
                                "value": 42266
                            },
                            {
                                "type": 0,
                                "name": "show.0",
                                "value": 1
                            },
                            {
                                "type": 0,
                                "name": "show.1",
                                "value": 2
                            },
```

```
                {
                    "type": 0,
                    "name": "show.2",
                    "value": 3
                },
                {
                    "type": 1,
                    "name": "description",
                    "value": "Agent status"
                },
                {
                    "type": 0,
                    "name": "desc_h_pos",
                    "value": 0
                },
                {
                    "type": 0,
                    "name": "desc_v_pos",
                    "value": 0
                },
                {
                    "type": 0,
                    "name": "desc_bold",
                    "value": 1
                },
                {
                    "type": 1,
                    "name": "desc_color",
                    "value": "F06291"
                },
                {
                    "type": 0,
                    "name": "value_h_pos",
                    "value": 0
                },
                {
                    "type": 0,
                    "name": "value_size",
                    "value": 25
                },
                {
                    "type": 1,
                    "name": "value_color",
                    "value": "FFFF00"
                },
                {
                    "type": 0,
                    "name": "units_show",
                    "value": 0
                },
                {
                    "type": 0,
                    "name": "time_h_pos",
                    "value": 2
                },
                {
                    "type": 0,
                    "name": "time_v_pos",
                    "value": 2
                },
                {
                    "type": 0,
```

```
                                "name": "time_size",
                                "value": 10
                            },
                            {
                                "type": 0,
                                "name": "time_bold",
                                "value": 1
                            },
                            {
                                "type": 1,
                                "name": "time_color",
                                "value": "9FA8DA"
                            },
                            {
                                "type": 1,
                                "name": "thresholds.0.color",
                                "value": "E1E1E1"
                            },
                            {
                                "type": 1,
                                "name": "thresholds.0.threshold",
                                "value": "0"
                            },
                            {
                                "type": 1,
                                "name": "thresholds.1.color",
                                "value": "D1C4E9"
                            },
                            {
                                "type": 1,
                                "name": "thresholds.1.threshold",
                                "value": "1"
                            }
                        ]
                    }
                ]
            }
        ],
        "userGroups": [
            {
                "usrgrpid": 7,
                "permission": 2
            }
        ],
        "users": [
            {
                "userid": 1,
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
```

```
    "id": 1
}
```

Voir également

- Champ du widget du tableau de bord
- `dashboard.create`
- `dashboard.update`

19 Carte

Description

Ces paramètres et les valeurs de propriété possibles pour les objets de champ respectifs du widget de tableau de bord vous permettent de configurer le widget *Map* dans les méthodes `dashboard.create` et `dashboard.update`.

Attention:
Les propriétés `fields` du widget ne sont pas validées lors de la création ou de la mise à jour d'un tableau de bord. Cela permet aux utilisateurs de modifier les widgets intégrés et de créer des widgets personnalisés, mais introduit également le risque de créer ou de mettre à jour des widgets de manière incorrecte. Pour garantir la création ou la mise à jour réussie du widget *Map*, veuillez vous reporter au comportement des paramètres décrit dans les tableaux ci-dessous.

Paramètres

Les paramètres suivants sont pris en charge pour le widget *Carte*.

Paramètre	type	name	value
Intervalle d'actualisation	0	rf_rate	0 - Aucune actualisation; 10 - 10 secondes; 30 - 30 secondes; 60 - 1 minute; 120 - 2 minutes; 600 - 10 minutes; 900 - *(par défaut)* 15 minutes.
Carte	8	sysmapid.0	ID de la carte. Comportement du paramètre : - *obligatoire* si *Carte (widget)* n'est pas défini
Carte (widget)	1	sysmapid._reference	`ABCDE._mapid` - définit un widget *Arborescence de navigation des cartes* (avec son paramètre *Référence* défini sur "`ABCDE`") comme source de données pour les cartes. Comportement du paramètre : - *obligatoire* si *Carte* n'est pas défini
Référence	1	reference	Toute valeur de chaîne composée de 5 caractères (par exemple, `ABCDE` ou `JBPNL`). Cette valeur doit être unique dans le tableau de bord auquel le widget appartient. Comportement du paramètre : - *obligatoire*

Exemples

Les exemples suivants visent à décrire uniquement la configuration des objets de champ du widget de tableau de bord pour le widget *Carte*. Pour plus d'informations sur la configuration d'un tableau de bord, consultez `dashboard.create`.

Configuration d'un widget *Map*

Configurez un widget *Map* qui affiche la carte « 1 ».

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": 30,
        "auto_start": 1,
        "pages": [
            {
                "widgets": [
                    {
                        "type": "map",
                        "name": "Map",
                        "x": 0,
                        "y": 0,
                        "width": 54,
                        "height": 5,
                        "view_mode": 0,
                        "fields": [
                            {
                                "type": 8,
                                "name": "sysmapid.0",
                                "value": 1
                            }
                        ]
                    }
                ]
            }
        ],
        "userGroups": [
            {
                "usrgrpid": 7,
                "permission": 2
            }
        ],
        "users": [
            {
                "userid": 1,
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Configuration d'un widget *Map* lié

Configurez un widget *Map* lié à un widget *Map navigation tree*.

Requête :

```
{
    "jsonrpc": "2.0",
```

```
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": 30,
        "auto_start": 1,
        "pages": [
            {
                "widgets": [
                    {
                        "type": "map",
                        "name": "Map",
                        "x": 0,
                        "y": 5,
                        "width": 54,
                        "height": 5,
                        "view_mode": 0,
                        "fields": [
                            {
                                "type": 1,
                                "name": "sysmapid._reference",
                                "value": "ABCDE._mapid"
                            }
                        ]
                    },
                    {
                        "type": "navtree",
                        "name": "Map navigation tree",
                        "x": 0,
                        "y": 0,
                        "width": 18,
                        "height": 5,
                        "view_mode": 0,
                        "fields": [
                            {
                                "type": 1,
                                "name": "navtree.1.name",
                                "value": "Element A"
                            },
                            {
                                "type": 1,
                                "name": "navtree.2.name",
                                "value": "Element B"
                            },
                            {
                                "type": 1,
                                "name": "navtree.3.name",
                                "value": "Element C"
                            },
                            {
                                "type": 1,
                                "name": "navtree.4.name",
                                "value": "Element A1"
                            },
                            {
                                "type": 1,
                                "name": "navtree.5.name",
                                "value": "Element A2"
                            },
                            {
                                "type": 1,
                                "name": "navtree.6.name",
                                "value": "Element B1"
```

```
    },
    {
        "type": 1,
        "name": "navtree.7.name",
        "value": "Element B2"
    },
    {
        "type": 0,
        "name": "navtree.4.parent",
        "value": 1
    },
    {
        "type": 0,
        "name": "navtree.5.parent",
        "value": 1
    },
    {
        "type": 0,
        "name": "navtree.6.parent",
        "value": 2
    },
    {
        "type": 0,
        "name": "navtree.7.parent",
        "value": 2
    },
    {
        "type": 0,
        "name": "navtree.1.order",
        "value": 1
    },
    {
        "type": 0,
        "name": "navtree.2.order",
        "value": 2
    },
    {
        "type": 0,
        "name": "navtree.3.order",
        "value": 3
    },
    {
        "type": 0,
        "name": "navtree.4.order",
        "value": 1
    },
    {
        "type": 0,
        "name": "navtree.5.order",
        "value": 2
    },
    {
        "type": 0,
        "name": "navtree.6.order",
        "value": 1
    },
    {
        "type": 0,
        "name": "navtree.7.order",
        "value": 2
    },
    {
```

```
                                "type": 8,
                                "name": "navtree.6.sysmapid",
                                "value": 1
                            },
                            {
                                "type": 1,
                                "name": "reference",
                                "value": "ABCDE"
                            }
                        ]
                    }
                ]
            }
        ],
        "userGroups": [
            {
                "usrgrpid": 7,
                "permission": 2
            }
        ],
        "users": [
            {
                "userid": 1,
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Voir également

- Champ du widget du tableau de bord
- `dashboard.create`
- `dashboard.update`
- Arbre de navigation de carte

20 Arborescence de navigation des cartes

Description

Ces paramètres et les valeurs de propriété possibles pour les objets de champ respectifs du widget de tableau de bord vous permettent de configurer le widget *Arborescence de navigation des cartes* dans les méthodes `dashboard.create` et `dashboard.update`.

Attention:
Les propriétés `fields` du widget ne sont pas validées lors de la création ou de la mise à jour d'un tableau de bord. Cela permet aux utilisateurs de modifier les widgets intégrés et de créer des widgets personnalisés, mais introduit également le risque de créer ou de mettre à jour des widgets de manière incorrecte. Pour garantir la création ou la mise à jour réussie du widget *Arborescence de navigation des cartes*, veuillez vous référer au comportement des paramètres décrit dans les tableaux ci-dessous.

Paramètres

Les paramètres suivants sont pris en charge pour le widget *Arborescence de navigation des cartes*.

Paramètre	type	name	value
Intervalle d'actualisation	0	rf_rate	0 - Aucune actualisation; 10 - 10 secondes; 30 - 30 secondes; 60 - 1 minute; 120 - 2 minutes; 600 - 10 minutes; 900 - *(par défaut)* 15 minutes.
Afficher les cartes indisponibles	1	show_unavailable	0 - *(par défaut)* Désactivé; 1 - Activé.
Référence	1	reference	Toute valeur de chaîne composée de 5 caractères (par exemple, `ABCDE` ou `JBPNL`). Cette valeur doit être unique dans le tableau de bord auquel appartient le widget. Comportement des paramètres : - *obligatoire*

Les paramètres suivants sont pris en charge pour configurer les éléments de l'arborescence de navigation des cartes.

Paramètre	type	name	value
Nom	1	navtree.1.name	Toute valeur de chaîne. Remarque : Le numéro dans le nom de la propriété définit le numéro de l'élément.
Carte liée	8	navtree.1.sysmapid	ID de la carte. Remarque : Le numéro dans le nom de la propriété fait référence à l'élément auquel la carte est liée.
Paramètres de création de la hiérarchie des éléments	0	navtree.1.parent	Numéro de l'élément parent. Remarque : Le numéro dans le nom de la propriété fait référence à l'élément enfant. La valeur de la propriété fait référence à l'élément parent.
	0	navtree.1.order	Position de l'élément dans l'arborescence de navigation des cartes. Remarque : Le numéro dans le nom de la propriété fait référence au numéro de l'élément. La valeur de la propriété fait référence à la position de l'élément dans l'arborescence de navigation des cartes. La position de l'élément parent est déterminée dans l'ensemble de l'arborescence de navigation des cartes. La position de l'élément enfant est déterminée au sein de l'élément parent.

Exemples

Les exemples suivants visent à décrire uniquement la configuration des objets de champ du widget de tableau de bord pour le widget *Arborescence de navigation de carte*. Pour plus d'informations sur la configuration d'un tableau de bord, consultez `dashboard.create`.

Configuration d'un widget *Map navigation tree*

Configurez un widget *Map navigation tree* qui affiche l'arborescence de navigation de cartes suivante :

- Élément A
  - Élément A1
  - Élément A2
- Élément B
  - Élément B1 (contient la carte liée « 1 » qui peut être affichée dans un widget *Map* lié)
  - Élément B2
- Élément C

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": 30,
        "auto_start": 1,
        "pages": [
            {
                "widgets": [
                    {
                        "type": "navtree",
                        "name": "Map navigation tree",
                        "x": 0,
                        "y": 0,
                        "width": 18,
                        "height": 5,
                        "view_mode": 0,
                        "fields": [
                            {
                                "type": 1,
                                "name": "navtree.1.name",
                                "value": "Element A"
                            },
                            {
                                "type": 1,
                                "name": "navtree.2.name",
                                "value": "Element B"
                            },
                            {
                                "type": 1,
                                "name": "navtree.3.name",
                                "value": "Element C"
                            },
                            {
                                "type": 1,
                                "name": "navtree.4.name",
                                "value": "Element A1"
                            },
                            {
                                "type": 1,
                                "name": "navtree.5.name",
                                "value": "Element A2"
                            },
                            {
                                "type": 1,
                                "name": "navtree.6.name",
                                "value": "Element B1"
                            },
                            {
                                "type": 1,
                                "name": "navtree.7.name",
                                "value": "Element B2"
                            },
```

```
                {
                    "type": 0,
                    "name": "navtree.4.parent",
                    "value": 1
                },
                {
                    "type": 0,
                    "name": "navtree.5.parent",
                    "value": 1
                },
                {
                    "type": 0,
                    "name": "navtree.6.parent",
                    "value": 2
                },
                {
                    "type": 0,
                    "name": "navtree.7.parent",
                    "value": 2
                },
                {
                    "type": 0,
                    "name": "navtree.1.order",
                    "value": 1
                },
                {
                    "type": 0,
                    "name": "navtree.2.order",
                    "value": 2
                },
                {
                    "type": 0,
                    "name": "navtree.3.order",
                    "value": 3
                },
                {
                    "type": 0,
                    "name": "navtree.4.order",
                    "value": 1
                },
                {
                    "type": 0,
                    "name": "navtree.5.order",
                    "value": 2
                },
                {
                    "type": 0,
                    "name": "navtree.6.order",
                    "value": 1
                },
                {
                    "type": 0,
                    "name": "navtree.7.order",
                    "value": 2
                },
                {
                    "type": 8,
                    "name": "navtree.6.sysmapid",
                    "value": 1
                },
                {
                    "type": 1,
```

```
                                "name": "reference",
                                "value": "HJQXF"
                            }
                        ]
                    }
                ]
            }
        ],
        "userGroups": [
            {
                "usrgrpid": 7,
                "permission": 2
            }
        ],
        "users": [
            {
                "userid": 1,
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Voir également

- Champ du widget du tableau de bord
- `dashboard.create`
- `dashboard.update`
- Carte

21 Diagramme circulaire

Description

Ces paramètres et les valeurs de propriété possibles pour les objets de champ respectifs du widget de tableau de bord vous permettent de configurer le widget *Graphique circulaire* dans les méthodes `dashboard.create` et `dashboard.update`.

Attention:
Les propriétés `fields` du widget ne sont pas validées lors de la création ou de la mise à jour d'un tableau de bord. Cela permet aux utilisateurs de modifier les widgets intégrés et de créer des widgets personnalisés, mais introduit également le risque de créer ou de mettre à jour des widgets de manière incorrecte. Pour garantir la création ou la mise à jour réussie du widget *Graphique circulaire*, veuillez vous reporter au comportement des paramètres décrit dans les tableaux ci-dessous.

Paramètres

Les paramètres suivants sont pris en charge pour le widget *Diagramme circulaire*.

Paramètre	type	name	value
Intervalle d'actualisation	0	rf_rate	0 - Aucune actualisation; 10 - 10 secondes; 30 - 30 secondes; 60 - *(par défaut)* 1 minute; 120 - 2 minutes; 600 - 10 minutes; 900 - 15 minutes.

Ensemble de données

Les paramètres suivants sont pris en charge pour configurer un *Ensemble de données*.

Note:
Le premier nombre dans le nom de la propriété (par exemple, ds.0.hosts.0, ds.0.items.0) représente l'ensemble de données concerné, tandis que le deuxième nombre, s'il est présent, représente le hôte ou l'élément configuré.

Parameter	type	name	value
Type d'ensemble de données	0	ds.0.dataset_type	0 - Liste d'éléments ; 1 - Motifs d'éléments *(par défaut)*.
Éléments	4	ds.0.itemids.0	ID de l'élément. Lors de la configuration du widget sur un tableau de bord de modèle, seuls les éléments configurés sur le modèle doivent être définis. Remarque : pour configurer plusieurs éléments, créez un objet de champ de widget de tableau de bord pour chaque élément, avec un nombre incrémenté dans le nom de la propriété. Comportement du paramètre : - *obligatoire* si le *Type d'ensemble de données* est défini sur « Liste d'éléments » et si *Éléments (Widget)* n'est pas défini
Éléments (Widget)	1	ds.0.itemids.0._reference	Au lieu de l'ID de l'élément : `ABCDE._itemid` - définissez un widget compatible (avec son paramètre *Référence* défini sur « `ABCDE` ») comme source de données des éléments. Remarque : pour configurer plusieurs widgets, créez un objet de champ de widget de tableau de bord pour chaque widget, avec un nombre incrémenté dans le nom de la propriété. Comportement du paramètre : - *obligatoire* si le *Type d'ensemble de données* est défini sur « Liste d'éléments » et si *Éléments* n'est pas défini
Couleur	1	ds.0.color.0	Code couleur hexadécimal (par exemple, `FF0000`). Comportement du paramètre : - *obligatoire* si le *Type d'ensemble de données* est défini sur « Liste d'éléments »
Type d'élément	0	ds.0.type.0	0 - Normal *(par défaut)* ; 1 - Total. La valeur « Total » ne peut être définie que pour un seul élément dans l'ensemble du graphique. Comportement du paramètre : - *pris en charge* si le *Type d'ensemble de données* est défini sur « Liste d'éléments »

Parameter	type	name	value
Motifs de hôtes	1	ds.0.hosts.0	Nom ou motif de hôte (par exemple, « Zabbix* »). Comportement du paramètre : - *obligatoire* si le *Type d'ensemble de données* est défini sur « Motifs d'éléments » Ce paramètre n'est pas pris en charge lors de la configuration du widget sur un tableau de bord de modèle.
Motifs d'éléments	1	ds.0.items.0	Nom ou motif d'élément (par exemple, « *: Nombre de *valeurs traitées par seconde »). Lors de la configuration du widget sur un tableau de bord de modèle, seuls les motifs des éléments configurés sur le modèle doivent être définis. Comportement du paramètre : - *obligatoire* si le *Type d'ensemble de données* est défini sur « Motifs d'éléments »
Couleur	1	ds.0.color	Code couleur hexadécimal (par exemple, FF0000). Comportement du paramètre : - *pris en charge* si le *Type d'ensemble de données* est défini sur « Motifs d'éléments » et si *Palette de couleurs* n'est pas défini
Palette de couleurs	0	ds.0.color_palette	Index de la palette de couleurs. Les valeurs possibles vont de 0 à 11. Valeur par défaut : 0. Comportement du paramètre : - *pris en charge* si le *Type d'ensemble de données* est défini sur « Motifs d'éléments » et si *Couleur* n'est pas défini
Fonction d'agrégation	0	ds.0.aggregate_function	1 - min ; 2 - max ; 3 - moyenne ; 4 - nombre ; 5 - somme ; 6 - premier ; 7 - dernier *(par défaut)*.
Agrégation de l'ensemble de données	0	ds.0.dataset_aggregation	0 - aucune *(par défaut)* ; 1 - min ; 2 - max ; 3 - moyenne ; 4 - nombre ; 5 - somme. Comportement du paramètre : - *pris en charge* si le *Type d'élément* est défini sur « Total »
Libellé de l'ensemble de données	1	ds.0.data_set_label	Toute valeur de chaîne. Valeur par défaut : "" (vide).
Étiquettes d'éléments			
Type d'évaluation	0	ds.0.item_tags_evaltype	0 - Et/Ou *(par défaut)* ; 2 - Ou.

Parameter	type	name	value
Nom de l'étiquette	1	ds.0.item_tags.0.tag	Toute valeur de chaîne. Remarque : le nombre dans le nom de la propriété fait référence à l'ordre de l'étiquette dans la liste d'évaluation des étiquettes. Comportement du paramètre : - *obligatoire* lors de la configuration des *Étiquettes d'éléments*
Opérateur	0	ds.0.item_tags.0.operator	0 - Contient ; 1 - Est égal à ; 2 - Ne contient pas ; 3 - N'est pas égal à ; 4 - Existe ; 5 - N'existe pas. Remarque : le nombre dans le nom de la propriété fait référence à l'ordre de l'étiquette dans la liste d'évaluation des étiquettes. Comportement du paramètre : - *obligatoire* lors de la configuration des *Étiquettes d'éléments*
Valeur de l'étiquette	1	ds.0.item_tags.0.value	Toute valeur de chaîne. Remarque : le nombre dans le nom de la propriété fait référence à l'ordre de l'étiquette dans la liste d'évaluation des étiquettes. Comportement du paramètre : - *obligatoire* lors de la configuration des *Étiquettes d'éléments*

Options d'affichage

Les paramètres suivants sont pris en charge pour configurer les *options d'affichage*.

Paramètre	type	name	value
Sélection des données d'historique	0	source	0 - *(par défaut)* Automatique ; 1 - Historique ; 2 - Tendances.
Dessiner	0	draw_type	0 - *(par défaut)* Secteurs ; 1 - Anneau.
Largeur	0	width	20 - 20 % du rayon ; 30 - 30 % du rayon ; 40 - 40 % du rayon ; 50 - *(par défaut)* 50 % du rayon. Comportement du paramètre : - *pris en charge* si *Dessiner* est défini sur « Anneau »
Largeur du contour	0	stroke	Les valeurs possibles vont de 0 à 10. Valeur par défaut : 0. Comportement du paramètre : - *pris en charge* si *Dessiner* est défini sur « Anneau »
Afficher la valeur totale	0	total_show	0 - *(par défaut)* Désactivé ; 1 - Activé. Comportement du paramètre : - *pris en charge* si *Dessiner* est défini sur « Anneau »

Paramètre	type	name	value
Taille	0	value_size_type	0 - *(par défaut)* Automatique ; 1 - Personnalisée.
Taille (valeur pour la taille personnalisée)	0	value_size	Comportement du paramètre : - *pris en charge* si *Afficher la valeur totale* est défini sur « Activé » Les valeurs possibles vont de 1 à 100. Valeur par défaut : 20.
Décimales	0	decimal_places	Comportement du paramètre : - *pris en charge* si *Afficher la valeur totale* est défini sur « Activé » Les valeurs possibles vont de 0 à 6. Valeur par défaut : 2.
Unités (case à cocher)	0	units_show	Comportement du paramètre : - *pris en charge* si *Afficher la valeur totale* est défini sur « Activé » 0 - *(par défaut)* Désactivé ; 1 - Activé.
Unités (valeur)	1	units	Comportement du paramètre : - *pris en charge* si *Afficher la valeur totale* est défini sur « Activé » Toute valeur de type chaîne.
Gras	0	value_bold	Comportement du paramètre : - *pris en charge* si *Unités* (case à cocher) est défini sur « Activé » 0 - *(par défaut)* Désactivé ; 1 - Activé.
Couleur	1	value_color	Comportement du paramètre : - *pris en charge* si *Afficher la valeur totale* est défini sur « Activé » Code couleur hexadécimal (par exemple, FF0000).
Espace entre les secteurs	0	space	Comportement du paramètre : - *pris en charge* si *Afficher la valeur totale* est défini sur « Activé » Les valeurs possibles vont de 0 à 10. Valeur par défaut : 1.
Fusionner les secteurs inférieurs à N % (case à cocher)	0	merge	0 - *(par défaut)* Désactivé ; 1 - Activé.
Fusionner les secteurs inférieurs à N % (valeur)	0	merge_percent	Les valeurs possibles vont de 1 à 10. Valeur par défaut : 1. Comportement du paramètre : - *pris en charge* si *Fusionner les secteurs inférieurs à N %* (case à cocher) est défini sur « Activé »

Paramètre	type	name	value
Fusionner les secteurs inférieurs à N % (couleur)	1	merge_color	Code couleur hexadécimal (par exemple, FF0000). Comportement du paramètre : - *pris en charge* si *Fusionner les secteurs inférieurs à N %* (case à cocher) est défini sur « Activé »

Période de temps

Les paramètres suivants sont pris en charge pour configurer la *Période de temps*.

Paramètre	type	name	value
Période de temps	1	time_period._reference	DASHBOARD._timeperiod - définit le sélecteur de période de temps du tableau de bord comme source de données ; ABCDE._timeperiod - définit un widget compatible (avec son paramètre *Reference* défini sur "ABCDE") comme source de données. Par défaut : DASHBOARD._timeperiod Vous pouvez également définir la période de temps uniquement dans les paramètres *From* et *To*.
From	1	time_period.from	Chaîne de temps valide en syntaxe temporelle absolue (YYYY-MM-DD hh:mm:ss) ou relative (now, now/d, now/w-1w, etc.). Comportement du paramètre : - *pris en charge* si *Période de temps* n'est pas défini - *obligatoire* si time_period.to est défini
To	1	time_period.to	Chaîne de temps valide en syntaxe temporelle absolue (YYYY-MM-DD hh:mm:ss) ou relative (now, now/d, now/w-1w, etc.). Comportement du paramètre : - *pris en charge* si *Période de temps* n'est pas défini - *obligatoire* si time_period.from est défini

Légende

Les paramètres suivants sont pris en charge pour configurer la *Légende*.

Paramètre	type	name	value
Afficher la légende	0	legend	0 - Désactivé ; 1 - *(par défaut)* Activé.
Afficher la valeur	0	legend_value	0 - *(par défaut)* Désactivé ; 1 - Activé. Comportement du paramètre : - *pris en charge* si *Afficher la légende* est défini sur « Activé »
Afficher la fonction d'agrégation	0	legend_aggregation	0 - *(par défaut)* Désactivé ; 1 - Activé. Comportement du paramètre : - *pris en charge* si *Afficher la légende* est défini sur « Activé »

Paramètre	type	name	value
Lignes	0	legend_lines_mode	0 - *(par défaut)* Fixe ; 1 - Variable.
Nombre de lignes/ Nombre maximal de lignes	0	legend_lines	Comportement du paramètre : - *pris en charge* si *Afficher la légende* est défini sur « Activé » Les valeurs possibles vont de 1 à 10. Valeur par défaut : 1.
Nombre de colonnes	0	legend_columns	Comportement du paramètre : - *pris en charge* si *Afficher la légende* est défini sur « Activé » Les valeurs possibles vont de 1 à 4. Valeur par défaut : 4.
			Comportement du paramètre : - *pris en charge* si *Afficher la légende* est défini sur « Activé » et *Afficher la valeur* sur « Désactivé »

Exemples

Les exemples suivants visent uniquement à décrire la configuration des objets de champ de widget de tableau de bord pour le widget *Pie chart*. Pour plus d'informations sur la configuration d'un tableau de bord, consultez `dashboard.create`.

Configuration d'un widget *Camembert*

Configurez un widget *Camembert* de la manière suivante :

- 2 jeux de données pour un total de 9 éléments sur 1 hôte.
- Le premier jeu de données est de type « Liste d'éléments » et se compose de 3 éléments, tous de type « Normal » et représentés par une couleur différente.
- Le second jeu de données est de type « Modèles d'éléments », se compose de 6 éléments, possède une agrégation configurée pour chaque élément et est représenté par une couleur personnalisée.
- Le second jeu de données possède également un libellé de jeu de données personnalisé.
- Les données du graphique circulaire sont affichées sous forme de diagramme en anneau avec une largeur personnalisée et la valeur totale avec les unités au centre.
- Les données du graphique circulaire sont affichées et agrégées pour une période personnalisée correspondant aux 3 dernières heures.
- La légende du graphique circulaire affiche les éléments configurés sur 4 lignes.

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": 30,
        "auto_start": 1,
        "pages": [
            {
                "widgets": [
                    {
                        "type": "piechart",
                        "name": "Pie chart",
                        "x": 0,
                        "y": 0,
                        "width": 24,
                        "height": 5,
                        "view_mode": 0,
                        "fields": [
                            {
                                "type": 0,
                                "name": "ds.0.dataset_type",
```

```
            "value": 0
        },
        {
            "type": 4,
            "name": "ds.0.itemids.0",
            "value": 23264
        },
        {
            "type": 1,
            "name": "ds.0.color.0",
            "value": "FF0000"
        },
        {
            "type": 0,
            "name": "ds.0.type.0",
            "value": 0
        },
        {
            "type": 4,
            "name": "ds.0.itemids.1",
            "value": 23269
        },
        {
            "type": 1,
            "name": "ds.0.color.1",
            "value": "BF00FF"
        },
        {
          "type": 0,
          "name": "ds.0.type.1",
          "value": 0
        },
        {
            "type": 4,
            "name": "ds.0.itemids.2",
            "value": 23257
        },
        {
            "type": 1,
            "name": "ds.0.color.2",
            "value": "0040FF"
        },
        {
          "type": 0,
          "name": "ds.0.type.2",
          "value": 0
        },
        {
            "type": 1,
            "name": "ds.1.hosts.0",
            "value": "Zabbix server"
        },
        {
            "type": 1,
            "name": "ds.1.items.0",
            "value": "*: Number of processed *values per second"
        },
        {
            "type": 1,
            "name": "ds.1.color",
            "value": "000000"
        },
```

```
                        {
                            "type": 0,
                            "name": "ds.1.aggregate_function",
                            "value": 3
                        },
                        {
                            "type": 1,
                            "name": "ds.1.data_set_label",
                            "value": "Number of processed values per second"
                        },
                        {
                            "type": 0,
                            "name": "draw_type",
                            "value": 1
                        },
                        {
                            "type": 0,
                            "name": "width",
                            "value": 30
                        },
                        {
                            "type": 0,
                            "name": "total_show",
                            "value": 1
                        },
                        {
                            "type": 0,
                            "name": "units_show",
                            "value": 1
                        },
                        {
                            "type": 0,
                            "name": "graph_time",
                            "value": 1
                        },
                        {
                            "type": 1,
                            "name": "time_period.from",
                            "value": "now-3h"
                        },
                        {
                            "type": 1,
                            "name": "time_period.to",
                            "value": "now"
                        },
                        {
                            "type": 0,
                            "name": "legend_lines",
                            "value": 4
                        }
                    ]
                }
            ]
        }
    ],
    "userGroups": [
        {
            "usrgrpid": 7,
            "permission": 2
        }
    ],
    "users": [
```

```
            {
                "userid": 1,
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Voir aussi

- Champ de widget de tableau de bord
- `dashboard.create`
- `dashboard.update`

22 Hôtes en problème

Description

Ces paramètres et les valeurs de propriété possibles pour les objets de champ respectifs du widget de tableau de bord vous permettent de configurer le widget *Hôtes avec problèmes* dans les méthodes `dashboard.create` et `dashboard.update`.

Attention:
Les propriétés `fields` du widget ne sont pas validées lors de la création ou de la mise à jour d'un tableau de bord. Cela permet aux utilisateurs de modifier les widgets intégrés et de créer des widgets personnalisés, mais introduit également le risque de créer ou de mettre à jour des widgets de manière incorrecte. Pour garantir la création ou la mise à jour réussie du widget *Hôtes avec problèmes*, veuillez vous reporter au comportement des paramètres décrit dans les tableaux ci-dessous.

Paramètres

Les paramètres suivants sont pris en charge pour le widget *Hôtes avec problèmes*.

Paramètre	type	name	value
Intervalle d'actualisation	0	rf_rate	0 - Pas d'actualisation ; 10 - 10 secondes ; 30 - 30 secondes ; 60 - *(par défaut)* 1 minute ; 120 - 2 minutes ; 600 - 10 minutes ; 900 - 15 minutes.
Groupes d'hôtes	2	groupids.0	ID du groupe d'hôtes. Remarque : pour configurer plusieurs groupes d'hôtes, créez un objet de champ de widget de tableau de bord pour chaque groupe d'hôtes avec un numéro incrémenté dans le nom de propriété. Ce paramètre n'est pas pris en charge lors de la configuration du widget sur un tableau de bord de modèle.

Paramètre	type	name	value
Groupes d'hôtes (Widget)	1	groupids._reference	Au lieu de l'ID du groupe d'hôtes : `ABCDE._hostgroupids` - définir un widget compatible (avec son paramètre *Référence* défini sur "`ABCDE`") comme source de données pour les groupes d'hôtes. Ce paramètre n'est pas pris en charge lors de la configuration du widget sur un tableau de bord de modèle.
Exclure les groupes d'hôtes	2	exclude_groupids.0	ID du groupe d'hôtes. Remarque : pour exclure plusieurs groupes d'hôtes, créez un objet de champ de widget de tableau de bord pour chaque groupe d'hôtes avec un numéro incrémenté dans le nom de propriété. Ce paramètre n'est pas pris en charge lors de la configuration du widget sur un tableau de bord de modèle.
Hôtes	3	hostids.0	ID de l'hôte. Remarque : pour configurer plusieurs hôtes, créez un objet de champ de widget de tableau de bord pour chaque hôte avec un numéro incrémenté dans le nom de propriété. Pour plusieurs hôtes, le paramètre *Groupes d'hôtes* ne doit soit pas être configuré du tout, soit être configuré avec au moins un groupe d'hôtes auquel appartiennent les hôtes configurés. Ce paramètre n'est pas pris en charge lors de la configuration du widget sur un tableau de bord de modèle.
Hôtes (Widget/Tableau de bord)	1	hostids._reference	Au lieu de l'ID de l'hôte : `DASHBOARD.hostids` - définir le sélecteur d'hôte du tableau de bord comme source de données pour les hôtes ; `ABCDE._hostids` - définir un widget compatible (avec son paramètre *Référence* défini sur "`ABCDE`") comme source de données pour les hôtes. Ce paramètre n'est pas pris en charge lors de la configuration du widget sur un tableau de bord de modèle.
Problème	1	problem	Nom d'événement du problème (insensible à la casse, nom complet ou partie de celui-ci).
Sévérité	0	severities.0	0 - Non classé ; 1 - Information ; 2 - Avertissement ; 3 - Moyen ; 4 - Élevé ; 5 - Désastre. Par défaut : vide (tous activés). Remarque : pour configurer plusieurs valeurs, créez un objet de champ de widget de tableau de bord pour chaque valeur avec un numéro incrémenté dans le nom de propriété.
Balises de prob-lème			
Type d'évaluation	0	evaltype	0 - *(par défaut)* Et/Ou ; 2 - Ou.
Nom de balise	1	tags.0.tag	Toute valeur de chaîne. Remarque : le numéro dans le nom de propriété fait référence à l'ordre des balises dans la liste d'évaluation des balises. Comportement du paramètre : - *obligatoire* lors de la configuration de *Balises de problème*

Paramètre	type	name	value
Opérateur	0	tags.0.operator	0 - Contient ; 1 - Égal à ; 2 - Ne contient pas ; 3 - N'est pas égal à ; 4 - Existe ; 5 - N'existe pas. Remarque : le numéro dans le nom de propriété fait référence à l'ordre des balises dans la liste d'évaluation des balises. Comportement du paramètre : - *obligatoire* lors de la configuration de *Balises de problème*
Valeur de balise	1	tags.0.value	Toute valeur de chaîne. Remarque : le numéro dans le nom de propriété fait référence à l'ordre des balises dans la liste d'évaluation des balises. Comportement du paramètre : - *obligatoire* lors de la configuration de *Balises de problème*
Afficher les problèmes supprimés	0	show_suppressed	0 - *(par défaut)* Désactivé ; 1 - Activé.
Masquer les groupes sans problèmes	0	hide_empty_groups	0 - *(par défaut)* Désactivé ; 1 - Activé. Ce paramètre n'est pas pris en charge lors de la configuration du widget sur un tableau de bord de modèle.
Affichage des problèmes	0	ext_ack	0 - *(par défaut)* Tous ; 1 - Uniquement non acquittés ; 2 - Séparés.
Référence	1	reference	Toute valeur de chaîne composée de 5 caractères (par exemple, `ABCDE` ou `JBPNL`). Cette valeur doit être unique dans le tableau de bord auquel appartient le widget. Comportement du paramètre : - *obligatoire*

Exemples

Les exemples suivants visent à décrire uniquement la configuration des objets de champ du widget de tableau de bord pour le widget *Hôtes à problèmes*. Pour plus d'informations sur la configuration d'un tableau de bord, consultez `dashboard.create`.

Configuration d'un widget *Hôtes avec problèmes*

Configurez un widget *Hôtes avec problèmes* qui affiche les hôtes des groupes d'hôtes « 2 » et « 4 » ayant des problèmes dont le nom contient la chaîne « CPU » et présentant les niveaux de gravité suivants : « Avertissement », « Moyen », « Élevé », « Désastre ».

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": 30,
        "auto_start": 1,
        "pages": [
            {
```

```
            "widgets": [
                {
                    "type": "problemhosts",
                    "name": "Problem hosts",
                    "x": 0,
                    "y": 0,
                    "width": 36,
                    "height": 5,
                    "view_mode": 0,
                    "fields": [
                        {
                            "type": 2,
                            "name": "groupids.0",
                            "value": 2
                        },
                        {
                            "type": 2,
                            "name": "groupids.1",
                            "value": 4
                        },
                        {
                            "type": 1,
                            "name": "problem",
                            "value": "cpu"
                        },
                        {
                            "type": 0,
                            "name": "severities.0",
                            "value": 2
                        },
                        {
                            "type": 0,
                            "name": "severities.1",
                            "value": 3
                        },
                        {
                            "type": 0,
                            "name": "severities.2",
                            "value": 4
                        },
                        {
                            "type": 0,
                            "name": "severities.3",
                            "value": 5
                        }
                    ]
                }
            ]
        }
    ],
    "userGroups": [
        {
            "usrgrpid": 7,
            "permission": 2
        }
    ],
    "users": [
        {
            "userid": 1,
            "permission": 3
        }
    ]
```

```
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Voir également

- Champ du widget du tableau de bord
- `dashboard.create`
- `dashboard.update`

23 Problèmes

Description

Ces paramètres et les valeurs de propriété possibles pour les objets de champ respectifs du widget de tableau de bord vous permettent de configurer le widget *Problèmes* dans les méthodes `dashboard.create` et `dashboard.update`.

Attention:
Les propriétés `fields` du widget ne sont pas validées lors de la création ou de la mise à jour d'un tableau de bord. Cela permet aux utilisateurs de modifier les widgets intégrés et de créer des widgets personnalisés, mais introduit également le risque de créer ou de mettre à jour des widgets de manière incorrecte. Pour garantir la création ou la mise à jour réussie du widget *Problèmes*, veuillez vous reporter au comportement des paramètres décrit dans les tableaux ci-dessous.

Paramètres

Les paramètres suivants sont pris en charge pour le widget *Problèmes*.

Paramètre	type	name	value
Intervalle d'actualisation	0	rf_rate	0 - Aucune actualisation ; 10 - 10 secondes ; 30 - 30 secondes ; 60 - *(par défaut)* 1 minute ; 120 - 2 minutes ; 600 - 10 minutes ; 900 - 15 minutes.
Afficher	0	show	1 - *(par défaut)* Problèmes récents ; 2 - Historique ; 3 - Problèmes.
Groupes d'hôtes	2	groupids.0	ID du groupe d'hôtes. Remarque : pour configurer plusieurs groupes d'hôtes, créez un objet de champ de widget de tableau de bord pour chaque groupe d'hôtes, avec un numéro incrémenté dans le nom de la propriété. Ce paramètre n'est pas pris en charge lors de la configuration du widget sur un tableau de bord de modèle.

Paramètre	type	name	value
Groupes d'hôtes (widget)	1	groupids._reference	À la place de l'ID du groupe d'hôtes : `ABCDE._hostgroupids` - définissez un widget compatible (avec son paramètre *Référence* défini sur "ABCDE") comme source de données pour les groupes d'hôtes. Ce paramètre n'est pas pris en charge lors de la configuration du widget sur un tableau de bord de modèle.
Exclure les groupes d'hôtes	2	exclude_groupids.0	ID du groupe d'hôtes. Remarque : pour exclure plusieurs groupes d'hôtes, créez un objet de champ de widget de tableau de bord pour chaque groupe d'hôtes, avec un numéro incrémenté dans le nom de la propriété. Ce paramètre n'est pas pris en charge lors de la configuration du widget sur un tableau de bord de modèle.
Hôtes	3	hostids.0	ID de l'hôte. Remarque : pour configurer plusieurs hôtes, créez un objet de champ de widget de tableau de bord pour chaque hôte, avec un numéro incrémenté dans le nom de la propriété. Pour plusieurs hôtes, le paramètre *Groupes d'hôtes* doit être soit entièrement non configuré, soit configuré avec au moins un groupe d'hôtes auquel appartiennent les hôtes configurés. Ce paramètre n'est pas pris en charge lors de la configuration du widget sur un tableau de bord de modèle.
Hôtes (widget/tableau de bord)	1	hostids._reference	À la place de l'ID de l'hôte : `DASHBOARD.hostids` - définissez le sélecteur d'hôtes du tableau de bord comme source de données pour les hôtes ; `ABCDE._hostids` - définissez un widget compatible (avec son paramètre *Référence* défini sur "ABCDE") comme source de données pour les hôtes. Ce paramètre n'est pas pris en charge lors de la configuration du widget sur un tableau de bord de modèle.
Problème	1	problem	Nom de l'événement du problème (insensible à la casse, nom complet ou partie du nom).
Gravité	0	severities.0	0 - Non classifié ; 1 - Information ; 2 - Avertissement ; 3 - Moyenne ; 4 - Élevée ; 5 - Désastre. Par défaut : vide (toutes activées). Remarque : pour configurer plusieurs valeurs, créez un objet de champ de widget de tableau de bord pour chaque valeur, avec un numéro incrémenté dans le nom de la propriété.
Étiquettes des prob-lèmes			
Type d'évaluation	0	evaltype	0 - *(par défaut)* Et/Ou ; 2 - Ou.
Nom de l'étiquette	1	tags.0.tag	Toute valeur de chaîne. Remarque : le numéro dans le nom de la propriété fait référence à l'ordre de l'étiquette dans la liste d'évaluation des étiquettes. Comportement du paramètre : - *obligatoire* lors de la configuration des *Étiquettes des problèmes*

Paramètre	type	name	value
Opérateur	0	tags.0.operator	0 - Contient ; 1 - Est égal à ; 2 - Ne contient pas ; 3 - N'est pas égal à ; 4 - Existe ; 5 - N'existe pas. Remarque : le numéro dans le nom de la propriété fait référence à l'ordre de l'étiquette dans la liste d'évaluation des étiquettes. Comportement du paramètre : - *obligatoire* lors de la configuration des *Étiquettes des problèmes*
Valeur de l'étiquette	1	tags.0.value	Toute valeur de chaîne. Remarque : le numéro dans le nom de la propriété fait référence à l'ordre de l'étiquette dans la liste d'évaluation des étiquettes. Comportement du paramètre : - *obligatoire* lors de la configuration des *Étiquettes des problèmes*
Afficher les étiquettes	0	show_tags	0 - *(par défaut)* Aucune ; 1 - 1 ; 2 - 2 ; 3 - 3.
Nom de l'étiquette (format)	0	tag_name_format	0 - *(par défaut)* Complet ; 1 - Raccourci ; 2 - Aucun. Comportement du paramètre : - *pris en charge* si *Afficher les étiquettes* est défini sur "1", "2" ou "3"
Priorité d'affichage des étiquettes	1	tag_priority	Liste d'étiquettes séparées par des virgules. Comportement du paramètre : - *pris en charge* si *Afficher les étiquettes* est défini sur "1", "2" ou "3"
Afficher les données opérationnelles	0	show_opdata	0 - *(par défaut)* Aucune ; 1 - Séparément ; 2 - Avec le nom du problème.
Afficher les problèmes supprimés	0	show_suppressed	0 - *(par défaut)* Désactivé ; 1 - Activé.
État de l'accusé de réception	0	acknowledgement_status	0 - *(par défaut)* Tous ; 1 - Non confirmés ; 2 - Confirmés.
Par moi	0	acknowledged_by_me	0 - *(par défaut)* Désactivé ; 1 - Activé.

Paramètre	type	name	value
Trier les entrées par	0	sort_triggers	1 - Gravité (décroissante) ; 2 - Hôte (croissant) ; 3 - Heure (croissante) ; 4 - *(par défaut)* Heure (décroissante) ; 13 - Gravité (croissante) ; 14 - Hôte (décroissant) ; 15 - Problème (croissant) ; 16 - Problème (décroissant). Pour toutes les valeurs, à l'exception de « Heure (décroissante) » et « Heure (croissante) », le paramètre *Afficher la chronologie* doit être défini sur « Désactivé ». Les valeurs « Hôte (croissant) » et « Hôte (décroissant) » ne sont pas prises en charge lors de la configuration du widget sur un tableau de bord de modèle.
Afficher la chronologie	0	show_timeline	0 - Désactivé ; 1 - *(par défaut)* Activé. Comportement du paramètre : - *pris en charge* si *Trier les entrées par* est défini sur « Heure (décroissante) » ou « Heure (croissante) »
Surligner toute la ligne	0	highlight_row	0 - *(par défaut)* Désactivé ; 1 - Activé. Ce paramètre n'est pas pris en charge dans les thèmes à contraste élevé.
Afficher les lignes	0	show_lines	Les valeurs possibles vont de 1 à 100. Par défaut : 25.
Référence	1	reference	Toute valeur de chaîne composée de 5 caractères (par exemple, `ABCDE` ou `JBPNL`). Cette valeur doit être unique au sein du tableau de bord auquel appartient le widget. Comportement du paramètre : - *obligatoire*

Exemples

Les exemples suivants visent à décrire uniquement la configuration des objets de champ du widget de tableau de bord pour le widget *Problèmes*. Pour plus d'informations sur la configuration d'un tableau de bord, consultez `dashboard.create`.

Configuration d'un widget *Problems*

Configurez un widget *Problems* qui affiche les problèmes pour le groupe d'hôtes « 4 » et qui satisfont aux conditions suivantes :

- Les problèmes qui ont un tag nommé « scope » contenant les valeurs « performance », « availability » ou « capacity ».
- Les problèmes ayant les sévérités suivantes : « Warning », « Average », « High », « Disaster ».

En outre, configurez le widget pour afficher les tags et les données opérationnelles.

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": 30,
        "auto_start": 1,
        "pages": [
            {
                "widgets": [
                    {
```

```
                "type": "problems",
                "name": "Problems",
                "x": 0,
                "y": 0,
                "width": 36,
                "height": 5,
                "view_mode": 0,
                "fields": [
                    {
                        "type": 2,
                        "name": "groupids.0",
                        "value": 4
                    },
                    {
                        "type": 1,
                        "name": "tags.0.tag",
                        "value": "scope"
                    },
                    {
                        "type": 0,
                        "name": "tags.0.operator",
                        "value": 0
                    },
                    {
                        "type": 1,
                        "name": "tags.0.value",
                        "value": "performance"
                    },
                    {
                        "type": 1,
                        "name": "tags.1.tag",
                        "value": "scope"
                    },
                    {
                        "type": 0,
                        "name": "tags.1.operator",
                        "value": 0
                    },
                    {
                        "type": 1,
                        "name": "tags.1.value",
                        "value": "availability"
                    },
                    {
                        "type": 1,
                        "name": "tags.2.tag",
                        "value": "scope"
                    },
                    {
                        "type": 0,
                        "name": "tags.2.operator",
                        "value": 0
                    },
                    {
                        "type": 1,
                        "name": "tags.2.value",
                        "value": "capacity"
                    },
                    {
                        "type": 0,
                        "name": "severities.0",
                        "value": 2
```

```
                            },
                            {
                                "type": 0,
                                "name": "severities.1",
                                "value": 3
                            },
                            {
                                "type": 0,
                                "name": "severities.2",
                                "value": 4
                            },
                            {
                                "type": 0,
                                "name": "severities.3",
                                "value": 5
                            },
                            {
                                "type": 0,
                                "name": "show_tags",
                                "value": 1
                            },
                            {
                                "type": 0,
                                "name": "show_opdata",
                                "value": 1
                            }
                        ]
                    }
                ]
            }
        ],
        "userGroups": [
            {
                "usrgrpid": 7,
                "permission": 2
            }
        ],
        "users": [
            {
                "userid": 1,
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Voir également

- Champ du widget du tableau de bord
- `dashboard.create`
- `dashboard.update`

Description

Ces paramètres et les valeurs de propriété possibles pour les objets de champ respectifs du widget de tableau de bord vous permettent de configurer le widget *Problèmes par gravité* dans les méthodes `dashboard.create` et `dashboard.update`.

Attention:
Les propriétés `fields` du widget ne sont pas validées lors de la création ou de la mise à jour d'un tableau de bord. Cela permet aux utilisateurs de modifier les widgets intégrés et de créer des widgets personnalisés, mais introduit également le risque de créer ou de mettre à jour des widgets de manière incorrecte. Pour garantir la création ou la mise à jour réussie du widget *Problèmes par gravité*, veuillez vous reporter au comportement des paramètres décrit dans les tableaux ci-dessous.

Paramètres

Les paramètres suivants sont pris en charge pour le widget *Problems by severity*.

Paramètre	type	name	value
Intervalle d'actualisation	0	rf_rate	0 - Pas d'actualisation ; 10 - 10 secondes ; 30 - 30 secondes ; 60 - *(par défaut)* 1 minute ; 120 - 2 minutes ; 600 - 10 minutes ; 900 - 15 minutes.
Groupes d'hôtes	2	groupids.0	ID du groupe d'hôtes. Remarque : pour configurer plusieurs groupes d'hôtes, créez un objet de champ de widget de tableau de bord pour chaque groupe d'hôtes avec un numéro incrémenté dans le nom de propriété. Ce paramètre n'est pas pris en charge lors de la configuration du widget sur un tableau de bord de modèle.
Groupes d'hôtes (Widget)	1	groupids._reference	Au lieu de l'ID du groupe d'hôtes : `ABCDE._hostgroupids` - définissez un widget compatible (avec son paramètre *Reference* défini sur "ABCDE") comme source de données pour les groupes d'hôtes. Ce paramètre n'est pas pris en charge lors de la configuration du widget sur un tableau de bord de modèle.
Exclure les groupes d'hôtes	2	exclude_groupids.0	ID du groupe d'hôtes. Remarque : pour exclure plusieurs groupes d'hôtes, créez un objet de champ de widget de tableau de bord pour chaque groupe d'hôtes avec un numéro incrémenté dans le nom de propriété. Ce paramètre n'est pas pris en charge lors de la configuration du widget sur un tableau de bord de modèle.
Hôtes	3	hostids.0	ID de l'hôte. Remarque : pour configurer plusieurs hôtes, créez un objet de champ de widget de tableau de bord pour chaque hôte avec un numéro incrémenté dans le nom de propriété. Pour plusieurs hôtes, le paramètre *Groupes d'hôtes* ne doit soit pas être configuré du tout, soit être configuré avec au moins un groupe d'hôtes auquel appartiennent les hôtes configurés. Ce paramètre n'est pas pris en charge lors de la configuration du widget sur un tableau de bord de modèle.

Paramètre	type	name	value
Hôtes (Widget/Tableau de bord)	1	hostids._reference	Au lieu de l'ID de l'hôte : `DASHBOARD.hostids` - définissez le sélecteur d'hôte du tableau de bord comme source de données pour les hôtes ; `ABCDE._hostids` - définissez un widget compatible (avec son paramètre *Reference* défini sur "ABCDE") comme source de données pour les hôtes. Ce paramètre n'est pas pris en charge lors de la configuration du widget sur un tableau de bord de modèle.
Problème	1	problem	Nom d'événement du problème (insensible à la casse, nom complet ou partie de celui-ci).
Gravité	0	severities.0	0 - Non classé ; 1 - Information ; 2 - Avertissement ; 3 - Moyenne ; 4 - Haute ; 5 - Désastre. Par défaut : vide (tous activés). Remarque : pour configurer plusieurs valeurs, créez un objet de champ de widget de tableau de bord pour chaque valeur avec un numéro incrémenté dans le nom de propriété.
Balises de problème			
Type d'évaluation	0	evaltype	0 - *(par défaut)* Et/Ou ; 2 - Ou.
Nom de balise	1	tags.0.tag	Toute valeur de chaîne. Remarque : le numéro dans le nom de propriété fait référence à l'ordre des balises dans la liste d'évaluation des balises. Comportement du paramètre : - *obligatoire* lors de la configuration de *Balises de problème*
Opérateur	0	tags.0.operator	0 - Contient ; 1 - Égal à ; 2 - Ne contient pas ; 3 - N'est pas égal à ; 4 - Existe ; 5 - N'existe pas. Remarque : le numéro dans le nom de propriété fait référence à l'ordre des balises dans la liste d'évaluation des balises. Comportement du paramètre : - *obligatoire* lors de la configuration de *Balises de problème*
Valeur de balise	1	tags.0.value	Toute valeur de chaîne. Remarque : le numéro dans le nom de propriété fait référence à l'ordre des balises dans la liste d'évaluation des balises. Comportement du paramètre : - *obligatoire* lors de la configuration de *Balises de problème*
Afficher	0	show_type	0 - *(par défaut)* Groupes d'hôtes ; 1 - Totaux. Ce paramètre n'est pas pris en charge lors de la configuration du widget sur un tableau de bord de modèle, et est défini par défaut sur "Totaux".

Paramètre	type	name	value
Disposition	0	layout	0 - *(par défaut)* Horizontale ; 1 - Verticale. Comportement du paramètre : - *pris en charge* si *Afficher* est défini sur "Totaux"
Afficher les données opérationnelles	0	show_opdata	0 - *(par défaut)* Aucune ; 1 - Séparément ; 2 - Avec le nom du problème.
Afficher les problèmes supprimés	0	show_suppressed	0 - *(par défaut)* Désactivé ; 1 - Activé.
Masquer les groupes sans problèmes	0	hide_empty_groups	0 - *(par défaut)* Désactivé ; 1 - Activé. Comportement du paramètre : - *pris en charge* si *Afficher* est défini sur "Groupes d'hôtes" Ce paramètre n'est pas pris en charge lors de la configuration du widget sur un tableau de bord de modèle.
Affichage des problèmes	0	ext_ack	0 - *(par défaut)* Tous ; 1 - Non acquittés uniquement ; 2 - Séparés.
Afficher la chronologie	0	show_timeline	0 - Désactivé ; 1 - *(par défaut)* Activé.
Reference	1	reference	Toute valeur de chaîne composée de 5 caractères (par exemple, `ABCDE` ou `JBPNL`). Cette valeur doit être unique dans le tableau de bord auquel appartient le widget. Comportement du paramètre : - *obligatoire*

Exemples

Les exemples suivants visent à décrire uniquement la configuration des objets de champ du widget de tableau de bord pour le widget *Problèmes par gravité*. Pour plus d'informations sur la configuration d'un tableau de bord, consultez `dashboard.create`.

Configuration d'un widget *Problems by severity*

Configurez un widget *Problems by severity* qui affiche le total des problèmes pour tous les groupes d'hôtes.

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": 30,
        "auto_start": 1,
        "pages": [
            {
                "widgets": [
                    {
                        "type": "problemsbysv",
```

```
                        "name": "Problems by severity",
                        "x": 0,
                        "y": 0,
                        "width": 36,
                        "height": 5,
                        "view_mode": 0,
                        "fields": [
                            {
                                "type": 0,
                                "name": "show_type",
                                "value": 1
                            }
                        ]
                    }
                ]
            }
        ],
        "userGroups": [
            {
                "usrgrpid": 7,
                "permission": 2
            }
        ],
        "users": [
            {
                "userid": 1,
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Voir également

- Champ du widget du tableau de bord
- `dashboard.create`
- `dashboard.update`

25 Nuage de points

Description

Ces paramètres et les valeurs de propriété possibles pour les objets de champ respectifs du widget de tableau de bord vous permettent de configurer le widget *Graphique de dispersion* dans les méthodes `dashboard.create` et `dashboard.update`.

Attention:
Les propriétés `fields` du widget ne sont pas validées lors de la création ou de la mise à jour d'un tableau de bord. Cela permet aux utilisateurs de modifier les widgets intégrés et de créer des widgets personnalisés, mais introduit également le risque de créer ou de mettre à jour des widgets de manière incorrecte. Pour garantir la création ou la mise à jour réussie du widget *Graphique de dispersion*, veuillez vous reporter au comportement des paramètres décrit dans les tableaux ci-dessous.

Paramètres

Les paramètres suivants sont pris en charge pour le widget *Nuage de points*.

Paramètre	type	name	value
Intervalle d'actualisation	0	rf_rate	0 - Aucune actualisation; 10 - 10 secondes; 30 - 30 secondes; 60 - *(par défaut)* 1 minute; 120 - 2 minutes; 600 - 10 minutes; 900 - 15 minutes.
Référence	1	reference	Toute valeur de chaîne composée de 5 caractères (par exemple, `ABCDE` ou `JBPNL`). Cette valeur doit être unique dans le tableau de bord auquel appartient le widget. Comportement du paramètre: - *obligatoire*

Jeu de données

Les paramètres suivants sont pris en charge pour configurer un *jeu de données*.

Note:
Le premier nombre dans le nom de propriété (par exemple, ds.0.hosts.0, ds.0.items.0) représente le jeu de données particulier, tandis que le second nombre, s'il est présent, représente l'hôte ou l'élément configuré.

Paramètre	type	name	value
Type de jeu de données ***Jeu de données : Liste d'éléments***	0	ds.0.dataset_type	0 - Liste d'éléments ; 1 - *(par défaut)* Modèles d'éléments.
Couleur	1	ds.0.color.0	Code couleur hexadécimal (par exemple, `FF0000`). Comportement du paramètre : - *obligatoire* si *Type de jeu de données* est défini sur "Liste d'éléments"

Paramètre	type	name	value
Axe X	4	ds.0.x_axis_itemids.0	ID d'élément. Lors de la configuration du widget sur un tableau de bord de modèle, seuls les éléments configurés sur le modèle doivent être définis. Remarque : Pour configurer plusieurs éléments, créez un objet de champ de widget de tableau de bord pour chaque élément avec un numéro incrémenté dans le nom de propriété. Comportement du paramètre : - *obligatoire* si *Type de jeu de données* est défini sur "Liste d'éléments" et que *Axe X (Widget)* n'est pas défini
Axe X (Widget)	1	ds.0.x_axis_itemids.0.reference	Référence d'ID d'élément : `ABCDE._itemid` - définir un widget compatible (avec son paramètre *Référence* défini sur "`ABCDE`") comme source de données pour les éléments. Remarque : Pour configurer plusieurs widgets, créez un objet de champ de widget de tableau de bord pour chaque widget avec un numéro incrémenté dans le nom de propriété. Comportement du paramètre : - *obligatoire* si *Type de jeu de données* est défini sur "Liste d'éléments" et que *Axe X* n'est pas défini
Axe Y	4	ds.0.y_axis_itemids.0	ID d'élément. Lors de la configuration du widget sur un tableau de bord de modèle, seuls les éléments configurés sur le modèle doivent être définis. Remarque : Pour configurer plusieurs éléments, créez un objet de champ de widget de tableau de bord pour chaque élément avec un numéro incrémenté dans le nom de propriété. Comportement du paramètre : - *obligatoire* si *Type de jeu de données* est défini sur "Liste d'éléments" et que *Axe Y (Widget)* n'est pas défini
Axe Y (Widget)	1	ds.0.y_axis_itemids.0.reference	Référence d'ID d'élément : `ABCDE._itemid` - définir un widget compatible (avec son paramètre *Référence* défini sur "`ABCDE`") comme source de données pour les éléments. Remarque : Pour configurer plusieurs widgets, créez un objet de champ de widget de tableau de bord pour chaque widget avec un numéro incrémenté dans le nom de propriété. Comportement du paramètre : - *obligatoire* si *Type de jeu de données* est défini sur "Liste d'éléments" et que *Axe Y* n'est pas défini
Remplacer l'hôte	1	ds.0.override_hostid.reference	`ABCDE._hostid`- définir un widget compatible (avec son paramètre "Référence" défini sur "`ABCDE`") comme source de données pour les hôtes ; `DASHBOARD._hostid` - définir le sélecteur d'hôte du tableau de bord comme source de données pour les hôtes. Ce paramètre n'est pas pris en charge lors de la configuration du widget sur un tableau de bord de modèle.

Paramètre	type	name	value
Jeu de données : Modèles d'éléments			
Couleur	1	ds.0.color	Code couleur hexadécimal (par exemple, FF0000). Comportement du paramètre : - *pris en charge* si *Type de jeu de données* est défini sur "Modèles d'éléments" et que *Palette de couleurs* n'est pas définie
Palette de couleurs	0	ds.0.color_palette	Indice de palette de couleurs. Plage de valeurs possibles : 0-11. Par défaut : 0. Comportement du paramètre : - *pris en charge* si *Type de jeu de données* est défini sur "Modèles d'éléments" et que *Couleur* n'est pas définie
Modèles d'hôte	1	ds.0.hosts.0	Nom ou modèle d'hôte (par exemple, "Zabbix*"). Comportement du paramètre : - *obligatoire* si *Type de jeu de données* est défini sur "Modèles d'éléments" Ce paramètre n'est pas pris en charge lors de la configuration du widget sur un tableau de bord de modèle.
Modèles d'éléments de l'axe X	1	ds.0.x_axis_items.0	Nom ou modèle d'élément (par exemple, "*: Number of processed *values per second"). Lors de la configuration du widget sur un tableau de bord de modèle, seuls les modèles pour les éléments configurés sur le modèle doivent être définis. Comportement du paramètre : - *obligatoire* si *Type de jeu de données* est défini sur "Modèles d'éléments"
Modèles d'éléments de l'axe Y	1	ds.0.y_axis_items.0	Nom ou modèle d'élément (par exemple, "*: Number of processed *values per second"). Lors de la configuration du widget sur un tableau de bord de modèle, seuls les modèles pour les éléments configurés sur le modèle doivent être définis. Comportement du paramètre : - *obligatoire* si *Type de jeu de données* est défini sur "Modèles d'éléments"
Groupes d'hôtes	2	groupids.0	ID de groupe d'hôtes. Remarque : Pour configurer plusieurs groupes d'hôtes, créez un objet de champ de widget de tableau de bord pour chaque groupe d'hôtes avec un numéro incrémenté dans le nom de propriété. Ce paramètre n'est pas pris en charge lors de la configuration du widget sur un tableau de bord de modèle.

Paramètre	type	name	value
Groupes d'hôtes (Widget)	1	groupids._reference	Au lieu de l'ID de groupe d'hôtes : `ABCDE._hostgroupids` - définir un widget compatible (avec son paramètre *Référence* défini sur "`ABCDE`") comme source de données pour les groupes d'hôtes. Ce paramètre n'est pas pris en charge lors de la configuration du widget sur un tableau de bord de modèle.
Remplacer l'hôte	1	ds.0.override_hostid._reference	Au lieu de l'ID d'hôte : `ABCDE._hostid`- définir un widget compatible (avec son paramètre "Référence" défini sur "`ABCDE`") comme source de données pour les hôtes ; `DASHBOARD._hostid` - définir le sélecteur d'hôte du tableau de bord comme source de données pour les hôtes. Ce paramètre n'est pas pris en charge lors de la configuration du widget sur un tableau de bord de modèle.
Balises d'hôte			
Type d'évaluation	0	evaltype_host	0 - *(par défaut)* Et/Ou ; 2 - Ou. Ce paramètre n'est pas pris en charge lors de la configuration du widget sur un tableau de bord de modèle.
Nom de balise	1	host_tags.0.tag	Toute valeur de chaîne. Remarque : Le nombre dans le nom de propriété fait référence à l'ordre des balises dans la liste d'évaluation des balises. Comportement du paramètre : - *obligatoire* lors de la configuration des *Balises d'hôte* Ce paramètre n'est pas pris en charge lors de la configuration du widget sur un tableau de bord de modèle.
Opérateur	0	host_tags.0.operator	0 - Contient ; 1 - Égal à ; 2 - Ne contient pas ; 3 - Différent de ; 4 - Existe ; 5 - N'existe pas. Remarque : Le nombre dans le nom de propriété fait référence à l'ordre des balises dans la liste d'évaluation des balises. Comportement du paramètre : - *obligatoire* lors de la configuration des *Balises d'hôte* Ce paramètre n'est pas pris en charge lors de la configuration du widget sur un tableau de bord de modèle.
Valeur de balise	1	host_tags.0.value	Toute valeur de chaîne. Remarque : Le nombre dans le nom de propriété fait référence à l'ordre des balises dans la liste d'évaluation des balises. Comportement du paramètre : - *obligatoire* lors de la configuration des *Balises d'hôte* Ce paramètre n'est pas pris en charge lors de la configuration du widget sur un tableau de bord de modèle.

Paramètre	type	name	value
Marqueur	0	ds.0.marker	0 - *(par défaut)* Ellipse ; 1 - Carré ; 2 - Triangle ; 3 - Losange ; 4 - Étoile ; 5 - Croix.
Taille du marqueur	0	ds.0.marker_size	0 - *(par défaut)* Petit (6px) ; 1 - Moyen (9px) ; 2 - Grand (15px).
Décalage temporel	1	ds.0.timeshift	Chaîne temporelle valide (par exemple, 3600, 1h, etc.). Vous pouvez utiliser des suffixes temporels. Les valeurs négatives sont également autorisées. Par défaut : "" (vide).
Intervalle d'agrégation	1	ds.0.aggregate_interval	Chaîne temporelle valide (par exemple, 3600, 1h, etc.). Vous pouvez utiliser des suffixes temporels. Par défaut : 15m.
Fonction d'agrégation	0	ds.0.aggregate_function	0 - non utilisé ; 1 - min ; 2 - max ; 3 - *(par défaut)* avg ; 4 - count ; 5 - sum ; 6 - first ; 7 - last.

Options d'affichage

Les paramètres suivants sont pris en charge pour configurer les *options d'affichage*.

Paramètre	type	name	value
Sélection des données d'historique	0	source	0 - *(par défaut)* Auto; 1 - Historique; 2 - Tendances.
Noms d'hôte dans les étiquettes	0	show_hostnames	0 - *(par défaut)* Auto; 1 - Afficher; 2 - Masquer. Ce paramètre n'est pas pris en charge lors de la configuration du widget sur un tableau de bord de modèle.

Période de temps

Les paramètres suivants sont pris en charge pour configurer la *Période de temps*.

Paramètre	type	name	value
Période de temps	1	time_period._reference	DASHBOARD._timeperiod - définit le sélecteur de période de temps du tableau de bord comme source de données ; ABCDE._timeperiod - définit un widget compatible (avec son paramètre *Reference* défini sur "ABCDE") comme source de données. Par défaut : DASHBOARD._timeperiod Vous pouvez également définir la période de temps uniquement dans les paramètres *From* et *To*.

Paramètre	type	name	value
From	1	time_period.from	Chaîne de temps valide en syntaxe temporelle absolue (`YYYY-MM-DD hh:mm:ss`) ou relative (`now`, `now/d`, `now/w-1w`, etc.). Comportement du paramètre : - *pris en charge* si *Période de temps* n'est pas défini - *obligatoire* si `time_period.to` est défini
To	1	time_period.to	Chaîne de temps valide en syntaxe temporelle absolue (`YYYY-MM-DD hh:mm:ss`) ou relative (`now`, `now/d`, `now/w-1w`, etc.). Comportement du paramètre : - *pris en charge* si *Période de temps* n'est pas défini - *obligatoire* si `time_period.from` est défini

Axes

Les paramètres suivants sont pris en charge pour configurer les *axes*.

Paramètre	type	name	value
Axe X	0	x_axis	0 - Désactivé; 1 - *(par défaut)* Activé.
Axe Y	0	y_axis	0 - *(par défaut)* Désactivé; 1 - Activé.
Min	1	x_axis_min	Toute valeur numérique. Par défaut : "" (vide).
		y_axis_min	
Max	1	x_axis_max	Toute valeur numérique. Par défaut : "" (vide).
		y_axis_max	
Unités (type)	0	x_axis_units	0 - *(par défaut)* Auto; 1 - Statique.
		y_axis_units	
Unités (valeur)	1	x_axis_static_units	Toute valeur de chaîne. Par défaut : "" (vide).
		y_axis_static_units	

Légende

Les paramètres suivants sont pris en charge pour configurer la *Légende*.

Paramètre	type	name	value
Afficher la légende	0	legend	0 - Désactivé ; 1 - *(par défaut)* Activé.
Afficher la fonction d'agrégation	0	legend_aggregation	0 - *(par défaut)* Désactivé ; 1 - Activé. Comportement du paramètre : - *pris en charge* si *Afficher la légende* est défini sur « Activé »
Lignes	0	legend_lines_mode	0 - *(par défaut)* Fixe ; 1 - Variable. Comportement du paramètre : - *pris en charge* si *Afficher la légende* est défini sur « Activé »

Paramètre	type	name	value
Nombre de lignes/ Nombre maximal de lignes	0	legend_lines	Plage de valeurs possible : 1-10. Par défaut : 1. Comportement du paramètre : - *pris en charge* si *Afficher la légende* est défini sur « Activé »
Nombre de colonnes	0	legend_columns	Plage de valeurs possible : 1-4. Par défaut : 4. Comportement du paramètre : - *pris en charge* si *Afficher la légende* est défini sur « Activé »

Seuils

Les paramètres suivants sont pris en charge pour configurer les *seuils*.

Note:
Le nombre dans le nom de propriété *Thresholds* (par exemple, thresholds.0.color) fait référence à la position du seuil dans une liste, triée par ordre croissant. Cependant, si les seuils sont configurés dans un ordre différent, les valeurs seront triées par ordre croissant après la mise à jour de la configuration du widget dans le frontend Zabbix (par exemple, `"thresholds.0.threshold":"5"` → `"thresholds.0.threshold":"1"` ; `"thresholds.1.threshold":"1"` → `"thresholds.1.threshold": "5"`).

Paramètre	type	name	value
Interpolation des couleurs	0	interpolation	0 - *(par défaut)* Désactivée ; 1 - Activée.
Couleur	1	thresholds.0.color	Code couleur hexadécimal (par exemple, FF0000).
Axe X	1	thresholds.0.x_axis_threshold	Seuil valeur numérique. Les suffixes (par exemple, ”1d”, ”2w”, ”4K”, ”8G”) sont pris en charge.
Axe Y	1	thresholds.0.y_axis_threshold	Seuil valeur numérique. Les suffixes (par exemple, ”1d”, ”2w”, ”4K”, ”8G”) sont pris en charge.

Exemples

Les exemples suivants visent uniquement à décrire la configuration des objets de champ de widget de tableau de bord pour le widget *Scatter plot*. Pour plus d'informations sur la configuration d'un tableau de bord, consultez `dashboard.create`.

Configuration d'un widget *Nuage de points*

Configurez un widget *Nuage de points* de la manière suivante :

- 5 ensembles de données de type « Item patterns », chacun avec l'agrégation par défaut (avg, 15m) et des marqueurs différents.
- La légende affiche les éléments configurés sur une seule ligne et en deux colonnes.
- Des seuils sont configurés pour colorer un marqueur en rouge lorsque sa valeur atteint 80 sur l'un des axes.

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": 30,
        "auto_start": 1,
        "pages": [
            {
                "widgets": [
                    {
                        "type": "scatterplot",
                        "name": "Scatter plot",
```

```
                "x": 0,
                "y": 0,
                "width": 36,
                "height": 5,
                "view_mode": 0,
                "fields": [
                    {
                        "type": 1,
                        "name": "ds.0.color",
                        "value": "0C5A87"
                    },
                    {
                        "type": 1,
                        "name": "ds.0.hosts.0",
                        "value": "de-frankfurt*"
                    },
                    {
                        "type": 1,
                        "name": "ds.0.x_axis_items.0",
                        "value": "Memory utilization"
                    },
                    {
                        "type": 1,
                        "name": "ds.0.y_axis_items.0",
                        "value": "CPU utilization"
                    },
                    {
                        "type": 0,
                        "name": "ds.0.marker",
                        "value": 0
                    },
                    {
                        "type": 0,
                        "name": "ds.0.marker_size",
                        "value": 1
                    },
                    {
                        "type": 1,
                        "name": "ds.1.color",
                        "value": "66B0D9"
                    },
                    {
                        "type": 1,
                        "name": "ds.1.hosts.0",
                        "value": "fr-paris*"
                    },
                    {
                        "type": 1,
                        "name": "ds.1.x_axis_items.0",
                        "value": "Memory utilization"
                    },
                    {
                        "type": 1,
                        "name": "ds.1.y_axis_items.0",
                        "value": "CPU utilization"
                    },
                    {
                        "type": 0,
                        "name": "ds.1.marker",
                        "value": 1
                    },
                    {
```

```
                "type": 0,
                "name": "ds.1.marker_size",
                "value": 1
            },
            {
                "type": 1,
                "name": "ds.2.color",
                "value": "0A466A"
            },
            {
                "type": 1,
                "name": "ds.2.hosts.0",
                "value": "lv-riga*"
            },
            {
                "type": 1,
                "name": "ds.2.x_axis_items.0",
                "value": "Memory utilization"
            },
            {
                "type": 1,
                "name": "ds.2.y_axis_items.0",
                "value": "CPU utilization"
            },
            {
                "type": 0,
                "name": "ds.2.marker",
                "value": 2
            },
            {
                "type": 0,
                "name": "ds.2.marker_size",
                "value": 1
            },
            {
                "type": 1,
                "name": "ds.3.color",
                "value": "3394C3"
            },
            {
                "type": 1,
                "name": "ds.3.hosts.0",
                "value": "pl-warsaw*"
            },
            {
                "type": 1,
                "name": "ds.3.x_axis_items.0",
                "value": "Memory utilization"
            },
            {
                "type": 1,
                "name": "ds.3.y_axis_items.0",
                "value": "CPU utilization"
            },
            {
                "type": 0,
                "name": "ds.3.marker",
                "value": 3
            },
            {
                "type": 0,
                "name": "ds.3.marker_size",
```

```
                "value": 1
            },
            {
                "type": 1,
                "name": "ds.4.color",
                "value": "1492C8"
            },
            {
                "type": 1,
                "name": "ds.4.hosts.0",
                "value": "se-stockholm*"
            },
            {
                "type": 1,
                "name": "ds.4.x_axis_items.0",
                "value": "Memory utilization"
            },
            {
                "type": 1,
                "name": "ds.4.y_axis_items.0",
                "value": "CPU utilization"
            },
            {
                "type": 0,
                "name": "ds.4.marker",
                "value": 4
            },
            {
                "type": 0,
                "name": "ds.4.marker_size",
                "value": 1
            },
            {
                "type": 0,
                "name": "legend_columns",
                "value": 2
            },
            {
                "type": 0,
                "name": "interpolation",
                "value": 1
            },
            {
                "type": 1,
                "name": "thresholds.0.color",
                "value": "D40000"
            },
            {
                "type": 1,
                "name": "thresholds.0.x_axis_threshold",
                "value": "80"
            },
            {
                "type": 1,
                "name": "thresholds.0.y_axis_threshold",
                "value": ""
            },
            {
                "type": 1,
                "name": "thresholds.1.color",
                "value": "D40000"
            },
```

```
                            {
                                "type": 1,
                                "name": "thresholds.1.x_axis_threshold",
                                "value": ""
                            },
                            {
                                "type": 1,
                                "name": "thresholds.1.y_axis_threshold",
                                "value": "80"
                            },
                            {
                                "type": 1,
                                "name": "reference",
                                "value": "JQISY"
                            }
                        ]
                    }
                ]
            }
        ],
        "userGroups": [
            {
                "usrgrpid": 7,
                "permission": 2
            }
        ],
        "users": [
            {
                "userid": 1,
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Voir aussi

- Champ de widget de tableau de bord
- `dashboard.create`
- `dashboard.update`

26 Rapport SLA

Description

Ces paramètres et les valeurs de propriété possibles pour les objets de champ respectifs du widget de tableau de bord vous permettent de configurer le widget *Rapport SLA* dans les méthodes `dashboard.create` et `dashboard.update`.

Attention:
Les propriétés `fields` des widgets ne sont pas validées lors de la création ou de la mise à jour d'un tableau de bord. Cela permet aux utilisateurs de modifier les widgets intégrés et de créer des widgets personnalisés, mais introduit également le risque de créer ou de mettre à jour des widgets de manière incorrecte. Pour garantir la création ou la mise à jour réussie du widget *Rapport SLA*, veuillez vous reporter au comportement des paramètres décrit dans les tableaux ci-dessous.

Paramètres

Les paramètres suivants sont pris en charge pour le widget *Rapport SLA*.

Paramètre	type	name	value
Intervalle d'actualisation	0	rf_rate	0 - *(par défaut)* Pas d'actualisation ; 10 - 10 secondes ; 30 - 30 secondes ; 60 - 1 minute ; 120 - 2 minutes ; 600 - 10 minutes ; 900 - 15 minutes.
SLA	10	slaid.0	ID du SLA. Comportement du paramètre : - *obligatoire*
Service	9	serviceid.0	ID du service.
Afficher les périodes	0	show_periods	Les valeurs possibles vont de 1 à 100. Valeur par défaut : 20.
Du	1	date_from	Chaîne de date valide au format `YYYY-MM-DD`. Les dates relatives avec les modificateurs `d`, `w`, `M`, `y` (par exemple, `now`, `now/d`, `now/w-1w`, etc.) sont prises en charge.
Au	1	date_to	Chaîne de date valide au format `YYYY-MM-DD`. Les dates relatives avec les modificateurs `d`, `w`, `M`, `y` (par exemple, `now`, `now/d`, `now/w-1w`, etc.) sont prises en charge.

Exemples

Les exemples suivants visent à décrire uniquement la configuration des objets de champ du widget de tableau de bord pour le widget *Rapport SLA*. Pour plus d'informations sur la configuration d'un tableau de bord, consultez `dashboard.create`.

Configuration d'un widget *Rapport SLA*

Configurez un widget *Rapport SLA* qui affiche le rapport SLA pour le SLA « 4 » du service « 2 » pour la période des 30 derniers jours.

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": 30,
        "auto_start": 1,
        "pages": [
            {
                "widgets": [
                    {
                        "type": "slareport",
                        "name": "SLA report",
                        "x": 0,
                        "y": 0,
                        "width": 36,
                        "height": 5,
                        "view_mode": 0,
                        "fields": [
```

```
                            {
                                "type": 10,
                                "name": "slaid.0",
                                "value": 4
                            },
                            {
                                "type": 9,
                                "name": "serviceid.0",
                                "value": 2
                            },
                            {
                                "type": 1,
                                "name": "date_from",
                                "value": "now-30d"
                            },
                            {
                                "type": 1,
                                "name": "date_to",
                                "value": "now"
                            }
                        ]
                    }
                ]
            }
        ],
        "userGroups": [
            {
                "usrgrpid": 7,
                "permission": 2
            }
        ],
        "users": [
            {
                "userid": 1,
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Voir également

- Champ du widget du tableau de bord
- `dashboard.create`
- `dashboard.update`

27 Informations système

Description

Ces paramètres et les valeurs de propriété possibles pour les objets de champ respectifs du widget de tableau de bord vous permettent de configurer le widget *Informations système* dans les méthodes `dashboard.create` et `dashboard.update`.

Attention:
Les propriétés `fields` du widget ne sont pas validées lors de la création ou de la mise à jour d'un tableau de bord. Cela permet aux utilisateurs de modifier les widgets intégrés et de créer des widgets personnalisés, mais introduit également le risque de créer ou de mettre à jour des widgets de manière incorrecte. Pour garantir la création ou la mise à jour réussie du widget *Informations système*, veuillez vous reporter au comportement des paramètres décrit dans les tableaux ci-dessous.

Paramètres

Les paramètres suivants sont pris en charge pour le widget *System Information*.

Parameter	type	name	value
Intervalle d'actualisation	0	rf_rate	0 - Pas d'actualisation; 10 - 10 secondes; 30 - 30 secondes; 60 - 1 minute; 120 - 2 minutes; 600 - 10 minutes; 900 - *(par défaut)* 15 minutes.
Afficher	0	info_type	0 - *(par défaut)* Statistiques du système; 1 - Nœuds de haute disponibilité.
Afficher les détails de la vérification des mises à jour logicielles	0	show_software_update_check_details	0 - *(par défaut)* Désactivé; 1 - Activé. Comportement du paramètre: - *pris en charge* si `AllowSoftwareUpdateCheck` dans la configuration du serveur Zabbix est activé et si *Afficher* est défini sur "System stats"

Exemples

Les exemples suivants visent à décrire uniquement la configuration des objets de champ du widget de tableau de bord pour le widget *Informations système*. Pour plus d'informations sur la configuration d'un tableau de bord, consultez `dashboard.create`.

Configuration d'un widget *Informations système*

Configurez un widget *Informations système* qui affiche les statistiques du système avec un intervalle d'actualisation de 10 minutes et la vérification des mises à jour logicielles activée.

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": 30,
        "auto_start": 1,
        "pages": [
            {
                "widgets": [
                    {
                        "type": "systeminfo",
                        "name": "System information",
                        "x": 0,
                        "y": 0,
                        "width": 36,
                        "height": 5,
                        "view_mode": 0,
                        "fields": [
                            {
                                "type": 0,
                                "name": "rf_rate",
```

```
                                "value": 600
                            },
                            {
                                "type": 0,
                                "name": "show_software_update_check_details",
                                "value": 1
                            }
                        ]
                    }
                ]
            }
        ],
        "userGroups": [
            {
                "usrgrpid": 7,
                "permission": 2
            }
        ],
        "users": [
            {
                "userid": 1,
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Voir également

- Champ du widget du tableau de bord
- `dashboard.create`
- `dashboard.update`

28 Hôtes principaux

Description

Ces paramètres et les valeurs de propriété possibles pour les objets de champ respectifs du widget de tableau de bord vous permettent de configurer le widget *Top hosts* dans les méthodes `dashboard.create` et `dashboard.update`.

Attention:
Les propriétés `fields` du widget ne sont pas validées lors de la création ou de la mise à jour d'un tableau de bord. Cela permet aux utilisateurs de modifier les widgets intégrés et de créer des widgets personnalisés, mais introduit également le risque de créer ou de mettre à jour des widgets de manière incorrecte. Pour garantir la création ou la mise à jour réussie du widget *Top hosts*, veuillez vous reporter au comportement des paramètres décrit dans les tableaux ci-dessous.

Paramètres

Les paramètres suivants sont pris en charge pour le widget *Hôtes principaux*.

Paramètre	type	name	value
Intervalle d'actualisation	0	rf_rate	0 - Pas d'actualisation ; 10 - 10 secondes ; 30 - 30 secondes ; 60 - *(par défaut)* 1 minute ; 120 - 2 minutes ; 600 - 10 minutes ; 900 - 15 minutes.
Groupes d'hôtes	2	groupids.0	ID du groupe d'hôtes. Remarque : pour configurer plusieurs groupes d'hôtes, créez un objet de champ de widget de tableau de bord pour chaque groupe d'hôtes, en incrémentant le nombre dans le nom de la propriété. Ce paramètre n'est pas pris en charge lors de la configuration du widget sur un tableau de bord de modèle.
Groupes d'hôtes (widget)	1	groupids._reference	Au lieu de l'ID du groupe d'hôtes : `ABCDE._hostgroupids` - définissez un widget compatible (avec son paramètre *Reference* défini sur "ABCDE") comme source de données pour les groupes d'hôtes. Ce paramètre n'est pas pris en charge lors de la configuration du widget sur un tableau de bord de modèle.
Hôtes	3	hostids.0	ID de l'hôte. Remarque : pour configurer plusieurs hôtes, créez un objet de champ de widget de tableau de bord pour chaque hôte, en incrémentant le nombre dans le nom de la propriété. Pour plusieurs hôtes, le paramètre *Groupes d'hôtes* doit soit ne pas être configuré, soit être configuré avec au moins un groupe d'hôtes auquel appartiennent les hôtes configurés. Ce paramètre n'est pas pris en charge lors de la configuration du widget sur un tableau de bord de modèle.
Hôtes (widget/tableau de bord)	1	hostids._reference	Au lieu de l'ID de l'hôte : `DASHBOARD.hostids` - définissez le sélecteur d'hôtes du tableau de bord comme source de données pour les hôtes ; `ABCDE._hostids` - définissez un widget compatible (avec son paramètre *Reference* défini sur "ABCDE") comme source de données pour les hôtes. Ce paramètre n'est pas pris en charge lors de la configuration du widget sur un tableau de bord de modèle.
Étiquettes d'hôtes			
Type d'évaluation	0	evaltype	0 - *(par défaut)* Et/ou ; 2 - Ou. Ce paramètre n'est pas pris en charge lors de la configuration du widget sur un tableau de bord de modèle.
Nom de l'étiquette	1	tags.0.tag	Toute valeur de chaîne. Remarque : le nombre dans le nom de la propriété fait référence à l'ordre de l'étiquette dans la liste d'évaluation des étiquettes. Comportement du paramètre : - *obligatoire* lors de la configuration des *Étiquettes d'hôtes* Ce paramètre n'est pas pris en charge lors de la configuration du widget sur un tableau de bord de modèle.

Paramètre	type	name	value
Opérateur	0	tags.0.operator	0 - Contient ; 1 - Est égal à ; 2 - Ne contient pas ; 3 - N'est pas égal à ; 4 - Existe ; 5 - N'existe pas. Remarque : le nombre dans le nom de la propriété fait référence à l'ordre de l'étiquette dans la liste d'évaluation des étiquettes. Comportement du paramètre : - *obligatoire* lors de la configuration des *Étiquettes d'hôtes* Ce paramètre n'est pas pris en charge lors de la configuration du widget sur un tableau de bord de modèle.
Valeur de l'étiquette	1	tags.0.value	Toute valeur de chaîne. Remarque : le nombre dans le nom de la propriété fait référence à l'ordre de l'étiquette dans la liste d'évaluation des étiquettes. Comportement du paramètre : - *obligatoire* lors de la configuration des *Étiquettes d'hôtes* Ce paramètre n'est pas pris en charge lors de la configuration du widget sur un tableau de bord de modèle.
Afficher les hôtes en maintenance	0	maintenance	0 - *(par défaut)* Désactivé ; 1 - Activé.
Colonnes (voir ci-dessous)			
Trier par	0	column	Valeur numérique de la colonne parmi les colonnes configurées.
Ordre	0	order	2 - *(par défaut)* N premiers ; 3 - N derniers.
Limite d'hôtes	0	show_lines	Les valeurs possibles vont de 1 à 1000. Valeur par défaut : 10. Ce paramètre n'est pas pris en charge lors de la configuration du widget sur un tableau de bord de modèle.

Colonnes

Les colonnes ont des paramètres communs et des paramètres supplémentaires selon la configuration du paramètre *Data*.

Note:
Pour tous les paramètres liés aux colonnes, le numéro dans le nom de la propriété (par exemple columns.0.name) fait référence à une colonne pour laquelle le paramètre est configuré.

Les paramètres suivants sont pris en charge pour toutes les colonnes.

Paramètre	type	name	value
Nom	1	columns.0.name	Toute valeur de chaîne. Comportement du paramètre : - *obligatoire*
Données	0	columns.0.data	1 - Valeur de l'élément ; 2 - Nom de l'hôte ; 3 - Texte. Comportement du paramètre : - *obligatoire*
Couleur de base	1	columns.0.base_color	Code couleur hexadécimal (par ex. FF0000). Comportement du paramètre : - *obligatoire*

Valeur de l'élément

Les paramètres suivants sont pris en charge si *Data* est défini sur « Item value ».

Note:
Le premier nombre dans le nom de la propriété *Thresholds* (par exemple, columnsthresholds.0.color.0) fait référence à la colonne pour laquelle les seuils sont configurés, tandis que le second nombre fait référence à la position du seuil dans une liste triée par ordre croissant. Cependant, si les seuils sont configurés dans un ordre différent, les valeurs seront triées par ordre croissant après la mise à jour de la configuration du widget dans l'interface Zabbix (par exemple, `"threshold.0.threshold":"5"` → `"threshold.0.threshold":"1"` ; `"threshold.1.threshold":"1"` → `"threshold.1.threshold": "5"`).

Paramètre	type	name	value
Élément	1	columns.0.item	Nom d'élément valide. Lors de la configuration du widget sur un tableau de bord de modèle, seuls les éléments configurés sur le modèle doivent être définis.
Afficher la valeur de l'élément en tant que	0	columns.0.display_value_as	0 - *(par défaut)* Numérique ; 1 - Texte ; 2 - Binaire.
Affichage	0	columns.0.display	1 - *(par défaut)* Tel quel ; 2 - Barre ; 3 - Indicateurs ; 6 - Sparkline. Comportement du paramètre : - *pris en charge* si *Afficher la valeur de l'élément en tant que* est défini sur « Numérique »
Min	1	columns.0.min	Toute valeur numérique. Comportement du paramètre : - *pris en charge* si *Afficher la valeur de l'élément en tant que* est défini sur « Numérique » et si *Affichage* est défini sur « Barre » ou « Indicateurs »
Max	1	columns.0.max	Toute valeur numérique. Comportement du paramètre : - *pris en charge* si *Afficher la valeur de l'élément en tant que* est défini sur « Numérique » et si *Affichage* est défini sur « Barre » ou « Indicateurs »

Paramètre	type	name	value
Décimales	0	columns.0.decimal_places	Les valeurs possibles vont de 0 à 10. Valeur par défaut : 2. Comportement du paramètre : - *pris en charge* si *Afficher la valeur de l'élément en tant que* est défini sur « Numérique »
Sparkline			
Largeur	0	columns.0.sparkline.width	Les valeurs possibles vont de 0 à 10. Valeur par défaut : 1. Comportement du paramètre : - *pris en charge* si *Afficher la valeur de l'élément en tant que* est défini sur « Numérique » et si *Affichage* est défini sur « Sparkline »
Remplissage	0	columns.0.sparkline.fill	Les valeurs possibles vont de 0 à 10. Valeur par défaut : 3. Comportement du paramètre : - *pris en charge* si *Afficher la valeur de l'élément en tant que* est défini sur « Numérique » et si *Affichage* est défini sur « Sparkline »
Couleur	1	columns.0.sparkline.color	Code couleur hexadécimal (par exemple, `FF0000`). Valeur par défaut : `42A5F5`. Comportement du paramètre : - *pris en charge* si *Afficher la valeur de l'élément en tant que* est défini sur « Numérique » et si *Affichage* est défini sur « Sparkline »
Période	1	columns.0.sparkline.time_period	`DASHBOARD._timeperiod` - définir le sélecteur de période du tableau de bord comme source de données ; `ABCDE._timeperiod` - définir un widget compatible (avec son paramètre `reference` égal à `ABCDE`) comme source de données. Valeur par défaut : `DASHBOARD._timeperiod` Vous pouvez également définir la période uniquement dans les paramètres *De* et *À*. Comportement du paramètre : - *pris en charge* si *Afficher la valeur de l'élément en tant que* est défini sur « Numérique » et si *Affichage* est défini sur « Sparkline »
De	1	columns.0.sparkline.time_period.from	Chaîne temporelle valide au format absolu (`YYYY-MM-DD hh:mm:ss`) ou au format temporel relatif (`now`, `now/d`, `now/w-1w`, etc.). Comportement du paramètre : - *pris en charge* si *Afficher la valeur de l'élément en tant que* est défini sur « Numérique » et si *Affichage* est défini sur « Sparkline » - *requis* si `columns.0.sparkline.time_period.to` est défini
À	1	columns.0.sparkline.time_period.to	Chaîne temporelle valide au format absolu (`YYYY-MM-DD hh:mm:ss`) ou au format temporel relatif (`now`, `now/d`, `now/w-1w`, etc.). Comportement du paramètre : - *pris en charge* si *Afficher la valeur de l'élément en tant que* est défini sur « Numérique » et si *Affichage* est défini sur « Sparkline » - *requis* si `columns.0.sparkline.time_period.from` est défini

Paramètre	type	name	value
Données d'historique	0	columns.0.sparkline.history	0 - *(par défaut)* Automatique ; 1 - Historique ; 2 - Tendances. Comportement du paramètre : - *pris en charge* si *Afficher la valeur de l'élément en tant que* est défini sur « Numérique » et si *Affichage* est défini sur « Sparkline »
Seuils			
Couleur	1	columnsthresholds.0.color.0	Code couleur hexadécimal (par exemple, FF0000). Comportement du paramètre : - *pris en charge* si *Afficher la valeur de l'élément en tant que* est défini sur « Numérique »
Seuil	1	columnsthresholds.0.threshold.0	Toute valeur de chaîne. Comportement du paramètre : - *pris en charge* si *Afficher la valeur de l'élément en tant que* est défini sur « Numérique »
Mises en évidence			
Couleur	1	columns.0.highlights.0.color	Code couleur hexadécimal (par exemple, FF0000). Comportement du paramètre : - *pris en charge* si *Afficher la valeur de l'élément en tant que* est défini sur « Texte »
Motif	1	columns.0.highlights.0.pattern	Toute valeur de chaîne. Comportement du paramètre : - *pris en charge* si *Afficher la valeur de l'élément en tant que* est défini sur « Texte »
Afficher la miniature	0	columns.0.show_thumbnail	0 - *(par défaut)* Désactivé ; 1 - Activé. Comportement du paramètre : - *pris en charge* si *Afficher la valeur de l'élément en tant que* est défini sur « Binaire »
Fonction d'agrégation	0	columns.0.aggregate_function	0 - *(par défaut)* non utilisée ; 1 - min ; 2 - max ; 3 - avg ; 4 - count ; 5 - sum ; 6 - first ; 7 - last.
Période	1	columns.0.time_period.reference	`DASHBOARD._timeperiod` - définir le sélecteur de période du tableau de bord comme source de données ; `ABCDE._timeperiod` - définir un widget compatible (avec son paramètre `reference` égal à `ABCDE`) comme source de données. Valeur par défaut : `DASHBOARD._timeperiod` Vous pouvez également définir la période uniquement dans les paramètres *De* et *À*. Comportement du paramètre : - *pris en charge* si *Fonction d'agrégation* est définie sur « min », « max », « avg », « count », « sum », « first » ou « last »

Paramètre	type	name	value
De	1	columns.0.time_period.from	Chaîne temporelle valide au format absolu (`YYYY-MM-DD hh:mm:ss`) ou au format temporel relatif (`now`, `now/d`, `now/w-1w`, etc.). Comportement du paramètre : - *pris en charge* si *Période* n'est pas définie et si *Fonction d'agrégation* est définie sur « min », « max », « avg », « count », « sum », « first » ou « last » - *requis* si `columns.0.time_period.to` est défini
À	1	columns.0.time_period.to	Chaîne temporelle valide au format absolu (`YYYY-MM-DD hh:mm:ss`) ou au format temporel relatif (`now`, `now/d`, `now/w-1w`, etc.). Comportement du paramètre : - *pris en charge* si *Période* n'est pas définie et si *Fonction d'agrégation* est définie sur « min », « max », « avg », « count », « sum », « first » ou « last » - *requis* si `columns.0.time_period.from` est défini
Données d'historique	0	columns.0.history	0 - *(par défaut)* Automatique ; 1 - Historique ; 2 - Tendances. Comportement du paramètre : - *pris en charge* si *Afficher la valeur de l'élément en tant que* est défini sur « Numérique »
Référence	1	reference	Toute valeur de chaîne composée de 5 caractères (par exemple, `ABCDE` ou `JBPNL`). Cette valeur doit être unique au sein du tableau de bord auquel appartient le widget. Comportement du paramètre : - *requis*

Texte

Les paramètres suivants sont pris en charge si *Data* est défini sur "Text".

Parameter	type	name	value
Text	1	columns.0.text	Toute valeur de chaîne, y compris les macros. Macros prises en charge : `{HOST.*}`, `{INVENTORY.*}`. Comportement du paramètre : - *required* si *Data* est défini sur "Text"

Exemples

Les exemples suivants visent uniquement à décrire la configuration des objets de champ de widget de tableau de bord pour le widget *Top hosts*. Pour plus d'informations sur la configuration d'un tableau de bord, consultez `dashboard.create`.

Configuration d'un widget *Top hosts*

Configurez un widget *Top hosts* qui affiche les principaux hôtes selon l'utilisation du CPU dans le groupe d'hôtes « 4 ». En outre, configurez les colonnes personnalisées suivantes : « Nom de l'hôte », « Utilisation du CPU en % », « Moy. 1 min », « Moy. 5 min », « Moy. 15 min », « Processus ».

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": 30,
```

```
        "auto_start": 1,
        "pages": [
            {
                "widgets": [
                    {
                        "type": "tophosts",
                        "name": "Top hosts",
                        "x": 0,
                        "y": 0,
                        "width": 36,
                        "height": 5,
                        "view_mode": 0,
                        "fields": [
                            {
                                "type": 2,
                                "name": "groupids.0",
                                "value": 4
                            },
                            {
                                "type": 1,
                                "name": "columns.0.name",
                                "value": "Host"
                            },
                            {
                                "type": 0,
                                "name": "columns.0.data",
                                "value": 2
                            },
                            {
                                "type": 1,
                                "name": "columns.0.base_color",
                                "value": "FFFFFF"
                            },
                            {
                                "type": 1,
                                "name": "columns.1.name",
                                "value": "CPU utilization in %"
                            },
                            {
                                "type": 0,
                                "name": "columns.1.data",
                                "value": 1
                            },
                            {
                                "type": 1,
                                "name": "columns.1.base_color",
                                "value": "4CAF50"
                            },
                            {
                                "type": 1,
                                "name": "columns.1.item",
                                "value": "CPU utilization"
                            },
                            {
                                "type": 0,
                                "name": "columns.1.display",
                                "value": 3
                            },
                            {
                                "type": 1,
                                "name": "columns.1.min",
                                "value": "0"
```

```
                    },
                    {
                        "type": 1,
                        "name": "columns.1.max",
                        "value": "100"
                    },
                    {
                        "type": 1,
                        "name": "columnsthresholds.1.color.0",
                        "value": "FFFF00"
                    },
                    {
                        "type": 1,
                        "name": "columnsthresholds.1.threshold.0",
                        "value": "50"
                    },
                    {
                        "type": 1,
                        "name": "columnsthresholds.1.color.1",
                        "value": "FF8000"
                    },
                    {
                        "type": 1,
                        "name": "columnsthresholds.1.threshold.1",
                        "value": "80"
                    },
                    {
                        "type": 1,
                        "name": "columnsthresholds.1.color.2",
                        "value": "FF4000"
                    },
                    {
                        "type": 1,
                        "name": "columnsthresholds.1.threshold.2",
                        "value": "90"
                    },
                    {
                        "type": 1,
                        "name": "columns.2.name",
                        "value": "1m avg"
                    },
                    {
                        "type": 0,
                        "name": "columns.2.data",
                        "value": 1
                    },
                    {
                        "type": 1,
                        "name": "columns.2.base_color",
                        "value": "FFFFFF"
                    },
                    {
                        "type": 1,
                        "name": "columns.2.item",
                        "value": "Load average (1m avg)"
                    },
                    {
                        "type": 1,
                        "name": "columns.3.name",
                        "value": "5m avg"
                    },
                    {
```

```
            "type": 0,
            "name": "columns.3.data",
            "value": 1
        },
        {
            "type": 1,
            "name": "columns.3.base_color",
            "value": "FFFFFF"
        },
        {
            "type": 1,
            "name": "columns.3.item",
            "value": "Load average (5m avg)"
        },
        {
            "type": 1,
            "name": "columns.4.name",
            "value": "15m avg"
        },
        {
            "type": 0,
            "name": "columns.4.data",
            "value": 1
        },
        {
            "type": 1,
            "name": "columns.4.base_color",
            "value": "FFFFFF"
        },
        {
            "type": 1,
            "name": "columns.4.item",
            "value": "Load average (15m avg)"
        },
        {
            "type": 1,
            "name": "columns.5.name",
            "value": "Processes"
        },
        {
            "type": 0,
            "name": "columns.5.data",
            "value": 1
        },
        {
            "type": 1,
            "name": "columns.5.base_color",
            "value": "FFFFFF"
        },
        {
            "type": 1,
            "name": "columns.5.item",
            "value": "Number of processes"
        },
        {
            "type": 0,
            "name": "columns.5.decimal_places",
            "value": 0
        },
        {
            "type": 0,
            "name": "column",
```

```
                                "value": 1
                            }
                        ]
                    }
                ]
            }
        ],
        "userGroups": [
            {
                "usrgrpid": 7,
                "permission": 2
            }
        ],
        "users": [
            {
                "userid": 1,
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Voir aussi

- Champ de widget de tableau de bord
- `dashboard.create`
- `dashboard.update`

29 Éléments principaux

Description

Ces paramètres et les valeurs de propriété possibles pour les objets de champ respectifs du widget de tableau de bord vous permettent de configurer le widget *Éléments principaux* dans les méthodes `dashboard.create` et `dashboard.update`.

Attention:
Les propriétés `fields` du widget ne sont pas validées lors de la création ou de la mise à jour d'un tableau de bord. Cela permet aux utilisateurs de modifier les widgets intégrés et de créer des widgets personnalisés, mais introduit également le risque de créer ou de mettre à jour des widgets de manière incorrecte. Pour garantir la création ou la mise à jour réussie du widget *Éléments principaux*, veuillez vous reporter au comportement des paramètres décrit dans les tableaux ci-dessous.

Paramètres

Les paramètres suivants sont pris en charge pour le widget *Top items*.

Paramètre	type	name	value
Intervalle d'actualisation	0	rf_rate	0 - Pas d'actualisation ; 10 - 10 secondes ; 30 - 30 secondes ; 60 - *(par défaut)* 1 minute ; 120 - 2 minutes ; 600 - 10 minutes ; 900 - 15 minutes.
Groupes d'hôtes	2	groupids.0	ID du groupe d'hôtes. Remarque : Pour configurer plusieurs groupes d'hôtes, créez un objet de champ de widget de tableau de bord pour chaque groupe d'hôtes avec un numéro incrémenté dans le nom de propriété. Ce paramètre n'est pas pris en charge lors de la configuration du widget sur un tableau de bord de modèle.
Groupes d'hôtes (Widget)	1	groupids._reference	Au lieu de l'ID du groupe d'hôtes : `ABCDE._hostgroupids` - définir un widget compatible (avec son paramètre *Reference* défini sur "`ABCDE`") comme source de données pour les groupes d'hôtes. Ce paramètre n'est pas pris en charge lors de la configuration du widget sur un tableau de bord de modèle.
Hôtes	3	hostids.0	ID de l'hôte. Remarque : Pour configurer plusieurs hôtes, créez un objet de champ de widget de tableau de bord pour chaque hôte avec un numéro incrémenté dans le nom de propriété. Pour plusieurs hôtes, le paramètre *Groupes d'hôtes* ne doit soit pas être configuré du tout, soit être configuré avec au moins un groupe d'hôtes auquel appartiennent les hôtes configurés. Ce paramètre n'est pas pris en charge lors de la configuration du widget sur un tableau de bord de modèle.
Hôtes (Widget/Tableau de bord)	1	hostids._reference	Au lieu de l'ID de l'hôte : `DASHBOARD.hostids` - définir le sélecteur d'hôte du tableau de bord comme source de données pour les hôtes ; `ABCDE._hostids` - définir un widget compatible (avec son paramètre *Reference* défini sur "`ABCDE`") comme source de données pour les hôtes. Ce paramètre n'est pas pris en charge lors de la configuration du widget sur un tableau de bord de modèle.
Tags d'hôte			
Type d'évaluation	0	host_tags_evaltype	0 - *(par défaut)* Et/Ou ; 2 - Ou. Ce paramètre n'est pas pris en charge lors de la configuration du widget sur un tableau de bord de modèle.
Nom du tag	1	host_tags.0.tag	Toute valeur de chaîne. Remarque : Le numéro dans le nom de propriété fait référence à l'ordre du tag dans la liste d'évaluation des tags. Comportement du paramètre : - *obligatoire* lors de la configuration de *Tags d'hôte* Ce paramètre n'est pas pris en charge lors de la configuration du widget sur un tableau de bord de modèle.

Paramètre	type	name	value
Opérateur	0	host_tags.0.operator	0 - Contient ; 1 - Égal ; 2 - Ne contient pas ; 3 - N'est pas égal ; 4 - Existe ; 5 - N'existe pas. Remarque : Le numéro dans le nom de propriété fait référence à l'ordre du tag dans la liste d'évaluation des tags. Comportement du paramètre : - *obligatoire* lors de la configuration de *Tags d'hôte* Ce paramètre n'est pas pris en charge lors de la configuration du widget sur un tableau de bord de modèle.
Valeur du tag	1	host_tags.0.value	Toute valeur de chaîne. Remarque : Le numéro dans le nom de propriété fait référence à l'ordre du tag dans la liste d'évaluation des tags. Comportement du paramètre : - *obligatoire* lors de la configuration de *Tags d'hôte* Ce paramètre n'est pas pris en charge lors de la configuration du widget sur un tableau de bord de modèle.
Disposition	0	layout	0 - *(par défaut)* Horizontale ; 1 - Verticale.
Afficher les problèmes	0	show_problems	0 - Tous ; 1 - *(par défaut)* Non supprimés ; 2 - Aucun.
Éléments (voir ci-dessous)			

Configuration avancée

Les paramètres de configuration avancée suivants sont pris en charge pour le widget *Éléments principaux*.

Parameter	type	name	value
Classement des hôtes			
Classer par	0	host_ordering_order_by	0 - *(par défaut)* Nom de l'hôte ; 3 - Valeur de l'élément.
Modèles d'éléments	1	host_ordering_item.0	Nom ou modèle de l'élément (par exemple, "*: Nombre de *valeurs traitées par seconde"). Remarque : pour configurer plusieurs modèles d'éléments, créez un objet de champ de widget de tableau de bord pour chaque modèle d'élément, avec un numéro incrémenté dans le nom de la propriété. Lors de la configuration du widget sur un tableau de bord de modèle, seuls les modèles correspondant aux éléments configurés sur le modèle doivent être définis. Comportement du paramètre : - *obligatoire* si *Classer par* est défini sur « Valeur de l'élément »

Parameter	type	name	value
Ordre	0	host_ordering_order	2 - *(par défaut)* N premiers ; 3 - N derniers.
Limite	0	host_ordering_limit	Les valeurs possibles vont de 1 à 100. Valeur par défaut : 10.
Classement des éléments			
Classer par	0	item_ordering_order_by	1 - Hôte ; 2 - Nom de l'élément ; 3 - *(par défaut)* Valeur de l'élément.
Modèles d'hôtes	1	item_ordering_host.0	Nom ou modèle de l'hôte. Remarque : pour configurer plusieurs modèles d'hôtes, créez un objet de champ de widget de tableau de bord pour chaque modèle d'hôte, avec un numéro incrémenté dans le nom de la propriété. Comportement du paramètre : - *obligatoire* si *Classer par* est défini sur « Hôte »
Ordre	0	item_ordering_order	2 - *(par défaut)* N premiers ; 3 - N derniers.
Limite	0	item_ordering_limit	Les valeurs possibles vont de 1 à 100. Valeur par défaut : 10.

Colonnes

Note:
Le premier nombre dans le nom de la propriété (par exemple, columns.0.items.0, columns.0.item_tags_evaltype) représente la colonne concernée, tandis que le deuxième nombre, s'il est présent, représente l'entité configurée (par exemple, le modèle d'élément, l'étiquette).

Paramètre	type	name	value
Modèles d'éléments	1	columns.0.items.0	Nom ou modèle de l'élément (par exemple, "*: Nombre de *valeurs traitées par seconde"). Remarque : pour configurer plusieurs modèles d'éléments, créez un objet de champ de widget de tableau de bord pour chaque modèle d'élément, avec un deuxième nombre incrémenté dans le nom de la propriété. Lors de la configuration du widget sur un tableau de bord de modèle, seuls les modèles des éléments configurés sur le modèle doivent être définis. Comportement du paramètre : - *obligatoire*
Étiquettes des éléments			
Type d'évaluation	0	columns.0.item_tags_evaltype	0 - *(par défaut)* Et/Ou ; 2 - Ou.

Paramètre	type	name	value
Nom de l'étiquette	1	columns.0.item_tags.0.tag	Toute valeur de chaîne. Remarque : le deuxième nombre dans le nom de la propriété fait référence à l'ordre de l'étiquette dans la liste d'évaluation des étiquettes. Comportement du paramètre : - *obligatoire* si vous configurez *Étiquettes des éléments*
Opérateur	0	columns.0.item_tags.0.operator	0 - Contient ; 1 - Est égal à ; 2 - Ne contient pas ; 3 - N'est pas égal à ; 4 - Existe ; 5 - N'existe pas. Remarque : le deuxième nombre dans le nom de la propriété fait référence à l'ordre de l'étiquette dans la liste d'évaluation des étiquettes. Comportement du paramètre : - *obligatoire* si vous configurez *Étiquettes des éléments*
Valeur de l'étiquette	1	columns.0.item_tags.0.value	Toute valeur de chaîne. Remarque : le deuxième nombre dans le nom de la propriété fait référence à l'ordre de l'étiquette dans la liste d'évaluation des étiquettes. Comportement du paramètre : - *obligatoire* si vous configurez *Étiquettes des éléments*
Couleur de base	1	columns.0.base_color	Code couleur hexadécimal (par exemple, FF0000).
Afficher la valeur comme	0	columns.0.display_value_as	1 - *(par défaut)* Numérique ; 2 - Texte.
Affichage	0	columns.0.display	1 - *(par défaut)* Tel quel ; 2 - Barre ; 3 - Indicateurs ; 6 - Sparkline.
Min	1	columns.0.min	Toute valeur numérique. Les suffixes (par exemple, "1d", "2w", "4K", "8G") sont pris en charge. Comportement du paramètre : - *pris en charge* si *Afficher la valeur comme* est défini sur "Numérique" et *Affichage* sur "Barre" ou "Indicateurs"
Max	1	columns.0.max	Toute valeur numérique. Les suffixes (par exemple, "1d", "2w", "4K", "8G") sont pris en charge. Comportement du paramètre : - *pris en charge* si *Afficher la valeur comme* est défini sur "Numérique" et *Affichage* sur "Barre" ou "Indicateurs"
Sparkline			
Largeur	0	columns.0.sparkline.width	Les valeurs possibles vont de 0 à 10. Valeur par défaut : 1. Comportement du paramètre : - *pris en charge* si *Affichage* est défini sur "Sparkline"

Paramètre	type	name	value
Remplissage	0	columns.0.sparkline.fill	Les valeurs possibles vont de 0 à 10. Valeur par défaut : 3. Comportement du paramètre : - *pris en charge* si *Affichage* est défini sur ”Sparkline”
Couleur	1	columns.0.sparkline.color	Code couleur hexadécimal (par exemple, `FF0000`). Valeur par défaut : `42A5F5`. Comportement du paramètre : - *pris en charge* si *Affichage* est défini sur ”Sparkline”
Période	1	columns.0.sparkline.time_period	`DASHBOARD._timeperiod` - définir le sélecteur de période du tableau de bord comme source de données ; `ABCDE._timeperiod` - définir un widget compatible (avec son paramètre `reference` égal à `ABCDE`) comme source de données. Valeur par défaut : `DASHBOARD._timeperiod` Vous pouvez également définir la période uniquement dans les paramètres *De* et *À*. Comportement du paramètre : - *pris en charge* si *Affichage* est défini sur ”Sparkline”
De	1	columns.0.sparkline.time_period.from	Chaîne temporelle valide au format absolu (`YYYY-MM-DD hh:mm:ss`) ou selon la syntaxe temporelle relative (`now`, `now/d`, `now/w-1w`, etc.). Comportement du paramètre : - *pris en charge* si *Affichage* est défini sur ”Sparkline” - *obligatoire* si `columns.0.sparkline.time_period.to` est défini
À	1	columns.0.sparkline.time_period.to	Chaîne temporelle valide au format absolu (`YYYY-MM-DD hh:mm:ss`) ou selon la syntaxe temporelle relative (`now`, `now/d`, `now/w-1w`, etc.). Comportement du paramètre : - *pris en charge* si *Affichage* est défini sur ”Sparkline” - *obligatoire* si `columns.0.sparkline.time_period.from` est défini
Données d’historique	0	columns.0.sparkline.history	0 *(par défaut)* Automatique ; 1 - Historique ; 2 - Tendances. Comportement du paramètre : - *pris en charge* si *Affichage* est défini sur ”Sparkline”
Seuils			
Couleur	1	columns.0.thresholds.color	Code couleur hexadécimal (par exemple, `FF0000`). Comportement du paramètre : - *pris en charge* si *Afficher la valeur comme* est défini sur ”Numérique”
Seuil	1	columns.0.thresholds.threshold	Une valeur numérique. Les suffixes (par exemple, ”1d”, ”2w”, ”4K”, ”8G”) sont pris en charge. Comportement du paramètre : - *pris en charge* si *Afficher la valeur comme* est défini sur ”Numérique”

Paramètre	type	name	value
Mises en évidence			
Mise en évidence	1	columns.0.highlights.0.color	Code couleur hexadécimal (par exemple, `FF0000`). Comportement du paramètre : - *pris en charge* si *Afficher la valeur comme* est défini sur "Texte"
Seuil	1	columns.0.highlights.0.pattern	Texte ou expression régulière. Comportement du paramètre : - *pris en charge* si *Afficher la valeur comme* est défini sur "Texte"
Décimales	0	columns.0.decimal_places	Les valeurs possibles vont de 0 à 10. Valeur par défaut : 2.
Configuration avancée			
Fonction d'agrégation	0	columns.0.aggregate_function	Si *Afficher la valeur comme* est défini sur "Numérique" : 0 - *(par défaut)* non utilisée ; 1 - min ; 2 - max ; 3 - moyenne ; 4 - nombre ; 5 - somme ; 6 - première ; 7 - dernière. Si *Afficher la valeur comme* est défini sur "Texte" : 0 - *(par défaut)* non utilisée ; 4 - nombre ; 6 - première ; 7 - dernière.
Période	1	columns.0.time_period.reference	`DASHBOARD._timeperiod` - définir le sélecteur de période du tableau de bord comme source de données ; `ABCDE._timeperiod` - définir un widget compatible (avec son paramètre `reference` égal à `ABCDE`) comme source de données. Valeur par défaut : `DASHBOARD._timeperiod` Vous pouvez également définir la période uniquement dans les paramètres *De* et *À*. Comportement du paramètre : - *pris en charge* si *Fonction d'agrégation* est définie sur "min", "max", "moyenne", "nombre", "somme", "première" ou "dernière"
De	1	columns.0.time_period.from	Chaîne temporelle valide au format absolu (`YYYY-MM-DD hh:mm:ss`) ou selon la syntaxe temporelle relative (`now`, `now/d`, `now/w-1w`, etc.). Comportement du paramètre : - *pris en charge* si *Période* n'est pas définie et si *Fonction d'agrégation* est définie sur "min", "max", "moyenne", "nombre", "somme", "première" ou "dernière" - *obligatoire* si `columns.0.time_period.to` est défini

Paramètre	type	name	value
À	1	columns.0.time_period.to	Chaîne temporelle valide au format absolu (`YYYY-MM-DD hh:mm:ss`) ou selon la syntaxe temporelle relative (`now`, `now/d`, `now/w-1w`, etc.). Comportement du paramètre : - *pris en charge* si *Période* n'est pas définie et si *Fonction d'agrégation* est définie sur "min", "max", "moyenne", "nombre", "somme", "première" ou "dernière" - *obligatoire* si `columns.0.time_period.from` est défini
Données d'historique	0	columns.0.history	0 - *(par défaut)* Automatique ; 1 - Historique ; 2 - Tendances. Comportement du paramètre : - *pris en charge* si *Afficher la valeur comme* est défini sur "Numérique"
Agrégation	0	columns.0.aggregate_function	0 - *(par défaut)* Chaque élément 1 - Combinés. Comportement du paramètre : - *pris en charge* si *Afficher la valeur comme* est défini sur "Numérique" et si *Affichage* est défini sur "Tel quel", "Barre" ou "Indicateurs".
Fonction d'agrégation combinée	0	columns.0.column_aggregate_function	1 - min ; 2 - max ; 3 - moyenne ; 4 - nombre ; 5 - somme *(par défaut)*. Comportement du paramètre : - *obligatoire* si vous configurez *Agrégation*.
Nom de la colonne combinée	1	columns.0.combined_column_name	Toute valeur de chaîne. Comportement du paramètre : - *obligatoire* si vous configurez *Agrégation*.

Exemples

Les exemples suivants visent uniquement à décrire la configuration des objets de champ de widget de tableau de bord pour le widget *Top items*. Pour plus d'informations sur la configuration d'un tableau de bord, consultez `dashboard.create`.

Configuration d'un widget *Top items*

Configurez un widget *Top items* qui affiche les données de l'hôte « 10084 » et uniquement pour les éléments dont le tag nommé « component » contient la valeur « cpu ». De plus, affichez les données avec les hôtes placés en haut et utilisez une jauge colorée pour la représentation des cellules.

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": 30,
        "auto_start": 1,
        "pages": [
            {
                "widgets": [
                    {
                        "type": "topitems",
```

```
                        "name": "Top items",
                        "x": 0,
                        "y": 0,
                        "width": 36,
                        "height": 5,
                        "view_mode": 0,
                        "fields": [
                            {
                                "type": 3,
                                "name": "hostids.0",
                                "value": 10084
                            },
                            {
                                "type": 1,
                                "name": "columns.0.items.0",
                                "value": "*"
                            },
                            {
                                "type": 1,
                                "name": "columns.0.item_tags.0.tag",
                                "value": "component"
                            },
                            {
                                "type": 0,
                                "name": "columns.0.item_tags.0.operator",
                                "value": 0
                            },
                            {
                                "type": 1,
                                "name": "columns.0.item_tags.0.value",
                                "value": "cpu"
                            },
                            {
                                "type": 0,
                                "name": "columns.0.display",
                                "value": 2
                            },
                            {
                                "type": 0,
                                "name": "layout",
                                "value": 1
                            }
                        ]
                    }
                ]
            }
        ],
        "userGroups": [
            {
                "usrgrpid": 7,
                "permission": 2
            }
        ],
        "users": [
            {
                "userid": 1,
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Voir aussi

- Champ de widget de tableau de bord
- `dashboard.create`
- `dashboard.update`

30 Déclencheurs principaux

Description

Ces paramètres et les valeurs de propriété possibles pour les objets de champ respectifs du widget de tableau de bord vous permettent de configurer le widget *Déclencheurs principaux* dans les méthodes `dashboard.create` et `dashboard.update`.

Attention:
Les propriétés `fields` du widget ne sont pas validées lors de la création ou de la mise à jour d'un tableau de bord. Cela permet aux utilisateurs de modifier les widgets intégrés et de créer des widgets personnalisés, mais introduit également le risque de créer ou de mettre à jour des widgets de manière incorrecte. Pour garantir la création ou la mise à jour réussie du widget *Déclencheurs principaux*, veuillez vous reporter au comportement des paramètres décrit dans les tableaux ci-dessous.

Paramètres

Les paramètres suivants sont pris en charge pour le widget *Déclencheurs principaux*.

Paramètre	type	name	value
Intervalle d'actualisation	0	rf_rate	0 - *(par défaut)* Pas d'actualisation ; 10 - 10 secondes ; 30 - 30 secondes ; 60 - 1 minute ; 120 - 2 minutes ; 600 - 10 minutes ; 900 - 15 minutes.
Groupes d'hôtes	2	groupids.0	ID du groupe d'hôtes. Remarque : pour configurer plusieurs groupes d'hôtes, créez un objet de champ de widget de tableau de bord pour chaque groupe d'hôtes, avec un numéro incrémenté dans le nom de la propriété. Ce paramètre n'est pas pris en charge lors de la configuration du widget sur un tableau de bord de modèle.
Hôtes	3	hostids.0	ID de l'hôte. Remarque : pour configurer plusieurs hôtes, créez un objet de champ de widget de tableau de bord pour chaque hôte, avec un numéro incrémenté dans le nom de la propriété. Pour plusieurs hôtes, le paramètre *Groupes d'hôtes* doit soit ne pas être configuré, soit être configuré avec au moins un groupe d'hôtes auquel appartiennent les hôtes configurés. Ce paramètre n'est pas pris en charge lors de la configuration du widget sur un tableau de bord de modèle.

Paramètre	type	name	value
Problème	1	problem	Nom de l'événement du problème (insensible à la casse, nom complet ou partie du nom).
Gravité	0	severities.0	Gravités des déclencheurs. 0 - Non classé ; 1 - Information ; 2 - Avertissement ; 3 - Moyenne ; 4 - Haute ; 5 - Désastre. Par défaut : vide (toutes activées). Remarque : pour configurer plusieurs valeurs, créez un objet de champ de widget de tableau de bord pour chaque valeur, avec un numéro incrémenté dans le nom de la propriété.
Étiquettes des problèmes			
Type d'évaluation	0	evaltype	0 - *(par défaut)* Et/Ou ; 2 - Ou.
Nom de l'étiquette	1	tags.0.tag	Toute valeur de chaîne. Remarque : le numéro dans le nom de la propriété fait référence à l'ordre de l'étiquette dans la liste d'évaluation des étiquettes. Comportement du paramètre : - *obligatoire* lors de la configuration des *Étiquettes des problèmes*
Opérateur	0	tags.0.operator	0 - Contient ; 1 - Est égal à ; 2 - Ne contient pas ; 3 - N'est pas égal à ; 4 - Existe ; 5 - N'existe pas. Remarque : le numéro dans le nom de la propriété fait référence à l'ordre de l'étiquette dans la liste d'évaluation des étiquettes. Comportement du paramètre : - *obligatoire* lors de la configuration des *Étiquettes des problèmes*
Valeur de l'étiquette	1	tags.0.value	Toute valeur de chaîne. Remarque : le numéro dans le nom de la propriété fait référence à l'ordre de l'étiquette dans la liste d'évaluation des étiquettes. Comportement du paramètre : - *obligatoire* lors de la configuration des *Étiquettes des problèmes*
Période	1	time_period._reference	`DASHBOARD._timeperiod` - définir le sélecteur de période du tableau de bord comme source de données ; `ABCDE._timeperiod` - définir un widget compatible (avec son paramètre *Référence* défini sur "`ABCDE`") comme source de données. Par défaut : `DASHBOARD._timeperiod` Vous pouvez également définir la période uniquement dans les paramètres *Du* et *Au*.

Paramètre	type	name	value
Du	1	time_period.from	Chaîne temporelle valide au format absolu (`YYYY-MM-DD hh:mm:ss`) ou au format temporel relatif (`now`, `now/d`, `now/w-1w`, etc.). Comportement du paramètre : - *pris en charge* si *Période* n'est pas défini - *obligatoire* si `time_period.to` est défini
Au	1	time_period.to	Chaîne temporelle valide au format absolu (`YYYY-MM-DD hh:mm:ss`) ou au format temporel relatif (`now`, `now/d`, `now/w-1w`, etc.). Comportement du paramètre : - *pris en charge* si *Période* n'est pas défini - *obligatoire* si `time_period.from` est défini
Limite de dé-clencheurs	0	show_lines	Les valeurs possibles vont de 1 à 100. Par défaut : 10.

Exemples

Les exemples suivants visent uniquement à décrire la configuration des objets de champ de widget de tableau de bord pour le widget *Top triggers*. Pour plus d'informations sur la configuration d'un tableau de bord, consultez `dashboard.create`.

Configuration d'un widget *Top triggers*

Configurez un widget *Top triggers* qui affiche les 5 principaux déclencheurs pour le groupe d'hôtes « 4 » avec le nombre de tous les problèmes pour chaque déclencheur. Le widget affiche uniquement les déclencheurs ayant les sévérités « Warning », « Average », « High » ou « Disaster », ainsi que les problèmes ayant un tag nommé « scope » contenant les valeurs « performance », « availability » ou « capacity ».

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": 30,
        "auto_start": 1,
        "pages": [
            {
                "widgets": [
                    {
                        "type": "toptriggers",
                        "name": "Top triggers",
                        "x": 0,
                        "y": 0,
                        "width": 36,
                        "height": 5,
                        "view_mode": 0,
                        "fields": [
                            {
                                "type": 2,
                                "name": "groupids.0",
                                "value": 4
                            },
                            {
                                "type": 0,
                                "name": "severities.0",
                                "value": 2
                            },
                            {
```

```
            "type": 0,
            "name": "severities.1",
            "value": 3
        },
        {
            "type": 0,
            "name": "severities.2",
            "value": 4
        },
        {
            "type": 0,
            "name": "severities.3",
            "value": 5
        },
        {
            "type": 1,
            "name": "tags.0.tag",
            "value": "scope"
        },
        {
            "type": 0,
            "name": "tags.0.operator",
            "value": 0
        },
        {
            "type": 1,
            "name": "tags.0.value",
            "value": "performance"
        },
        {
            "type": 1,
            "name": "tags.1.tag",
            "value": "scope"
        },
        {
            "type": 0,
            "name": "tags.1.operator",
            "value": 0
        },
        {
            "type": 1,
            "name": "tags.1.value",
            "value": "availability"
        },
        {
            "type": 1,
            "name": "tags.2.tag",
            "value": "scope"
        },
        {
            "type": 0,
            "name": "tags.2.operator",
            "value": 0
        },
        {
            "type": 1,
            "name": "tags.2.value",
            "value": "capacity"
        },
        {
            "type": 0,
            "name": "show_lines",
```

```
                                "value": 5
                            }
                        ]
                    }
                ]
            }
        ],
        "userGroups": [
            {
                "usrgrpid": 7,
                "permission": 2
            }
        ],
        "users": [
            {
                "userid": 1,
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Voir aussi

- Champ de widget de tableau de bord
- `dashboard.create`
- `dashboard.update`

31 Aperçu des déclencheurs

Description

Ces paramètres et les valeurs de propriété possibles pour les objets de champ respectifs du widget de tableau de bord vous permettent de configurer le widget *Vue d'ensemble des déclencheurs* dans les méthodes `dashboard.create` et `dashboard.update`.

Attention:
Les propriétés `fields` du widget ne sont pas validées lors de la création ou de la mise à jour d'un tableau de bord. Cela permet aux utilisateurs de modifier les widgets intégrés et de créer des widgets personnalisés, mais introduit également le risque de créer ou de mettre à jour des widgets de manière incorrecte. Pour garantir la création ou la mise à jour réussie du widget *Vue d'ensemble des déclencheurs*, veuillez vous reporter au comportement des paramètres décrit dans les tableaux ci-dessous.

Paramètres

Les paramètres suivants sont pris en charge pour le widget *Aperçu des déclencheurs*.

Paramètre	type	name	value
Intervalle d'actualisation	0	rf_rate	0 - Pas d'actualisation ; 10 - 10 secondes ; 30 - 30 secondes ; 60 - *(par défaut)* 1 minute ; 120 - 2 minutes ; 600 - 10 minutes ; 900 - 15 minutes.
Afficher	0	show	1 - *(par défaut)* Problèmes récents ; 2 - Tous ; 3 - Problèmes.
Groupes d'hôtes	2	groupids.0	ID du groupe d'hôtes. Remarque : Pour configurer plusieurs groupes d'hôtes, créez un objet de champ de widget de tableau de bord pour chaque groupe d'hôtes avec un numéro incrémenté dans le nom de propriété. Ce paramètre n'est pas pris en charge lors de la configuration du widget sur un tableau de bord de modèle.
Groupes d'hôtes (Widget)	1	groupids._reference	Au lieu de l'ID du groupe d'hôtes : `ABCDE._hostgroupids` - définissez un widget compatible (avec son paramètre *Référence* défini sur "ABCDE") comme source de données pour les groupes d'hôtes. Ce paramètre n'est pas pris en charge lors de la configuration du widget sur un tableau de bord de modèle.
Hôtes	3	hostids.0	ID de l'hôte. Remarque : Pour configurer plusieurs hôtes, créez un objet de champ de widget de tableau de bord pour chaque hôte avec un numéro incrémenté dans le nom de propriété. Pour plusieurs hôtes, le paramètre *Groupes d'hôtes* ne doit soit pas être configuré du tout, soit être configuré avec au moins un groupe d'hôtes auquel appartiennent les hôtes configurés. Ce paramètre n'est pas pris en charge lors de la configuration du widget sur un tableau de bord de modèle.
Hôtes (Widget/Tableau de bord)	1	hostids._reference	Au lieu de l'ID de l'hôte : `DASHBOARD.hostids` - définissez le sélecteur d'hôte du tableau de bord comme source de données pour les hôtes ; `ABCDE._hostids` - définissez un widget compatible (avec son paramètre *Référence* défini sur "ABCDE") comme source de données pour les hôtes. Ce paramètre n'est pas pris en charge lors de la configuration du widget sur un tableau de bord de modèle.
Balises de problème			
Type d'évaluation	0	evaltype	0 - *(par défaut)* Et/Ou ; 2 - Ou.
Nom de balise	1	tags.0.tag	Toute valeur de chaîne. Remarque : Le numéro dans le nom de propriété fait référence à l'ordre des balises dans la liste d'évaluation des balises. Comportement du paramètre : - *obligatoire* lors de la configuration de *Balises de problème*

Paramètre	type	name	value
Opérateur	0	tags.0.operator	0 - Contient ; 1 - Égal à ; 2 - Ne contient pas ; 3 - Différent de ; 4 - Existe ; 5 - N'existe pas. Remarque : Le numéro dans le nom de propriété fait référence à l'ordre des balises dans la liste d'évaluation des balises.
Valeur de balise	1	tags.0.value	Comportement du paramètre : - *obligatoire* lors de la configuration de *Balises de problème* Toute valeur de chaîne. Remarque : Le numéro dans le nom de propriété fait référence à l'ordre des balises dans la liste d'évaluation des balises.
Afficher les problèmes supprimés	0	show_suppressed	Comportement du paramètre : - *obligatoire* lors de la configuration de *Balises de problème* 0 - *(par défaut)* Désactivé ; 1 - Activé.
Disposition	0	layout	0 - *(par défaut)* Horizontale ; 1 - Verticale.

Exemples

Les exemples suivants visent à décrire uniquement la configuration des objets de champ du widget de tableau de bord pour le widget *Aperçu des déclencheurs*. Pour plus d'informations sur la configuration d'un tableau de bord, consultez `dashboard.create`.

Configuration d'un widget *Aperçu des déclencheurs*

Configurez un widget *Aperçu des déclencheurs* qui affiche les états des déclencheurs pour tous les groupes d'hôtes ayant des déclencheurs avec un tag nommé "scope" et contenant la valeur "availability".

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": 30,
        "auto_start": 1,
        "pages": [
            {
                "widgets": [
                    {
                        "type": "trigover",
                        "name": "Trigger overview",
                        "x": 0,
                        "y": 0,
                        "width": 36,
                        "height": 5,
                        "view_mode": 0,
                        "fields": [
                            {
                                "type": 1,
                                "name": "tags.0.tag",
                                "value": "scope"
```

```
                            },
                            {
                                "type": 0,
                                "name": "tags.0.operator",
                                "value": 0
                            },
                            {
                                "type": 1,
                                "name": "tags.0.value",
                                "value": "availability"
                            }
                        ]
                    }
                ]
            }
        ],
        "userGroups": [
            {
                "usrgrpid": 7,
                "permission": 2
            }
        ],
        "users": [
            {
                "userid": 1,
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Voir également

- Champ du widget du tableau de bord
- `dashboard.create`
- `dashboard.update`

32 URL

Description

Ces paramètres et les valeurs de propriété possibles pour les objets de champ respectifs du widget de tableau de bord vous permettent de configurer le widget *URL* dans les méthodes `dashboard.create` et `dashboard.update`.

Attention:
Les propriétés `fields` du widget ne sont pas validées lors de la création ou de la mise à jour d'un tableau de bord. Cela permet aux utilisateurs de modifier les widgets intégrés et de créer des widgets personnalisés, mais introduit également le risque de créer ou de mettre à jour des widgets de manière incorrecte. Pour garantir la création ou la mise à jour réussie du widget *URL*, veuillez vous reporter au comportement des paramètres décrit dans les tableaux ci-dessous.

Paramètres

Les paramètres suivants sont pris en charge pour le widget *URL*.

Paramètre	type	name	value
Intervalle d'actualisation	0	rf_rate	0 - *(par défaut)* Pas d'actualisation; 10 - 10 secondes; 30 - 30 secondes; 60 - 1 minute; 120 - 2 minutes; 600 - 10 minutes; 900 - 15 minutes.
URL	1	url	Chaîne URL valide. Comportement du paramètre : - *obligatoire*
Remplacer l'hôte	1	override_hostid._reference	`ABCDE._hostid` - définir un widget compatible (avec son paramètre *Référence* défini sur "`ABCDE`") comme source de données pour les hôtes ; `DASHBOARD._hostid` - définir le sélecteur d'hôte du tableau de bord comme source de données pour les hôtes. Ce paramètre n'est pas pris en charge lors de la configuration du widget sur un tableau de bord de modèle.

Exemples

Les exemples suivants visent à décrire uniquement la configuration des objets de champ du widget de tableau de bord pour le widget *URL*. Pour plus d'informations sur la configuration d'un tableau de bord, consultez `dashboard.create`.

Configuration d'un widget *URL*

Configurez un widget *URL* qui affiche la page d'accueil du manuel Zabbix.

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": 30,
        "auto_start": 1,
        "pages": [
            {
                "widgets": [
                    {
                        "type": "url",
                        "name": "URL",
                        "x": 0,
                        "y": 0,
                        "width": 36,
                        "height": 5,
                        "view_mode": 0,
                        "fields": [
                            {
                                "type": 1,
                                "name": "url",
                                "value": "https://www.zabbix.com/documentation/8.0/en"
                            }
                        ]
                    }
                ]
            }
        ],
```

```
        "userGroups": [
            {
                "usrgrpid": 7,
                "permission": 2
            }
        ],
        "users": [
            {
                "userid": 1,
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Voir également

- Dashboard widget field
- `dashboard.create`
- `dashboard.update`

33 Surveillance Web

Description

Ces paramètres et les valeurs de propriété possibles pour les objets de champ respectifs du widget permettent de configurer le widget *Surveillance Web* dans les méthodes `dashboard.create` et `dashboard.update`.

Attention:
Les propriétés `fields` du widget ne sont pas validées lors de la création ou de la mise à jour d'un tableau de bord. Cela permet aux utilisateurs de modifier les widgets intégrés et de créer des widgets personnalisés, mais introduit également le risque de créer ou de mettre à jour des widgets de manière incorrecte. Pour garantir la création ou la mise à jour réussie du widget *Surveillance Web*, veuillez vous reporter au comportement des paramètres décrit dans les tableaux ci-dessous.

Paramètres

Les paramètres suivants sont pris en charge pour le widget de *supervision Web*.

Paramètre	type	name	value
Intervalle d'actualisation	0	rf_rate	0 - Aucune actualisation; 10 - 10 secondes; 30 - 30 secondes; 60 - *(par défaut)* 1 minute; 120 - 2 minutes; 600 - 10 minutes; 900 - 15 minutes.

Paramètre	type	name	value
Groupes d'hôtes	2	groupids.0	ID du groupe d'hôtes. Remarque : Pour configurer plusieurs groupes d'hôtes, créez un objet de champ de widget de tableau de bord pour chaque groupe d'hôtes avec un numéro incrémenté dans le nom de propriété. Ce paramètre n'est pas pris en charge lors de la configuration du widget sur un tableau de bord de modèle.
Groupes d'hôtes (widget)	1	groupids._reference	Au lieu de l'ID du groupe d'hôtes : `ABCDE._hostgroupids` - définir un widget compatible (avec son paramètre *Référence* défini sur "`ABCDE`") comme source de données pour les groupes d'hôtes. Ce paramètre n'est pas pris en charge lors de la configuration du widget sur un tableau de bord de modèle.
Exclure les groupes d'hôtes	2	exclude_groupids.0	ID du groupe d'hôtes. Remarque : Pour exclure plusieurs groupes d'hôtes, créez un objet de champ de widget de tableau de bord pour chaque groupe d'hôtes avec un numéro incrémenté dans le nom de propriété. Ce paramètre n'est pas pris en charge lors de la configuration du widget sur un tableau de bord de modèle.
Hôtes	3	hostids.0	ID de l'hôte. Remarque : Pour configurer plusieurs hôtes, créez un objet de champ de widget de tableau de bord pour chaque hôte avec un numéro incrémenté dans le nom de propriété. Pour plusieurs hôtes, le paramètre *Groupes d'hôtes* ne doit soit pas être configuré du tout, soit être configuré avec au moins un groupe d'hôtes auquel appartiennent les hôtes configurés. Ce paramètre n'est pas pris en charge lors de la configuration du widget sur un tableau de bord de modèle.
Hôtes (widget/tableau de bord)	1	hostids._reference	Au lieu de l'ID de l'hôte : `DASHBOARD.hostids` - définir le sélecteur d'hôte du tableau de bord comme source de données pour les hôtes ; `ABCDE._hostids` - définir un widget compatible (avec son paramètre *Référence* défini sur "`ABCDE`") comme source de données pour les hôtes. Ce paramètre n'est pas pris en charge lors de la configuration du widget sur un tableau de bord de modèle.
Balises de scénario			
Type d'évaluation	0	evaltype	0 - *(par défaut)* Et/Ou; 2 - Ou.
Nom de balise	1	tags.0.tag	Toute valeur de chaîne. Remarque : Le numéro dans le nom de propriété fait référence à l'ordre des balises dans la liste d'évaluation des balises. Comportement du paramètre : - *obligatoire* lors de la configuration de *Balises de scénario*

Paramètre	type	name	value
Opérateur	0	tags.0.operator	0 - Contient; 1 - Égal à; 2 - Ne contient pas; 3 - Différent de; 4 - Existe; 5 - N'existe pas. Remarque : Le numéro dans le nom de propriété fait référence à l'ordre des balises dans la liste d'évaluation des balises. Comportement du paramètre : - *obligatoire* lors de la configuration de *Balises de scénario*
Valeur de balise	1	tags.0.value	Toute valeur de chaîne. Remarque : Le numéro dans le nom de propriété fait référence à l'ordre des balises dans la liste d'évaluation des balises. Comportement du paramètre : - *obligatoire* lors de la configuration de *Balises de scénario*
Afficher les hôtes en maintenance	0	maintenance	0 - Désactivé; 1 - *(par défaut)* Activé.
Référence	1	reference	Toute valeur de chaîne composée de 5 caractères (par exemple, `ABCDE` ou `JBPNL`). Cette valeur doit être unique dans le tableau de bord auquel appartient le widget. Comportement du paramètre : - *obligatoire*

Exemples

Les exemples suivants ont pour objectif de décrire uniquement la configuration des objets champs du widget de tableau de bord pour le widget *Surveillance Web*. Pour plus d'informations sur la configuration d'un tableau de bord, consultez `dashboard.create`.

Configurer un widget *Surveillance Web*

Configurez un widget *Surveillance Web* qui affiche un résumé de l'état des scénarios de surveillance Web actifs pour le groupe d'hôtes « 4 ».

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "Mon tableau de bord",
        "display_period": 30,
        "auto_start": 1,
        "pages": [
            {
                "widgets": [
                    {
                        "type": "web",
                        "name": "Web monitoring",
                        "x": 0,
                        "o": 0,
                        "width": 18,
                        "height": 3,
```

```
                        "view_mode": 0,
                        "fields": [
                            {
                                "type 2,
                                "name": "groupids.0",
                                "valueuserGroups": 4
                            }
                        ]
                    }
                ]
            }
        ],
        "userGroups": [
            {
                "usrgrpid": 7,
                "permission": 2
            }
        ],
        "users": [
            {
                "userid": 1,
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Voir également

- Champ du widget du tableau de bord
- `dashboard.create`
- `dashboard.update`

Tableau de bord de modèle

Cette classe est conçue pour fonctionner avec les tableaux de bord de modèle.

Références d'objet :

- Tableau de bord de modèle
- Page de tableau de bord de modèle
  - Widget de tableau de bord de modèle
    - Champ de widget de tableau de bord de modèle

Méthodes disponibles :

- templatedashboard.create - créer de nouveaux tableaux de bord de modèle
- templatedashboard.delete - supprimer des tableaux de bord de modèle
- templatedashboard.get - récupérer des tableaux de bord de modèle
- templatedashboard.update - mettre à jour des tableaux de bord de modèle

Objet de tableau de bord de modèle

Les objets suivants sont directement liés à l'API `templatedashboard`.

Tableau de bord de modèle

L'objet tableau de bord de modèle possède les propriétés suivantes.

Property	Type	Description
dashboardid	ID	ID du tableau de bord de modèle. Comportement de la propriété: - *lecture seule* - *obligatoire* pour les opérations de mise à jour
name	string	Nom du tableau de bord de modèle. Comportement de la propriété: - *obligatoire* pour les opérations de création
templateid	ID	ID du modèle auquel appartient le tableau de bord. Comportement de la propriété: - *constante* - *obligatoire* pour les opérations de création
display_period	integer	Intervalle de diaporama par défaut (en secondes). Valeurs possibles: 10, 30, 60, 120, 600, 1800, 3600. Par défaut: 30.
auto_start	integer	Démarrage automatique du diaporama. Valeurs possibles: 0 - ne pas démarrer automatiquement le diaporama; 1 - *(par défaut)* démarrer automatiquement le diaporama.
uuid	string	Identifiant unique universel, utilisé pour lier les tableaux de bord de modèle importés à ceux déjà existants. Généré automatiquement s'il n'est pas fourni.

Page de tableau de bord du modèle

L'objet page de tableau de bord du modèle possède les propriétés suivantes.

Property	Type	Description
dashboard_pageid	ID	ID de la page de tableau de bord. Comportement de la propriété: - *lecture seule*
name	string	Nom de la page de tableau de bord. Par défaut: chaîne vide.
display_period	integer	Intervalle du diaporama du tableau de bord (en secondes). Valeurs possibles: 0, 10, 30, 60, 120, 600, 1800, 3600. Par défaut: 0 (utilisera l'intervalle de diaporama par défaut).
widgets	array	Tableau des objets widget de tableau de bord du modèle.

Widget de tableau de bord de modèle

L'objet widget de tableau de bord de modèle possède les propriétés suivantes.

Propriété	Type	Description
widgetid	ID	ID du widget de tableau de bord. Comportement de la propriété: - *lecture seule*
type	string	Type du widget de tableau de bord. Valeurs possibles : actionlog - Journal des actions ; clock - Horloge ; discovery - Statut de découverte ; favgraphs - Graphiques favoris ; favmaps - Cartes favorites ; gauge - Jauge ; graph - Graphique (classique) ; graphprototype - Prototype de graphique ; honeycomb - Nid d'abeilles ; hostavail - Disponibilité de l'hôte ; hostcard - Carte de l'hôte ; hostnavigator - Navigateur d'hôtes ; itemcard - Carte d'élément ; itemnavigator - Navigateur d'éléments ; item - Valeur d'élément ; map - Carte ; navtree - Arborescence de navigation des cartes ; piechart - Diagramme circulaire ; plaintext - Texte brut ; problemhosts - Hôtes avec problèmes ; problems - Problèmes ; problemsbysv - Problèmes par sévérité ; scatterplot - Nuage de points ; slareport - Rapport SLA ; svggraph - Graphique ; systeminfo - Informations système ; tophosts - Principaux hôtes ; topitems - Principaux éléments ; toptriggers - Principaux déclencheurs ; trigover - Vue d'ensemble des déclencheurs ; url - URL ; web - Supervision web. Comportement de la propriété: - *obligatoire*
name	string	Nom personnalisé du widget.
x	integer	Position horizontale à partir du côté gauche du tableau de bord. Les valeurs possibles vont de 0 à 71.
y	integer	Position verticale à partir du haut du tableau de bord. Les valeurs possibles vont de 0 à 63.
width	integer	Largeur du widget. Les valeurs possibles vont de 1 à 72.
height	integer	Hauteur du widget. Les valeurs possibles vont de 1 à 64.
view_mode	integer	Mode d'affichage du widget. Valeurs possibles : 0 - *(par défaut)* affichage par défaut du widget ; 1 - avec en-tête masqué ;
fields	array	Tableau d'objets champ de widget de tableau de bord de modèle.

Champ de widget de tableau de bord de modèle

L'objet champ de widget de tableau de bord de modèle possède les propriétés suivantes.

Propriété	Type	Description
type	integer	Type du champ de widget. Valeurs possibles : 0 - Entier ; 1 - Chaîne ; 4 - Élément ; 5 - Prototype d'élément ; 6 - Graphe ; 7 - Prototype de graphe ; 8 - Carte ; 9 - Service ; 10 - SLA ; 11 - Utilisateur ; 12 - Action ; 13 - Type de média. Comportement de la propriété : - *obligatoire*
name	string	Nom du champ de widget. Valeurs possibles : voir Champs de widget de tableau de bord. Notez que certains paramètres liés à l'hôte (par exemple, *Groupes d'hôtes*, *Exclure les groupes d'hôtes* et *Hôtes* dans le widget *Problèmes*, *Groupes d'hôtes* dans le widget *Disponibilité des hôtes*, etc.) ne sont pas disponibles lors de la configuration du widget sur un tableau de bord de modèle. Cela s'explique par le fait que les tableaux de bord de modèle affichent uniquement les données de l'hôte auquel le modèle est lié. Comportement de la propriété : - *obligatoire*
value	mixed	Valeur du champ de widget selon le type. Valeurs possibles : voir Champs de widget de tableau de bord. Notez que certains paramètres liés à l'hôte (par exemple, *Groupes d'hôtes*, *Exclure les groupes d'hôtes* et *Hôtes* dans le widget *Problèmes*, *Groupes d'hôtes* dans le widget *Disponibilité des hôtes*, etc.) ne sont pas disponibles lors de la configuration du widget sur un tableau de bord de modèle. Cela s'explique par le fait que les tableaux de bord de modèle affichent uniquement les données de l'hôte auquel le modèle est lié. Comportement de la propriété : - *obligatoire*

templatedashboard.create

Description

`object templatedashboard.create(object/array templateDashboards)`

Cette méthode permet de créer de nouveaux tableaux de bord de modèle.

> **Note:**
> Cette méthode est disponible uniquement pour les types d'utilisateur *Admin* et *Super admin*. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres du rôle utilisateur. Consultez Rôles d'utilisateur pour plus d'informations.

Paramètres

`(object/array)` Tableaux de bord de modèle à créer.

En plus des propriétés standard des tableaux de bord de modèle, la méthode accepte les paramètres suivants.

Paramètre	Type	Description
pages	array	Pages de tableau de bord de modèle à créer pour le tableau de bord. Les pages du tableau de bord seront ordonnées dans le même ordre que celui spécifié. Comportement du paramètre: - *obligatoire*

Valeurs retournées

`(object)` Retourne un objet contenant les ID des modèles de tableaux de bord créés dans la propriété `dashboardids`. L'ordre des ID retournés correspond à l'ordre des modèles de tableaux de bord transmis.

Exemples

Création d'un tableau de bord de modèle

Créez un tableau de bord de modèle nommé « Graphs » avec un widget Graph sur une seule page de tableau de bord.

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "templatedashboard.create",
    "params": {
        "templateid": "10318",
        "name": "Graphs",
        "pages": [
            {
                "widgets": [
                    {
                        "type": "graph",
                        "x": 0,
                        "y": 0,
                        "width": 12,
                        "height": 5,
                        "view_mode": 0,
                        "fields": [
                            {
                                "type": 6,
                                "name": "graphid",
                                "value": "1123"
                            }
                        ]
                    }
                ]

            }
        ]
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "32"
```

```
        ]
    },
    "id": 1
}
```

Voir aussi

- Page de tableau de bord du modèle
- Widget de tableau de bord du modèle
- Champ de widget de tableau de bord du modèle

Source

CTemplateDashboard::create() dans *ui/include/classes/api/services/CTemplateDashboard.php*.

templatedashboard.delete

Description

`object templatedashboard.delete(array templateDashboardIds)`

Cette méthode vous permet de supprimer des tableaux de bord de modèle.

> **Note:**
> Cette méthode est disponible uniquement pour les types d'utilisateur *Admin* et *Super admin*. Les autorisations d'appel de cette méthode peuvent être révoquées dans les paramètres des rôles utilisateur. Consultez Rôles utilisateur pour plus d'informations.

Paramètres

`(array)` Identifiants des tableaux de bord de modèle à supprimer.

Valeurs de retour

`(object)` Renvoie un objet contenant les ID des tableaux de bord de modèle supprimés dans la propriété `dashboardids`.

Exemples

Suppression de plusieurs tableaux de bord de modèle

Supprimez deux tableaux de bord de modèle.

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "templatedashboard.delete",
    "params": [
        "45",
        "46"
    ],
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "45",
            "46"
        ]
    },
    "id": 1
}
```

Source

CTemplateDashboard::delete() dans *ui/include/classes/api/services/CTemplateDashboard.php*.

templatedashboard.get

Description

`integer/array templatedashboard.get(object parameters)`

La méthode permet de récupérer les tableaux de bord de modèle selon les paramètres fournis.

> **Note:**
> Cette méthode est disponible pour les utilisateurs de tout type. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres des rôles utilisateur. Consultez Rôles utilisateur pour plus d'informations.

Paramètres

`(object)` Paramètres définissant la sortie souhaitée.

La méthode prend en charge les paramètres suivants.

Paramètre	Type	Description
dashboardids	ID/array	Renvoie uniquement les tableaux de bord de modèle avec les ID donnés.
templateids	ID/array	Renvoie uniquement les tableaux de bord de modèle qui appartiennent aux modèles donnés.
selectPages	query	Renvoie une propriété `pages` avec les pages du tableau de bord de modèle, correctement ordonnées.
sortfield	string/array	Trie le résultat selon les propriétés données. Valeurs possibles : `dashboardid`, `name`.
countOutput	boolean	Ces paramètres sont décrits dans les commentaires de référence.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Valeurs retournées

`(integer/array)` Retourne soit :

- un tableau d'objets ;
- le nombre d'objets récupérés, si le paramètre `countOutput` a été utilisé.

Exemples

Récupération des tableaux de bord de modèle

Récupérez tous les tableaux de bord de modèle avec widgets pour un modèle spécifié.

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "templatedashboard.get",
    "params": {
        "output": "extend",
        "selectPages": "extend",
        "templateids": "10001"
    },
    "id": 1
}
```

Réponse :


```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "dashboardid": "23",
            "name": "Docker overview",
            "templateid": "10001",
            "display_period": "30",
            "auto_start": "1",
            "uuid": "6dfcbe0bc5ad400ea9c1c2dd7649282f",
            "pages": [
                {
                    "dashboard_pageid": "1",
                    "name": "",
                    "display_period": "0",
                    "widgets": [
                        {
                            "widgetid": "220",
                            "type": "graph",
                            "name": "",
                            "x": "0",
                            "y": "0",
                            "width": "36",
                            "height": "5",
                            "view_mode": "0",
                            "fields": [
                                {
                                    "type": "6",
                                    "name": "graphid",
                                    "value": "1125"
                                }
                            ]
                        },
                        {
                            "widgetid": "221",
                            "type": "graph",
                            "name": "",
                            "x": "12",
                            "y": "0",
                            "width": "36",
                            "height": "5",
                            "view_mode": "0",
                            "fields": [
                                {
                                    "type": "6",
                                    "name": "graphid",
                                    "value": "1129"
                                }
                            ]
                        },
                        {
                            "widgetid": "222",
                            "type": "graph",
                            "name": "",
                            "x": "0",
                            "y": "5",
                            "width": "36",
                            "height": "5",
                            "view_mode": "0",
                            "fields": [
                                {
```

```
                                "type": "6",
                                "name": "graphid",
                                "value": "1128"
                            }
                        ]
                    },
                    {
                        "widgetid": "223",
                        "type": "graph",
                        "name": "",
                        "x": "12",
                        "y": "5",
                        "width": "36",
                        "height": "5",
                        "view_mode": "0",
                        "fields": [
                            {
                                "type": "6",
                                "name": "graphid",
                                "value": "1126"
                            }
                        ]
                    },
                    {
                        "widgetid": "224",
                        "type": "graph",
                        "name": "",
                        "x": "0",
                        "y": "10",
                        "width": "36",
                        "height": "5",
                        "view_mode": "0",
                        "fields": [
                            {
                                "type": "6",
                                "name": "graphid",
                                "value": "1127"
                            }
                        ]
                    }
                ]
            }
        ]
    }
],
"id": 1
}
```


Voir aussi

- Page de tableau de bord du modèle
- Widget de tableau de bord du modèle
- Champ de widget de tableau de bord du modèle

Source

CTemplateDashboard::get() dans *ui/include/classes/api/services/CTemplateDashboard.php*.

templatedashboard.update

Description

`object templatedashboard.update(object/array templateDashboards)`

Cette méthode permet de mettre à jour des tableaux de bord de modèle existants.

Note:
Cette méthode est disponible uniquement pour les types d'utilisateur *Admin* et *Super admin*. Les autorisations d'appel de cette méthode peuvent être révoquées dans les paramètres des rôles utilisateur. Consultez Rôles utilisateur pour plus d'informations.

Paramètres

`(object/array)` Propriétés du tableau de bord du modèle à mettre à jour.

La propriété `dashboardid` doit être spécifiée pour chaque tableau de bord, toutes les autres propriétés sont facultatives. Seules les propriétés spécifiées seront mises à jour.

En plus des propriétés standard du tableau de bord du modèle, la méthode accepte les paramètres suivants.

Paramètre	Type	Description
pages	array	Pages du tableau de bord du modèle pour remplacer les pages existantes du tableau de bord. Les pages du tableau de bord sont mises à jour par la propriété `dashboard_pageid`. De nouvelles pages de tableau de bord seront créées pour les objets sans propriété `dashboard_pageid` et les pages existantes du tableau de bord seront supprimées si elles ne sont pas réutilisées. Les pages du tableau de bord seront ordonnées dans le même ordre que celui spécifié. Seules les propriétés spécifiées des pages du tableau de bord seront mises à jour. Au moins un objet de page de tableau de bord est requis pour la propriété `pages`.

Valeurs retournées

`(object)` Retourne un objet contenant les identifiants des modèles de tableaux de bord mis à jour dans la propriété `dashboardids`.

Exemples

Renommer un tableau de bord de modèle

Renommez un tableau de bord de modèle en « Performance graphs ».

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "templatedashboard.update",
    "params": {
        "dashboardid": "23",
        "name": "Performance graphs"
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "23"
        ]
    },
    "id": 1
}
```

Mise à jour des pages du tableau de bord du modèle

Renommez la première page du tableau de bord, remplacez les widgets sur la deuxième page du tableau de bord et ajoutez une nouvelle page en troisième position. Supprimez toutes les autres pages du tableau de bord.

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "templatedashboard.update",
    "params": {
        "dashboardid": "2",
        "pages": [
            {
                "dashboard_pageid": 1,
                "name": "Page renommée"
            },
            {
                "dashboard_pageid": 2,
                "widgets": [
                    {
                        "type": "clock",
                        "x": 0,
                        "y": 0,
                        "width": 12,
                        "height": 3
                    }
                ]
            },
            {
                "display_period": 60
            }
        ]
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "2"
        ]
    },
    "id": 1
}
```

Voir aussi

- Widget de tableau de bord de modèle
- Champ de widget de tableau de bord de modèle

Source

CTemplateDashboard::update() dans *ui/include/classes/api/services/CTemplateDashboard.php*.

Tendance

Cette classe est conçue pour fonctionner avec les données de tendance.

Références d'objet :

- Tendance flottante
- Tendance entière

Méthodes disponibles :

- trend.get - récupérer les tendances

Objet de tendance

Les objets suivants sont directement liés à l'API `trend`.

Note:
Les objets de tendance diffèrent selon le type d'information de l'élément. Ils sont créés par le serveur Zabbix et ne peuvent pas être modifiés via l'API.

Tendance flottante

L'objet de tendance flottante possède les propriétés suivantes.

Propriété	Type	Description
clock	timestamp	Horodatage d'une heure pour laquelle la valeur a été calculée. Par exemple, l'horodatage « 04:00:00 » signifie que les valeurs ont été calculées pour la période « 04:00:00-04:59:59 ».
itemid	ID	ID de l'élément associé.
num	integer	Nombre de valeurs disponibles pour l'heure.
value_min	float	Valeur minimale horaire.
value_avg	float	Valeur moyenne horaire.
value_max	float	Valeur maximale horaire.

Tendance entière

L'objet de tendance entière possède les propriétés suivantes.

Propriété	Type	Description
clock	timestamp	Horodatage d'une heure pour laquelle la valeur a été calculée. Par exemple, l'horodatage de "04:00:00" signifie des valeurs calculées pour la période "04:00:00-04:59:59".
itemid	ID	ID de l'élément associé.
num	integer	Nombre de valeurs disponibles pour l'heure.
value_min	integer	Valeur minimale horaire.
value_avg	integer	Valeur moyenne horaire.
value_max	integer	Valeur maximale horaire.

trend.get

Description

`integer/array trend.get(object parameters)`

La méthode permet de récupérer les données de tendance selon les paramètres fournis.

Note:
Cette méthode est disponible pour les utilisateurs de tout type. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres du rôle utilisateur. Consultez Rôles utilisateur pour plus d'informations.

Paramètres

`(object)` Paramètres définissant la sortie souhaitée.

La méthode prend en charge les paramètres suivants.

Paramètre	Type	Description
itemids	ID/array	Retourner uniquement les tendances avec les ID d'élément indiqués.
time_from	timestamp	Retourner uniquement les valeurs qui ont été collectées après ou à l'heure indiquée.
time_till	timestamp	Retourner uniquement les valeurs qui ont été collectées avant ou à l'heure indiquée.
countOutput	boolean	Compter le nombre d'objets récupérés.
limit	integer	Limiter le nombre d'objets récupérés.
output	query	Définir les propriétés de l'objet Trend à retourner.

Valeurs retournées

`(integer/array)` Retourne soit :

- un tableau d'objets ;
- le nombre d'objets récupérés, si le paramètre `countOutput` a été utilisé.

Exemples

Récupération des données de tendance d'un élément

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "trend.get",
    "params": {
        "output": [
            "itemid",
            "clock",
            "num",
            "value_min",
            "value_avg",
            "value_max"
        ],
        "itemids": [
            "23715"
        ],
        "limit": "1"
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "itemid": "23715",
            "clock": "1446199200",
            "num": "60",
            "value_min": "0.165",
            "value_avg": "0.2168",
            "value_max": "0.35"
        }
    ],
    "id": 1
}
```

Source

CTrend::get() dans *ui/include/classes/api/services/CTrend.php*.

Token

Cette classe est conçue pour travailler avec les tokens.

Référentiel des objets :

- Token

Méthodes disponibles :

- token.create - créer de nouveaux tokens
- token.delete - supprimer des tokens
- token.get - récupérer des tokens
- token.update - mettre à jour des tokens
- token.generate - générer des tokens

Objet jeton

Les objets suivants sont directement liés à l'API `token`.

Jeton

L'objet jeton possède les propriétés suivantes.

Propriété	Type	Description
tokenid	ID	ID du jeton. Comportement de la propriété : - *lecture seule* - *obligatoire* pour les opérations de mise à jour
name	string	Nom du jeton. Comportement de la propriété : - *obligatoire* pour les opérations de création
description	text	Description du jeton.
userid	ID	ID de l'utilisateur auquel le jeton a été attribué. Par défaut : *utilisateur actuel*. Comportement de la propriété : - *constant*
lastaccess	timestamp	Date et heure les plus récentes auxquelles le jeton a été authentifié. "0" si le jeton n'a jamais été authentifié. Comportement de la propriété : - *lecture seule*
status	integer	Statut du jeton. Valeurs possibles : 0 - jeton activé *(par défaut)* ; 1 - jeton désactivé.
expires_at	timestamp	Date et heure d'expiration du jeton. "0" pour les jetons n'expirant jamais.
created_at	timestamp	Date et heure de création du jeton. Comportement de la propriété : - *lecture seule*
creator_userid	ID	ID de l'utilisateur qui a créé le jeton. Comportement de la propriété : - *lecture seule*

token.create

Description

`object token.create(object/array tokens)`

Cette méthode vous permet de créer de nouveaux jetons.

Note:
L'autorisation *Gérer les jetons API* est requise pour que le rôle utilisateur puisse gérer les jetons d'autres utilisateurs.

Attention:
Un jeton créé par cette méthode doit également être généré avant de pouvoir être utilisé.

Paramètres

`(object/array)` Jetons à créer.

La méthode accepte les jetons avec les propriétés de jeton standard.

Valeurs retournées

`(object)` Renvoie un objet contenant les ID des jetons créés sous la propriété `tokenids`. L'ordre des ID renvoyés correspond à l'ordre des jetons passés.

Exemples

Créer un jeton

Créez un jeton activé qui n'expire jamais et authentifie l'utilisateur d'ID 2.

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "token.create",
    "params": {
        "name": "Your token",
        "userid": "2"
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "tokenids": [
            "188"
        ]
    },
    "id": 1
}
```

Créez un jeton désactivé qui expire le 21 janvier 2021. Ce jeton authentifiera l'utilisateur actuel.

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "token.create",
    "params": {
        "name": "Your token",
        "status": "1",
        "expires_at": "1611238072"
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "tokenids": [
            "189"
        ]
    },
    "id": 1
}
```

Source

CToken::create() dans *ui/include/classes/api/services/CToken.php*.

token.delete

Description

`object token.delete(array tokenids)`

Cette méthode permet de supprimer des jetons.

> **Note:**
> L'autorisation *Gérer les jetons API* est requise pour que le rôle utilisateur puisse gérer les jetons d'autres utilisateurs.

Paramètres

`(array)` ID des jetons à supprimer.

Valeurs de retour

`(object)` Renvoie un objet contenant les ID des jetons supprimés dans la propriété `tokenids`.

Exemples

Supprimer plusieurs jetons

Supprimez deux jetons.

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "token.delete",
    "params": [
        "188",
        "192"
    ],
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "tokenids": [
            "188",
            "192"
        ]
    },
    "id": 1
}
```

Source

CToken::delete() dans *ui/include/classes/api/services/CToken.php*.

token.generate

Description

`object token.generate(array tokenids)`

Cette méthode vous permet de générer des jetons.

> **Note:**
> L'autorisation *Manage API tokens* est requise pour que le rôle utilisateur puisse gérer les jetons d'autres utilisateurs.

> **Attention:**
> Un jeton peut être généré par cette méthode uniquement s'il a été créé.

Paramètres

`(array)` ID des jetons à générer.

Valeurs de retour

`(array)` Renvoie un tableau d'objets contenant l'ID du jeton généré dans la propriété `tokenid` et la chaîne d'autorisation générée dans la propriété `token`.

Propriété	Type	Description
tokenid	ID	ID du jeton.
token	string	La chaîne d'autorisation générée pour ce jeton.

Exemples

Générer plusieurs jetons

Générez deux jetons.

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "token.generate",
    "params": [
        "1",
        "2"
    ],
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "tokenid": "1",
            "token": "bbcfce79a2d95037502f7e9a534906d3466c9a1484beb6ea0f4e7be28e8b8ce2"
        },
        {
            "tokenid": "2",
            "token": "fa1258a83d518eabd87698a96bd7f07e5a6ae8aeb8463cae33d50b91dd21bd6d"
        }
    ],
    "id": 1
}
```

Source

CToken::generate() dans *ui/include/classes/api/services/CToken.php*.

token.get

Description

`integer/array token.get(object parameters)`

La méthode permet de récupérer des jetons selon les paramètres fournis.

> **Note:**
> Seul le type d'utilisateur *Super admin* est autorisé à afficher les jetons d'autres utilisateurs.

Paramètres

`(object)` Paramètres définissant la sortie souhaitée.

La méthode prend en charge les paramètres suivants.

Paramètre	Type	Description
tokenids	ID/array	Retourner uniquement les tokens avec les ID donnés.
userids	ID/array	Retourner uniquement les tokens créés pour les utilisateurs donnés.
token	string	Retourner uniquement les tokens créés pour le *jeton d'authentification* donné.
valid_at	timestamp	Retourner uniquement les tokens qui sont valides (non expirés) à la date et à l'heure données.
expired_at	timestamp	Retourner uniquement les tokens qui sont expirés (non valides) à la date et à l'heure données.
sortfield	string/array	Trier le résultat selon les propriétés données. Valeurs possibles : `tokenid`, `name`, `lastaccess`, `status`, `expires_at`, `created_at`.
countOutput	boolean	Ces paramètres sont décrits dans le commentaire de référence.
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Valeurs retournées

`(integer/array)` Retourne soit :

- un tableau d'objets ;
- le nombre d'objets récupérés, si le paramètre `countOutput` a été utilisé.

Exemples

Récupérer un jeton

Récupérez toutes les données du jeton avec l'ID « 2 ».

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "token.get",
    "params": {
        "output": "extend",
        "tokenids": "2"
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "tokenid": "1",
            "name": "The Token",
            "description": "",
            "userid": "1",
            "lastaccess": "0",
            "status": "0",
            "expires_at": "1609406220",
            "created_at": "1611239454",
            "creator_userid": "1"
        }
```

```
    ],
    "id": 1
}
```

Source

CToken::get() dans *ui/include/classes/api/services/CToken.php*.

token.update

Description

`object token.update(object/array tokens)`

Cette méthode vous permet de mettre à jour des jetons existants.

> **Note:**
> L'autorisation *Gérer les jetons API* est requise pour que le rôle utilisateur puisse gérer les jetons d'autres utilisateurs.

Paramètres

`(object/array)` Propriétés du jeton à mettre à jour.

La propriété `tokenid` doit être définie pour chaque jeton, toutes les autres propriétés sont facultatives. Seules les propriétés transmises seront mises à jour, toutes les autres resteront inchangées.

La méthode accepte les jetons avec les propriétés de jeton standard.

Valeurs retournées

`(object)` Renvoie un objet contenant les ID des jetons mis à jour sous la propriété `tokenids`.

Exemples

Supprimer l'expiration du jeton

Supprimez la date d'expiration du jeton.

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "token.update",
    "params": {
        "tokenid": "2",
        "expires_at": "0"
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "tokenids": [
            "2"
        ]
    },
    "id": 1
}
```

Source

CToken::update() dans *ui/include/classes/api/services/CToken.php*.

Trace APM

Cette classe est conçue pour fonctionner avec les traces collectées pour la surveillance des performances des applications (APM).

Références aux objets :

- Trace APM
- Événement
- Lien

Méthodes disponibles :

- apmtrace.get - récupérer les traces

Objet de trace APM

Les objets suivants sont directement liés à l'API `apmtrace`.

Note:
Chaque propriété, à l'exception de `span_count` et `error_count`, correspond à une colonne de la table `otel_traces` de la source de données de télémétrie. La plupart des propriétés correspondent à la colonne portant le même nom (par exemple, la propriété `span_name` correspond à la colonne `SpanName`). Lorsque les noms diffèrent, les colonnes de la source de données sont indiquées dans les tableaux ci-dessous. Pour obtenir la liste des colonnes, consultez la page Colonnes de données de télémétrie prises en charge.

Trace APM

L'objet de trace APM représente une trace (le chemin qu'une requête suit à travers vos applications et services). Une trace se compose d'un ou plusieurs spans ; chaque span représente une opération unique.

L'objet de trace APM est basé sur le span racine de la trace (le span qui démarre la trace et qui n'a pas de span parent). Toutes les propriétés contiennent les données du span racine, à l'exception de `span_count` et `error_count`, qui sont calculées à partir de tous les spans de la trace.

L'objet de trace APM possède les propriétés suivantes :

Property	Type	Description
timestamp	integer	Heure Unix (en nanosecondes) à laquelle le span racine a démarré. Comportement de la propriété : - *lecture seule*
traceid	string	ID de la trace, partagé par tous les spans appartenant à la trace. Comportement de la propriété : - *lecture seule*
spanid	string	ID du span racine. Comportement de la propriété : - *lecture seule*
trace_state	string	Données supplémentaires, le cas échéant, que les outils de traçage associent à la trace (voir W3C Trace Context). Comportement de la propriété : - *lecture seule*
span_name	string	Nom de l'opération représentée par le span racine (par exemple, `POST /zabbix.php`). Comportement de la propriété : - *lecture seule*

Property	Type	Description
span_kind	string	Rôle du span racine. Valeurs possibles : `Unspecified` - le type de span n'est pas spécifié ; `Internal` - opération effectuée au sein d'une application ; `Server` - traitement d'une requête entrante ; `Client` - requête sortante ; `Producer` - création d'une opération asynchrone (par exemple, l'envoi d'un message) ; `Consumer` - traitement d'une opération asynchrone. Comportement de la propriété : - *lecture seule*
service_name	string	Nom du service ayant produit le span racine (par exemple, `zabbix-nginx`), extrait de l'attribut de ressource OpenTelemetry `service.name`. Comportement de la propriété : - *lecture seule*
resource_attributes	object	Attributs de ressource clé-valeur du span racine (par exemple, `host.name: eabf8c2a5ddc`). Comportement de la propriété : - *lecture seule*
scope_name	string	Nom de la portée d'instrumentation ayant produit le span racine. Comportement de la propriété : - *lecture seule*
scope_version	string	Version de la portée d'instrumentation. Comportement de la propriété : - *lecture seule*
span_attributes	object	Attributs du span clé-valeur du span racine (par exemple, `http.method: POST`). Comportement de la propriété : - *lecture seule*
duration	integer	Durée de la trace en nanosecondes. Comportement de la propriété : - *lecture seule*
status_code	string	État de l'opération représentée par le span racine. Valeurs possibles : `Unset` - aucun état explicite n'a été attribué au span racine ; `Ok` - l'opération s'est terminée avec succès ; `Error` - l'opération s'est terminée avec une erreur. Comportement de la propriété : - *lecture seule*
status_message	string	Informations supplémentaires, le cas échéant, sur l'état du span racine. Comportement de la propriété : - *lecture seule*
events	array	Tableau d'objets event du span racine, c'est-à-dire les moments importants survenus pendant l'opération (par exemple, des exceptions). Comportement de la propriété : - *lecture seule*

Property	Type	Description
links	array	Tableau d'objets link du span racine, qui pointent vers des spans associés dans la même trace ou dans une autre trace. Comportement de la propriété : - *lecture seule*
span_count	integer	Nombre total de spans dans la trace. Comportement de la propriété : - *lecture seule*
error_count	integer	Nombre de spans dans la trace dont `status_code` est défini sur `Error`. Comportement de la propriété : - *lecture seule*

Événement

L'objet événement possède les propriétés suivantes.

Property	Type	Description
timestamp	integer	Heure Unix (en nanosecondes) à laquelle l'événement s'est produit. Colonne de la source de données : `Events.Timestamp`
name	string	Nom de l'événement. Colonne de la source de données : `Events.Name`
attributes	object	Attributs clé-valeur contenant, le cas échéant, des informations supplémentaires sur l'événement. Colonne de la source de données : `Events.Attributes`

Lien

L'objet lien possède les propriétés suivantes.

Propriété	Type	Description
traceid	string	ID de la trace contenant le span lié. Colonne de la source de données : `Links.TraceId`
spanid	string	ID du span lié. Avec `traceid`, identifie de manière unique le span lié. Colonne de la source de données : `Links.SpanId`
trace_state	string	État de la trace, le cas échéant, du span lié (voir W3C Trace Context). Colonne de la source de données : `Links.TraceState`
attributes	object	Attributs clé-valeur contenant des informations supplémentaires sur le lien. Colonne de la source de données : `Links.Attributes`

apmtrace.get

Description

`integer/array apmtrace.get(object parameters)`

Cette méthode vous permet de récupérer des traces selon les paramètres indiqués.

> **Note:**
> Cette méthode est disponible pour les utilisateurs de tout type. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres des rôles utilisateur. Consultez la page Rôles utilisateur pour plus d'informations.

Paramètres

`(object)` Paramètres définissant le résultat souhaité.

La méthode prend en charge les paramètres suivants.

Paramètre	Type	Description
time_from	timestamp	Retourne uniquement les traces dont le span racine a commencé à l'heure indiquée ou après. Comportement du paramètre : - *obligatoire*
time_till	timestamp	Retourne uniquement les traces dont le span racine a commencé avant l'heure indiquée. Comportement du paramètre : - *obligatoire*
traceids	string/array	Retourne uniquement les traces correspondant aux ID indiqués.
resource_attributes	object/array	Retourne uniquement les traces dont le span racine possède les attributs de ressource indiqués. Format : `[{"key": "<key>", "value": "<value>", "operator": "<operator>"}, ...]`. Un tableau vide retourne toutes les traces. Valeurs possibles de l'opérateur : 0 - *(par défaut)* Contient ; 1 - Est égal à ; 2 - Ne contient pas ; 3 - N'est pas égal à ; 4 - Existe ; 5 - N'existe pas.
resource_attributes_evaltype	integer	Méthode d'évaluation des conditions dans `resource_attributes`. Valeurs possibles : 0 - *(par défaut)* Et/Ou ; 2 - Ou.
span_attributes	object/array	Retourne uniquement les traces dont le span racine possède les attributs de span indiqués. Format : `[{"key": "<key>", "value": "<value>", "operator": "<operator>"}, ...]`. Un tableau vide retourne toutes les traces. Valeurs possibles de l'opérateur : 0 - *(par défaut)* Contient ; 1 - Est égal à ; 2 - Ne contient pas ; 3 - N'est pas égal à ; 4 - Existe ; 5 - N'existe pas.
span_attributes_evaltype	integer	Méthode d'évaluation des conditions dans `span_attributes`. Valeurs possibles : 0 - *(par défaut)* Et/Ou ; 2 - Ou.
min_duration	integer	Retourne uniquement les traces dont la `duration` du span racine est supérieure ou égale à la valeur indiquée, en nanosecondes.
max_duration	integer	Retourne uniquement les traces dont la `duration` du span racine est inférieure à la valeur indiquée, en nanosecondes.

Paramètre	Type	Description
filter	object	Retourne uniquement les résultats correspondant exactement au filtre indiqué. Accepte un objet dont les clés sont des noms de propriétés et dont les valeurs sont soit une valeur unique, soit un tableau de valeurs à comparer. Propriétés prises en charge : `traceid`, `spanid`, `trace_state`, `span_name`, `span_kind`, `service_name`, `scope_name`, `scope_version`, `status_code`, `status_message`.
search	object	Retourne les résultats correspondant au modèle indiqué (sans distinction entre majuscules et minuscules). Accepte un objet dont les clés sont des noms de propriétés et dont les valeurs sont des chaînes à rechercher. Si aucune option supplémentaire n'est indiquée, une recherche `LIKE "%…%"` est effectuée. Propriétés prises en charge : `trace_state`, `span_name`, `service_name`, `scope_name`, `scope_version`, `status_message`.
sortfield	string/array	Trie le résultat selon les propriétés indiquées. Valeurs possibles : `timestamp`, `traceid`, `spanid`, `trace_state`, `span_name`, `span_kind`, `service_name`, `scope_name`, `scope_version`, `duration`, `status_code`, `status_message`.
offset	integer	Nombre d'enregistrements à ignorer avant de retourner le résultat.
countOutput	boolean	Ces paramètres sont décrits dans le commentaire de référence.
excludeSearch	boolean	
limit	integer	
output	query	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Valeurs de retour

`(integer/array)` Renvoie l'un des éléments suivants :

- un tableau d'objets ;
- le nombre d'objets récupérés, si le paramètre `countOutput` a été utilisé.

Exemples

Récupération des traces lentes

Récupérez les deux traces les plus lentes du service `zabbix-nginx`. Recherchez les traces entre 10:00 et 11:00 le 1er octobre 2026 (UTC), et incluez uniquement celles dont la durée est d'au moins 1 seconde.

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "apmtrace.get",
    "params": {
        "output": ["timestamp", "traceid", "span_name", "service_name", "duration", "status_code", "span_c
        "time_from": 1790848800,
        "time_till": 1790852400,
        "filter": {
            "service_name": "zabbix-nginx"
        },
        "min_duration": 1000000000,
        "sortfield": "duration",
        "sortorder": "DESC",
```

```
        "limit": 2
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "timestamp": "1790849274910226417",
            "traceid": "8a4f1c7e2b9d4e6fa0c3b5d7e9f1a2c4",
            "span_name": "POST /zabbix.php",
            "service_name": "zabbix-nginx",
            "duration": "30412907815",
            "status_code": "Error",
            "span_count": "49",
            "error_count": "2"
        },
        {
            "timestamp": "1790850512338104952",
            "traceid": "dee625f33cb0994e86c5cd4640e3b09c",
            "span_name": "POST /zabbix.php",
            "service_name": "zabbix-nginx",
            "duration": "1104826531",
            "status_code": "Unset",
            "span_count": "27",
            "error_count": "0"
        }
    ],
    "id": 1
}
```

Voir aussi

- Span APM
- Métrique APM
- Journal APM

Source

CApmTrace::get() dans *ui/include/classes/api/services/CApmTrace.php*.

Type de média

Cette classe est conçue pour fonctionner avec les types de média.

Références d'objet :

- Type de média
  - Paramètres de webhook
  - Paramètres de script
- Modèle de message

Méthodes disponibles :

- mediatype.create - créer de nouveaux types de média
- mediatype.delete - supprimer des types de média
- mediatype.get - récupérer des types de média
- mediatype.update - mettre à jour des types de média

Objet type de média

Les objets suivants sont directement liés à l'API `mediatype`.

Type de média

L'objet de type de média possède les propriétés suivantes.

Propriété	Type	Description
mediatypeid	ID	ID du type de média. Comportement de la propriété : - *lecture seule* - *obligatoire* pour les opérations de mise à jour
name	string	Nom du type de média. Comportement de la propriété : - *obligatoire* pour les opérations de création
type	integer	Transport utilisé par le type de média. Valeurs possibles : 0 - E-mail ; 1 - Script ; 2 - SMS ; 4 - Webhook ; 5 - Push. Comportement de la propriété : - *obligatoire* pour les opérations de création
exec_path	string	Nom du fichier de script (par exemple, notification.sh) situé dans le répertoire spécifié dans le paramètre de configuration du serveur AlertScriptsPath. Comportement de la propriété : - *obligatoire* si `type` est défini sur « Script »
gsm_modem	string	Nom du périphérique série du modem GSM. Comportement de la propriété : - *obligatoire* si `type` est défini sur « SMS »
passwd	string	Mot de passe d'authentification. Comportement de la propriété : - *pris en charge* si `smtp_authentication` est défini sur « Normal password »
provider	integer	Fournisseur de messagerie. Valeurs possibles : 0 - *(par défaut)* SMTP générique ; 1 - Gmail ; 2 - Relais Gmail ; 3 - Office365 ; 4 - Relais Office365.
smtp_email	string	Adresse e-mail depuis laquelle les notifications seront envoyées. Comportement de la propriété : - *obligatoire* si `type` est défini sur « Email »
smtp_helo	string	HELO SMTP. Comportement de la propriété : - *pris en charge* si `type` est défini sur « Email »
smtp_server	string	Serveur SMTP. Comportement de la propriété : - *obligatoire* si `type` est défini sur « Email »

Propriété	Type	Description
smtp_port	integer	Port du serveur SMTP auquel se connecter. Valeur par défaut : 25. Comportement de la propriété : - *pris en charge* si `type` est défini sur « Email »
smtp_security	integer	Niveau de sécurité de la connexion SMTP à utiliser. Valeurs possibles : 0 - *(par défaut)* Aucun ; 1 - STARTTLS ; 2 - SSL/TLS. Comportement de la propriété : - *pris en charge* si `type` est défini sur « Email »
smtp_verify_host	integer	Vérification SSL de l'hôte pour SMTP. Valeurs possibles : 0 - *(par défaut)* Non ; 1 - Oui. Comportement de la propriété : - *pris en charge* si `smtp_security` est défini sur « STARTTLS » ou « SSL/TLS »
smtp_verify_peer	integer	Vérification SSL du pair pour SMTP. Valeurs possibles : 0 - *(par défaut)* Non ; 1 - Oui. Comportement de la propriété : - *pris en charge* si `smtp_security` est défini sur « STARTTLS » ou « SSL/TLS »
smtp_authentication	integer	Méthode d'authentification SMTP à utiliser. Valeurs possibles : 0 - *(par défaut)* Aucune ; 1 - Mot de passe normal ; 2 - Jeton OAuth. L'authentification OAuth n'est pas autorisée pour le fournisseur de messagerie *Relais Office365*. Comportement de la propriété : - *pris en charge* si `type` est défini sur « Email »
redirection_url	string	URL de l'interface Zabbix vers laquelle rediriger après l'autorisation OAuth. Valeur par défaut : Valeur de la propriété `url` des paramètres de l'API avec la partie `zabbix.php?action=oauth.authorize` Comportement de la propriété : - *obligatoire* si `smtp_authentication` est défini sur « OAuth token »
client_id	string	Identifiant client enregistré auprès du serveur d'autorisation OAuth. Comportement de la propriété : - *obligatoire* si `smtp_authentication` est défini sur « OAuth token »
client_secret	string	Secret client enregistré auprès du serveur d'autorisation OAuth. Accessible uniquement à un utilisateur de type Super administrateur. Comportement de la propriété : - *obligatoire* si `smtp_authentication` est défini sur « OAuth token »

Propriété	Type	Description
authorization_url	string	URL OAuth, avec les paramètres, permettant d'obtenir les jetons d'accès et d'actualisation. Comportement de la propriété : - *obligatoire* si `smtp_authentication` est défini sur « OAuth token »
token_url	string	URL OAuth permettant d'échanger le jeton d'autorisation contre des jetons d'accès et d'actualisation. Cette URL est également utilisée par le serveur pour actualiser un jeton d'accès invalide. Comportement de la propriété : - *obligatoire* si `smtp_authentication` est défini sur « OAuth token »
tokens_status	integer	Masque de bits indiquant l'état des jetons. Valeurs possibles : 0 - (par défaut) Les deux jetons contiennent une valeur non valide ; 1 - Le jeton d'accès contient une valeur valide ; 2 - Le jeton d'actualisation contient une valeur valide ; 3 - Les deux jetons contiennent une valeur valide. Comportement de la propriété : - *pris en charge* si `smtp_authentication` est défini sur « OAuth token »
access_token	string	Valeur du jeton d'accès OAuth. Comportement de la propriété : - *obligatoire* si `smtp_authentication` est défini sur « OAuth token »
access_token_updated	timestamp	Horodatage de la dernière modification de `access_token` effectuée par le serveur lors de l'actualisation avec `refresh_token` ou par l'API lors de modifications du jeton. Comportement de la propriété : - *pris en charge* si `smtp_authentication` est défini sur « OAuth token »
access_expires_in	integer	Durée, en secondes, avant que `access_token` n'expire et qu'une requête vers `refresh_url` soit nécessaire. Cette valeur est définie par le serveur Zabbix lors de l'actualisation de `access_token` ou par l'API lors de modifications du jeton. L'horodatage est calculé en ajoutant la valeur de `access_token_updated`. Comportement de la propriété : - *pris en charge* si `smtp_authentication` est défini sur « OAuth token »
refresh_token	string	Valeur du jeton d'actualisation OAuth. Comportement de la propriété : - *obligatoire* si `smtp_authentication` est défini sur « OAuth token »
status	integer	Indique si le type de média est activé. Valeurs possibles : 0 - *(par défaut)* Activé ; 1 - Désactivé.
username	string	Nom d'utilisateur. Comportement de la propriété : - *pris en charge* si `smtp_authentication` est défini sur « Normal password »

Propriété	Type	Description
maxsessions	integer	Nombre maximal d'alertes pouvant être traitées en parallèle. Valeurs possibles si `type` est défini sur « SMS » : 1. Valeurs possibles si `type` est défini sur « Email », « Script » ou « Webhook » : 0-100. Valeur par défaut : 1.
maxattempts	integer	Nombre maximal de tentatives d'envoi d'une alerte. Valeurs possibles : 1-100. Valeur par défaut : 3.
attempt_interval	string	Intervalle entre les tentatives successives. Accepte un nombre de secondes (par exemple, 1, 30, 60) ou une valeur avec un suffixe temporel (par exemple, 30s, 1m, 1h). Valeurs possibles : 0-1h. Valeur par défaut : 10s.
message_format	integer	Format du message. Valeurs possibles : 0 - Texte brut ; 1 - *(par défaut)* HTML. Comportement de la propriété : - *pris en charge* si `type` est défini sur « Email »
script	text	Corps du script du webhook (JavaScript). Comportement de la propriété : - *obligatoire* si `type` est défini sur « Webhook »
timeout	string	Délai d'expiration du script du webhook. Accepte un nombre de secondes (par exemple, 1, 60) ou une valeur avec un suffixe temporel (par exemple, 30s, 1m). Valeurs possibles : 1-60s. Valeur par défaut : 30s.
process_tags	integer	Comportement de la propriété : - *pris en charge* si `type` est défini sur « Webhook » Traite les valeurs des propriétés JSON de la réponse du script Webhook comme des tags. Ces tags sont ajoutés aux tags de problème existants. Valeurs possibles : 0 - *(par défaut)* Ignorer la réponse du script webhook ; 1 - Traiter la réponse du script webhook comme des tags. Comportement de la propriété : - *pris en charge* si `type` est défini sur « Webhook »

Propriété	Type	Description
show_event_menu	integer	Inclut une entrée dans le menu des événements qui renvoie vers une URL personnalisée. Ajoute également la propriété `urls` à la sortie de problem.get et event.get. Valeurs possibles : 0 - *(par défaut)* Ne pas inclure l'entrée du menu des événements ni la propriété `urls` ; 1 - Inclure l'entrée du menu des événements et la propriété `urls`.
event_menu_url	string	Comportement de la propriété : - *pris en charge* si `type` est défini sur « Webhook » URL utilisée dans l'entrée du menu des événements et dans la propriété `urls` renvoyée par problem.get et event.get.
event_menu_name	string	Comportement de la propriété : - *pris en charge* si `show_event_menu` est défini sur « Include event menu entry and `urls` property » Nom utilisé pour l'entrée du menu des événements et dans la propriété `urls` renvoyée par problem.get et event.get.
parameters	array	Comportement de la propriété : - *pris en charge* si `show_event_menu` est défini sur « Include event menu entry and `urls` property » Paramètres du Webhook ou du script.
description	text	Comportement de la propriété : - *pris en charge* si `type` est défini sur « Webhook » ou « Script » Description du type de média.

Paramètres du webhook

Les paramètres du webhook ont les propriétés suivantes.

Propriété	Type	Description
name	string	Nom du paramètre. Comportement de la propriété : - *obligatoire*
value	string	Valeur du paramètre, prend en charge les macros. Les macros prises en charge sont décrites sur la page Macros prises en charge.

Paramètres du script

Les paramètres du script ont les propriétés suivantes.

Propriété	Type	Description
sortorder	integer	L'ordre dans lequel les valeurs des paramètres seront transmises au script en tant qu'arguments de ligne de commande, en commençant par 0 comme premier argument. Comportement de la propriété : - *obligatoire*
value	string	Valeur du paramètre, prend en charge les macros. Les macros prises en charge sont décrites sur la page Macros prises en charge.

Modèle de message

L'objet modèle de message définit un modèle qui sera utilisé comme message par défaut pour les opérations d'action afin d'envoyer une notification. Il possède les propriétés suivantes.

Propriété	Type	Description
eventsource	integer	Source de l'événement. Valeurs possibles : 0 - Déclencheurs ; 1 - Découverte ; 2 - Enregistrement automatique ; 3 - Interne ; 4 - Services. Comportement de la propriété : - *obligatoire*
recovery	integer	Mode de fonctionnement. Valeurs possibles : 0 - Opérations ; 1 - Opérations de récupération ; 2 - Opérations de mise à jour. Comportement de la propriété : - *obligatoire*
subject	string	Sujet du message.
message	string	Texte du message.

mediatype.create

Description

`object mediatype.create(object/array mediaTypes)`

Cette méthode vous permet de créer de nouveaux types de média.

Note:
Cette méthode est uniquement disponible pour le type d'utilisateur *Super administrateur*. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres des rôles utilisateur. Consultez la page Rôles utilisateur pour plus d'informations.
 Si des méthodes de livraison sont spécifiées dans l'indicateur de fonctionnalité `$ZBX_FEATURE_FLAGS['media_type_denylist']` du fichier de configuration de l'interface, l'utilisateur n'est pas autorisé à appeler la méthode pour les types de média correspondants.

Paramètres

`(object/array)` Types de média à créer.

En plus des propriétés standard des types de média, la méthode accepte les paramètres suivants.

Paramètre	Type	Description
message_templates	array	Modèles de message à créer pour le type de média.

Valeurs retournées

`(object)` Retourne un objet contenant les identifiants des types de média créées sous la propriété `mediatypeids`. L'ordre des ID retournés correspond à l'ordre des types de média transmis.

Exemples

Création d'un type de média e-mail

Créez un nouveau type de média e-mail avec un port SMTP personnalisé et des modèles de message.

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "mediatype.create",
    "params": {
        "type": "0",
        "name": "Email",
        "smtp_server": "mail.example.com",
        "smtp_helo": "example.com",
        "smtp_email": "zabbix@example.com",
        "smtp_port": "587",
        "message_format": "1",
        "message_templates": [
            {
                "eventsource": "0",
                "recovery": "0",
                "subject": "Problème : {EVENT.NAME}",
                "message": "Le problème \"{EVENT.NAME}\" sur l'hôte \"{HOST.NAME}\" a commencé à {EVENT.TIN
            },
            {
                "eventsource": "0",
                "recovery": "1",
                "subject": "Résolu en {EVENT.DURATION} : {EVENT.NAME}",
                "message": "Le problème \"{EVENT.NAME}\" sur l'hôte \"{HOST.NAME}\" a été résolu à {EVENT.I
            },
            {
                "eventsource": "0",
                "recovery": "2",
                "subject": "Problème mis à jour dans {EVENT.AGE} : {EVENT.NAME}",
                "message": "{USER.FULLNAME} a {EVENT.UPDATE.ACTION} le problème \"{EVENT.NAME}\" sur l'hôte
            }
        ]
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "mediatypeids": [
            "7"
        ]
    },
    "id": 1
}
```

Création d'un type de média de script

Créez un nouveau type de média de script avec une valeur personnalisée pour le nombre de tentatives et l'intervalle entre celles-ci.

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "mediatype.create",
    "params": {
        "type": "1",
        "name": "Push notifications",
        "exec_path": "push-notification.sh",
        "maxattempts": "5",
        "attempt_interval": "11s",
        "parameters": [
            {
                "sortorder": "0",
```

```
                "value": "{ALERT.SENDTO}"
            },
            {
                "sortorder": "1",
                "value": "{ALERT.SUBJECT}"
            },
            {
                "sortorder": "2",
                "value": "{ALERT.MESSAGE}"
            }
        ]
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "mediatypeids": [
            "8"
        ]
    },
    "id": 1
}
```

Création d'un type de média webhook

Créez un nouveau type de média webhook.

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "mediatype.create",
    "params": {
        "type": "4",
        "name": "Webhook",
        "script": "var Webhook = {\r\n    token: null,\r\n    to: null,\r\n    subject: null,\r\n    messa
        "parameters": [
            {
                "name": "Message",
                "value": "{ALERT.MESSAGE}"
            },
            {
                "name": "Subject",
                "value": "{ALERT.SUBJECT}"
            },
            {
                "name": "To",
                "value": "{ALERT.SENDTO}"
            },
            {
                "name": "Token",
                "value": "<Token>"
            }
        ]
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
```

```
    "result": {
        "mediatypeids": [
            "9"
        ]
    },
    "id": 1
}
```

Source

CMediaType::create() dans *ui/include/classes/api/services/CMediaType.php*.

mediatype.delete

Description

`object mediatype.delete(array mediaTypeIds)`

Cette méthode vous permet de supprimer des types de média.

> **Note:**
> Cette méthode est uniquement disponible pour le type d'utilisateur *Super administrateur*. Les autorisations permettant d'appeler la méthode peuvent être révoquées dans les paramètres des rôles utilisateur. Consultez la page Rôles utilisateur pour plus d'informations.
 Si des méthodes de livraison sont spécifiées dans l'indicateur de fonctionnalité `$ZBX_FEATURE_FLAGS['media_type_denylist']` du fichier de configuration de l'interface, l'utilisateur n'est pas autorisé à appeler la méthode pour les types de média correspondants.

Paramètres

`(array)` ID des types de média à supprimer.

Valeurs de retour

`(object)` Renvoie un objet contenant les ID des types de média supprimés dans la propriété `mediatypeids`.

Exemples

Suppression de plusieurs types de média

Supprimez deux types de média.

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "mediatype.delete",
    "params": [
        "3",
        "5"
    ],
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "mediatypeids": [
            "3",
            "5"
        ]
    },
    "id": 1
}
```

Source

CMediaType::delete() dans *ui/include/classes/api/services/CMediaType.php*.

mediatype.get

Description

`integer/array mediatype.get(object parameters)`

Cette méthode permet de récupérer les types de média en fonction des paramètres indiqués.

Note:
Cette méthode est disponible pour les utilisateurs de tout type. Les autorisations d'appeler la méthode peuvent être révoquées dans les paramètres du rôle utilisateur. Consultez la page Rôles utilisateur pour plus d'informations.
 Si des méthodes de livraison sont spécifiées dans l'indicateur de fonctionnalité `$ZBX_FEATURE_FLAGS['media_type_denylist']` du fichier de configuration de l'interface, l'utilisateur n'est pas autorisé à appeler la méthode pour les types de média correspondants.

Paramètres

`(object)` Paramètres définissant la sortie souhaitée.

La méthode prend en charge les paramètres suivants.

Note:
Lors de la demande d'informations liées à l'utilisateur sur les types de média, les utilisateurs de type *Admin* et *User* ne peuvent récupérer des données que sur leur propre utilisateur. Pour un exemple, voir Récupération des types de média en tant qu'*Admin*.

Paramètre	Type	Description
mediatypeids	ID/array	Retourner uniquement les types de média avec les ID donnés.
mediaids	ID/array	Retourner uniquement les types de média utilisés par les media donnés.
userids	ID/array	Retourner uniquement les types de média utilisés par les utilisateurs donnés.
selectActions	query	Retourner une propriété `actions` avec les actions qui utilisent le type de média.
selectMessageTemplates	query	Retourner une propriété `message_templates` avec un tableau de messages de type de média. Comportement des paramètres : - *pris en charge* pour les utilisateurs de type *Super admin*
selectUsers	query	Retourner une propriété `users` avec les utilisateurs qui utilisent le type de média. Voir `user.get` pour les restrictions selon le type d'utilisateur.
sortfield	string/array	Trier le résultat selon les propriétés données. Valeurs possibles : `mediatypeid`.
filter	object	Retourner uniquement les résultats qui correspondent exactement au filtre donné. Accepte un objet, où les clés sont des noms de propriété et les valeurs sont soit une valeur unique, soit un tableau de valeurs à comparer. Ne prend pas en charge les propriétés du type de données `text`. Propriétés possibles de l'objet type de média pour les utilisateurs de type *Admin* et *User* : `mediatypeid`, `name`, `type`, `status`, `maxattempts`.

Paramètre	Type	Description
output	query	Propriétés de l'objet type de média à retourner. Les utilisateurs de type *Admin* et *User* ne peuvent récupérer que les propriétés suivantes : `mediatypeid`, `name`, `type`, `status`, `maxattempts`, `description`. Pour un exemple, voir Récupération des types de média en tant qu'*Admin*. Par défaut : `extend`.
search	object	Retourner les résultats correspondant au motif donné (insensible à la casse). Accepte un objet, où les clés sont des noms de propriété et les valeurs sont des chaînes à rechercher. Si aucune option supplémentaire n'est fournie, cela effectuera une recherche `LIKE "%…%"`. Prend en charge uniquement les propriétés des types de données `string` et `text`.
		Propriétés possibles de l'objet type de média pour les utilisateurs de type *Admin* et *User* : `name`, `description`.
countOutput	boolean	Ces paramètres sont décrits dans la référence commentée.
editable	boolean	
excludeSearch	boolean	
limit	integer	
preservekeys	boolean	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Valeurs retournées

`(integer/array)` Retourne soit :

- un tableau d'objets ;
- le nombre d'objets récupérés, si le paramètre `countOutput` a été utilisé.

Exemples

Récupération des types de média

Récupérez tous les types de média configurés. L'exemple suivant renvoie deux types de média :

- type de média e-mail ;
- type de média SMS.

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "mediatype.get",
    "params": {
        "output": "extend",
        "selectMessageTemplates": "extend"
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "mediatypeid": "1",
```

```
        "type": "0",
        "name": "Email",
        "smtp_server": "mail.example.com",
        "smtp_helo": "example.com",
        "smtp_email": "zabbix@example.com",
        "exec_path": "",
        "gsm_modem": "",
        "username": "",
        "passwd": "",
        "status": "0",
        "smtp_port": "25",
        "smtp_security": "0",
        "smtp_verify_peer": "0",
        "smtp_verify_host": "0",
        "smtp_authentication": "0",
        "maxsessions": "1",
        "maxattempts": "3",
        "attempt_interval": "10s",
        "message_format": "0",
        "script": "",
        "timeout": "30s",
        "process_tags": "0",
        "show_event_menu": "1",
        "event_menu_url": "",
        "event_menu_name": "",
        "description": "",
        "provider": "0",
        "message_templates": [
            {
                "eventsource": "0",
                "recovery": "0",
                "subject": "Problème : {EVENT.NAME}",
                "message": "Le problème a commencé à {EVENT.TIME} le {EVENT.DATE}\r\nNom du problème :
            },
            {
                "eventsource": "0",
                "recovery": "1",
                "subject": "Résolu : {EVENT.NAME}",
                "message": "Le problème a été résolu à {EVENT.RECOVERY.TIME} le {EVENT.RECOVERY.DATE}\
            },
            {
                "eventsource": "0",
                "recovery": "2",
                "subject": "Problème mis à jour : {EVENT.NAME}",
                "message": "{USER.FULLNAME} a effectué l'action {EVENT.UPDATE.ACTION} sur le problème
            },
            {
                "eventsource": "1",
                "recovery": "0",
                "subject": "Découverte : {DISCOVERY.DEVICE.STATUS} {DISCOVERY.DEVICE.IPADDRESS}",
                "message": "Règle de découverte : {DISCOVERY.RULE.NAME}\r\n\r\nIP du périphérique : {D
            },
            {
                "eventsource": "2",
                "recovery": "0",
                "subject": "Auto-enregistrement : {HOST.HOST}",
                "message": "Nom de l'hôte : {HOST.HOST}\r\nIP de l'hôte : {HOST.IP}\r\nPort de l'agent
            }
        ],
        "parameters": []
    },
    {
```

```
        "mediatypeid": "3",
        "type": "2",
        "name": "SMS",
        "smtp_server": "",
        "smtp_helo": "",
        "smtp_email": "",
        "exec_path": "",
        "gsm_modem": "/dev/ttyS0",
        "username": "",
        "passwd": "",
        "status": "0",
        "smtp_port": "25",
        "smtp_security": "0",
        "smtp_verify_peer": "0",
        "smtp_verify_host": "0",
        "smtp_authentication": "0",
        "maxsessions": "1",
        "maxattempts": "3",
        "attempt_interval": "10s",
        "message_format": "1",
        "script": "",
        "timeout": "30s",
        "process_tags": "0",
        "show_event_menu": "1",
        "event_menu_url": "",
        "event_menu_name": "",
        "description": "",
        "provider": "0",
        "message_templates": [
            {
                "eventsource": "0",
                "recovery": "0",
                "subject": "",
                "message": "{EVENT.SEVERITY} : {EVENT.NAME}\r\nHôte : {HOST.NAME}\r\n{EVENT.DATE} {EVE
            },
            {
                "eventsource": "0",
                "recovery": "1",
                "subject": "",
                "message": "RÉSOLU : {EVENT.NAME}\r\nHôte : {HOST.NAME}\r\n{EVENT.DATE} {EVENT.TIME}"
            },
            {
                "eventsource": "0",
                "recovery": "2",
                "subject": "",
                "message": "{USER.FULLNAME} a effectué l'action {EVENT.UPDATE.ACTION} sur le problème
            },
            {
                "eventsource": "1",
                "recovery": "0",
                "subject": "",
                "message": "Découverte : {DISCOVERY.DEVICE.STATUS} {DISCOVERY.DEVICE.IPADDRESS}"
            },
            {
                "eventsource": "2",
                "recovery": "0",
                "subject": "",
                "message": "Auto-enregistrement : {HOST.HOST}\r\nIP de l'hôte : {HOST.IP}\r\nPort de l
            }
        ],
        "parameters": []
    }
```

```
    ],
    "id": 1
}
```

Récupération des types de média en tant qu'*Admin*

En tant qu'utilisateur de type *Admin*, récupérez tous les types de média activés, avec les utilisateurs qui utilisent ces types de média. L'exemple suivant renvoie deux types de média :

- le type de média email avec un utilisateur (uniquement l'utilisateur du type *Admin* lui-même) ;
- le type de média SMS sans utilisateur.

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "mediatype.get",
    "params": {
        "output": "extend",
        "filter": {
            "status": 0
        },
        "selectUsers": "extend"
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "mediatypeid": "1",
            "type": "0",
            "name": "Email",
            "status": "0",
            "description": "",
            "maxattempts": "3",
            "users": [
                {
                    "userid": "3",
                    "username": "database-admin",
                    "name": "John",
                    "surname": "Doe",
                    "url": "",
                    "autologin": "0",
                    "autologout": "0",
                    "lang": "default",
                    "refresh": "30s",
                    "theme": "default",
                    "attempt_failed": "0",
                    "attempt_ip": "",
                    "attempt_clock": "0",
                    "rows_per_page": "50",
                    "timezone": "default",
                    "roleid": "2",
                    "provisioned": "0"
                }
            ]
        },
        {
            "mediatypeid": "3",
            "type": "2",
            "name": "SMS",
            "status": "0",
```

```
            "description": "",
            "maxattempts": "3",
            "users": []
        }
    ],
    "id": 1
}
```

Retrieve script and webhook media types

The following example returns three media types:

- script media type with parameters;
- script media type without parameters;
- webhook media type with parameters.

Request:

```
{
    "jsonrpc": "2.0",
    "method": "mediatype.get",
    "params": {
        "output": ["mediatypeid", "name", "parameters"],
        "filter": {
            "type": [1, 4]
        }
    },
    "id": 1
}
```

Response:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "mediatypeid": "10",
            "name": "Script with parameters",
            "parameters": [
                {
                    "sortorder": "0",
                    "value": "{ALERT.SENDTO}"
                },
                {
                    "sortorder": "1",
                    "value": "{EVENT.NAME}"
                },
                {
                    "sortorder": "2",
                    "value": "{ALERT.MESSAGE}"
                },
                {
                    "sortorder": "3",
                    "value": "Zabbix alert"
                }
            ]
        },
        {
            "mediatypeid": "13",
            "name": "Script without parameters",
            "parameters": []
        },
        {
            "mediatypeid": "11",
            "name": "Webhook",
            "parameters": [
```

```
                {
                    "name": "alert_message",
                    "value": "{ALERT.MESSAGE}"
                },
                {
                    "name": "event_update_message",
                    "value": "{EVENT.UPDATE.MESSAGE}"
                },
                {
                    "name": "host_name",
                    "value": "{HOST.NAME}"
                },
                {
                    "name": "trigger_description",
                    "value": "{TRIGGER.DESCRIPTION}"
                },
                {
                    "name": "trigger_id",
                    "value": "{TRIGGER.ID}"
                },
                {
                    "name": "alert_source",
                    "value": "Zabbix"
                }
            ]
        }
    ],
    "id": 1
}
```

Voir également

- Utilisateur

Source

CMediaType::get() dans *ui/include/classes/api/services/CMediaType.php*.

mediatype.update

Description

`object mediatype.update(object/array mediaTypes)`

Cette méthode vous permet de mettre à jour des types de média existants.

> **Note:**
> Cette méthode est uniquement disponible pour le type d'utilisateur *Super administrateur*. Les autorisations d'appeler la méthode peuvent être révoquées dans les paramètres des rôles utilisateur. Consultez la page Rôles utilisateur pour plus d'informations.
 Si des méthodes de livraison sont spécifiées dans l'indicateur de fonctionnalité `$ZBX_FEATURE_FLAGS['media_type_denylist']` du fichier de configuration de l'interface, l'utilisateur n'est pas autorisé à appeler la méthode pour les types de média correspondants.

Paramètres

`(object/array)` Propriétés du type de média à mettre à jour.

La propriété `mediatypeid` doit être définie pour chaque type de média, toutes les autres propriétés sont facultatives. Seules les propriétés transmises seront mises à jour, toutes les autres resteront inchangées.

En plus des propriétés standard du type de média, la méthode accepte les paramètres suivants.

Paramètre	Type	Description
message_templates	array	Modèles de message pour remplacer les modèles de message actuels.

Valeurs retournées

`(object)` Retourne un objet contenant les identifiants des types de média mis à jour sous la propriété `mediatypeids`.

Exemples

Activation d'un type de média

Activez un type de média, c'est-à-dire définissez son statut sur "0".

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "mediatype.update",
    "params": {
        "mediatypeid": "6",
        "status": "0"
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "mediatypeids": [
            "6"
        ]
    },
    "id": 1
}
```

Source

CMediaType::update() dans *ui/include/classes/api/services/CMediaType.php*.

Tâche

Cette classe est conçue pour fonctionner avec des tâches (telles que la vérification des éléments ou des règles de découverte de bas niveau sans rechargement de la configuration).

Références d'objet :

- Task
  - Objet de requête 'Execute now'
  - Objet de requête 'Refresh proxy configuration'
  - Objet de requête 'Diagnostic information'
    - Objet de requête Statistic
    - Objet de résultat Statistic

Méthodes disponibles :

- task.create - créer de nouvelles tâches
- task.get - récupérer des tâches

Objet de tâche

Les objets suivants sont directement liés à l'API `task`.

Tâche

L'objet tâche possède les propriétés suivantes :

Propriété	Type	Description
taskid	ID	ID de la tâche. Comportement de la propriété : - *lecture seule*
type	integer	Type de la tâche. Valeurs possibles : 1 - Informations de diagnostic ; 2 - Actualiser la configuration du proxy ; 6 - Exécuter maintenant. Les utilisateurs de type *Admin* et *User* peuvent créer uniquement des tâches « Exécuter maintenant ». Notez que `task.get` renvoie toujours « 7 » (Résumé d'exécution de tâche). Comportement de la propriété : - *obligatoire*
status	integer	Statut de la tâche. Valeurs possibles : 1 - nouvelle tâche ; 2 - tâche en cours ; 3 - tâche terminée ; 4 - tâche expirée. Comportement de la propriété : - *lecture seule*
clock	timestamp	Heure de création de la tâche. Comportement de la propriété : - *lecture seule*
ttl	integer	Temps en secondes après lequel la tâche expire. Comportement de la propriété : - *lecture seule*
proxyid	ID	ID du proxy pour lequel les statistiques des informations de diagnostic sont collectées. Comportement de la propriété : - *pris en charge* si `type` est défini sur « Informations de diagnostic » ou « Actualiser la configuration du proxy »
request	object	Objet de requête de tâche selon le type de tâche : L'objet de la tâche « Exécuter maintenant » est décrit en détail ci-dessous ; L'objet de la tâche « Actualiser la configuration du proxy » est décrit en détail ci-dessous ; L'objet de la tâche « Informations de diagnostic » est décrit en détail ci-dessous. Comportement de la propriété : - *obligatoire*
result	object	Objet de résultat de la tâche d'informations de diagnostic. Peut contenir NULL si le résultat n'est pas encore prêt. L'objet résultat est décrit en détail ci-dessous. Comportement de la propriété : - *lecture seule*

Objet de requête « Execute now »

L'objet de requête de tâche « Execute now » possède les propriétés suivantes.

Propriété	Type	Description
itemid	ID	ID de l'élément et des règles de découverte de bas niveau. Les utilisateurs de type *Admin* et *User* peuvent utiliser « Execute now » pour les éléments sur les hôtes pour lesquels ils disposent d'une permission *lecture-écriture*, ou d'une permission *lecture* avec l'action `invoke_execute_now` activée pour leur rôle. Il en va de même pour les utilisateurs de type *Admin* concernant les règles de découverte de bas niveau (LLD).

Objet de requête « Refresh proxy configuration »

L'objet de requête de tâche « Refresh proxy configuration » possède les propriétés suivantes.

Propriété	Type	Description
proxyids	array	ID de proxy.

Objet de requête « Diagnostic information »

L'objet de requête de tâche d'informations de diagnostic possède les propriétés suivantes. L'objet de requête de statistiques pour tous les types de propriétés est décrit en détail ci-dessous.

Propriété	Type	Description
historycache	object	Requête de statistiques du cache d'historique. Disponible sur le serveur et le proxy.
valuecache	object	Requête de statistiques du cache des éléments. Disponible sur le serveur.
preprocessing	object	Requête de statistiques du gestionnaire de prétraitement. Disponible sur le serveur et le proxy.
alerting	object	Requête de statistiques du gestionnaire d'alertes. Disponible sur le serveur.
lld	object	Requête de statistiques du gestionnaire LLD. Disponible sur le serveur.

Objet de requête de statistiques

L'objet de requête de statistiques est utilisé pour définir quel type d'informations doit être collecté sur les processus internes du serveur/proxy. Il possède les propriétés suivantes.

Propriété	Type	Description
stats	query	Propriétés de l'objet de statistiques à renvoyer. La liste des champs disponibles pour chaque type de statistique d'informations de diagnostic est décrite en détail ci-dessous. Par défaut : `extend` renverra tous les champs de statistiques disponibles.
top	object	Objet permettant de trier et de limiter les valeurs de statistiques renvoyées. La liste des champs disponibles pour chaque type de statistique d'informations de diagnostic est décrite en détail ci-dessous. Exemple : { "source.alerts": 10 }

Liste des champs statistiques disponibles pour chaque type de requête d'informations de diagnostic

Les champs statistiques suivants peuvent être demandés pour chaque type de propriété de requête d'informations de diagnostic.

Type de diagnostic	Champs disponibles	Description
historycache	items	Nombre d'éléments en cache.
	values	Nombre de valeurs en cache.
	memory	Statistiques de mémoire partagée (espace libre, nombre de segments utilisés, nombre de segments libres, taille maximale du segment libre).
	memory.data	Statistiques de mémoire partagée du cache des données d'historique.
	memory.index	Statistiques de mémoire partagée du cache d'index d'historique.
valuecache	items	Nombre d'éléments en cache.
	values	Nombre de valeurs en cache.
	memory	Statistiques de mémoire partagée (espace libre, nombre de segments utilisés, nombre de segments libres, taille maximale du segment libre).
	mode	Mode du cache de valeurs.
preprocessing	values	Nombre de valeurs en file d'attente.
	preproc.values	Nombre de valeurs en file d'attente avec des étapes de prétraitement.
alerting	alerts	Nombre d'alertes en file d'attente.
lld	rules	Nombre de règles en file d'attente.
	values	Nombre de valeurs en file d'attente.

Liste des champs de tri disponibles pour chaque type de demande d'informations de diagnostic

Les champs statistiques suivants peuvent être utilisés pour trier et limiter les informations demandées.

Type de diagnostic	Champs disponibles	Type
historycache	values	integer
valuecache	values	integer
	request.values	integer
preprocessing	values	integer
alerting	media.alerts	integer
	source.alerts	integer
lld	values	integer

Objet de résultat statistique

L'objet de résultat statistique est récupéré dans le champ `result` de l'objet de tâche.

Propriété	Type	Description
status	integer	Statut du résultat de la tâche. Valeurs possibles : -1 - une erreur s'est produite lors de l'exécution de la tâche ; 0 - le résultat de la tâche est créé. Comportement de la propriété : - *lecture seule*
data	string/object	Résultats selon l'objet de requête statistique de la tâche d'informations de diagnostic particulière. Contient une chaîne de message d'erreur si une erreur s'est produite lors de l'exécution de la tâche.

task.create

Description

`object task.create(object/array tasks)`

Cette méthode vous permet de créer des tâches.

> **Note:**
> Cette méthode est disponible pour les utilisateurs de tout type. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres des rôles utilisateur. Consultez Rôles utilisateur pour plus d'informations.

Paramètres

`(object/array)` Une tâche à créer.

La méthode accepte des tâches avec les propriétés de tâche standard.

Notez que les tâches « Exécuter maintenant » ne peuvent être créées que pour les types d'éléments/règles de découverte suivants :

- Agent Zabbix (passif)
- Vérification simple
- Agent SNMP (v1/v2/v3)
- Interne Zabbix
- Vérification externe
- Surveillance de base de données
- Agent HTTP
- Agent IPMI
- Agent SSH
- Agent TELNET
- Agent JMX
- Calculé
- Élément dépendant
- Script
- Navigateur

Si l'élément/la règle de découverte est de type « Élément dépendant », alors son élément maître doit également être de l'un des types ci-dessus.

Valeurs retournées

`(object)` Retourne un objet contenant les identifiants des tâches créés sous la propriété `taskids`. L'ordre des ID retournés correspond à l'ordre des tâches transmises.

Exemples

Création d'une tâche

Créez une tâche « Exécuter maintenant » pour un élément et une règle de découverte de bas niveau.

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "task.create",
    "params": [
        {
            "type": 6,
            "request": {
                "itemid": "10092"
            }
        },
        {
            "type": 6,
            "request": {
                "itemid": "10093"
            }
        }
    ],
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "taskids": [
            "1",
            "2"
        ]
```

```
    },
    "id": 1
}
```

Créez une tâche « Actualiser la configuration du proxy » pour deux proxys.

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "task.create",
    "params": [
        {
            "type": 2,
            "request": {
                "proxyids": ["10459", "10460"]
            }
        }
    ],
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "taskids": [
            "1"
        ]
    },
    "id": 1
}
```

Créez une tâche « Informations de diagnostic ».

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "task.create",
    "params": [
        {
            "type": 1,
            "request": {
                "alerting": {
                    "stats": [
                        "alerts"
                    ],
                    "top": {
                        "media.alerts": 10
                    }
                },
                "lld": {
                    "stats": "extend",
                    "top": {
                        "values": 5
                    }
                }
            },
            "proxyid": 0
        }
    ],
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "taskids": [
            "3"
        ]
    },
    "id": 1
}
```

Voir aussi

- Tâche
- Objet de requête « Exécuter maintenant »
- Objet de requête « Actualiser la configuration du proxy »
- Objet de requête « Informations de diagnostic »

Source

CTask::create() dans *ui/include/classes/api/services/CTask.php*.

task.get

Description

`integer/array task.get(object parameters)`

La méthode permet de récupérer des tâches selon les paramètres fournis.

> **Note:**
> Cette méthode est disponible uniquement pour le type d'utilisateur *Super admin*. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres du rôle utilisateur. Voir User roles pour plus d'informations.

Paramètres

`(object)` Paramètres définissant la sortie souhaitée.

La méthode prend en charge les paramètres suivants.

Paramètre	Type	Description
taskids	ID/array	Retourner uniquement les tâches avec les ID donnés.
output	query	Ces paramètres sont décrits dans le commentaire de référence.
preservekeys	boolean	

Valeurs retournées

`(integer/array)` Retourne un tableau d'objets.

Exemples

Récupérer une tâche par ID

Récupérez toutes les données concernant la tâche avec l'ID « 1 ».

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "task.get",
    "params": {
        "output": "extend",
        "taskids": "1"
    },
    "id": 1
}
```

Réponse :


```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "taskid": "1",
            "type": "7",
            "status": "3",
            "clock": "1601039076",
            "ttl": "3600",
            "proxyid": null,
            "request": {
                "alerting": {
                    "stats": [
                        "alerts"
                    ],
                    "top": {
                        "media.alerts": 10
                    }
                },
                "lld": {
                    "stats": "extend",
                    "top": {
                        "values": 5
                    }
                }
            },
            "result": {
                "data": {
                    "alerting": {
                        "alerts": 0,
                        "top": {
                            "media.alerts": []
                        },
                        "time": 0.000663
                    },
                    "lld": {
                        "rules": 0,
                        "values": 0,
                        "top": {
                            "values": []
                        },
                        "time": 0.000442
                    }
                },
                "status": "0"
            }
        }
    ],
    "id": 1
}
```


Voir aussi

- Tâche
- Objet de requête « Exécuter maintenant »
- Objet de requête « Actualiser la configuration du proxy »
- Objet de requête « Informations de diagnostic »

Source

CTask::get() dans *ui/include/classes/api/services/CTask.php*.

Utilisateur

Cette classe est conçue pour fonctionner avec les utilisateurs.

Références d'objet :

- Utilisateur
- Média

Méthodes disponibles :

- user.checkauthentication - vérifier et prolonger les sessions utilisateur
- user.create - créer de nouveaux utilisateurs
- user.delete - supprimer des utilisateurs
- user.get - récupérer des utilisateurs
- user.login - se connecter à l'API
- user.logout - se déconnecter de l'API
- user.provision - provisionner des utilisateurs LDAP
- user.resettotp - réinitialiser les secrets TOTP des utilisateurs
- user.unblock - débloquer des utilisateurs
- user.update - mettre à jour des utilisateurs

Objet utilisateur

Les objets suivants sont directement liés à l'API `user`.

Utilisateur

L'objet utilisateur possède les propriétés suivantes.

Propriété	Type	Description
userid	ID	ID de l'utilisateur. Comportement de la propriété : - *en lecture seule* - *obligatoire* pour les opérations de mise à jour
username	string	Nom de l'utilisateur. Comportement de la propriété : - *obligatoire* pour les opérations de création - *en lecture seule* pour les utilisateurs provisionnés si l'utilisateur est lié à un annuaire d'utilisateurs (`userdirectoryid` est défini sur une valeur valide différente de « 0 »), que le provisionnement de l'annuaire d'utilisateurs est activé (`provision_status` de l'objet Annuaire d'utilisateurs est défini sur « 1 ») et que l'authentification de tous les provisionnements LDAP ou SAML est activée (`ldap_jit_status` de l'objet Authentification est défini sur « Enabled for configured LDAP IdPs » ou `saml_jit_status` de l'objet Authentification est défini sur « Enabled for configured SAML IdPs »)
passwd	string	Mot de passe de l'utilisateur. La valeur de ce paramètre peut être une chaîne vide si l'utilisateur est lié à un annuaire d'utilisateurs. Comportement de la propriété : - *en écriture seule*
roleid	ID	ID du rôle de l'utilisateur. Notez que les utilisateurs sans rôle peuvent se connecter à Zabbix uniquement à l'aide de l'authentification LDAP ou SAML, à condition que leurs informations LDAP/SAML correspondent aux mappages de groupes d'utilisateurs configurés dans Zabbix.

Propriété	Type	Description
attempt_clock	timestamp	Heure de la dernière tentative de connexion infructueuse. Comportement de la propriété : - *en lecture seule*
attempt_failed	integer	Nombre de tentatives de connexion infructueuses récentes. Comportement de la propriété : - *en lecture seule*
attempt_ip	string	Adresse IP depuis laquelle la dernière tentative de connexion infructueuse a été effectuée. Comportement de la propriété : - *en lecture seule*
autologin	integer	Indique s'il faut activer la connexion automatique. Valeurs possibles : 0 - *(par défaut)* connexion automatique désactivée ; 1 - connexion automatique activée.
autologout	string	Durée de vie de la session utilisateur. Accepte un nombre de secondes (par exemple, 1, 60, 300) ou une valeur avec un suffixe temporel (par exemple, 30s, 1m, 2h). Si la valeur est définie sur 0s, la session n'expire jamais. Valeur par défaut : 15m.
default_maintenance_period	string	Période de maintenance par défaut pour une maintenance créée à l'aide d'un raccourci. Accepte un nombre de secondes ou une unité de temps avec suffixe (300s, 5m, 1h, 1d, 1w). La valeur indiquée est arrondie à la minute entière inférieure. Valeur par défaut : 1h.
lang	string	Code de langue de l'utilisateur, par exemple `en_US`. Valeur par défaut : `default` - valeur par défaut du système.
name	string	Prénom de l'utilisateur.
provisioned	integer	Indique si l'utilisateur a été provisionné. Valeurs possibles : 0 - non provisionné ; 1 - provisionné. Comportement de la propriété : - *en lecture seule*
refresh	string	Période d'actualisation automatique. Accepte un nombre de secondes (par exemple, 1, 60, 300) ou une valeur avec un suffixe temporel (par exemple, 30s, 1m, 1h). Valeur par défaut : 30s.
rows_per_page	integer	Nombre de lignes d'objets à afficher par page. Valeur par défaut : 50.
surname	string	Nom de famille de l'utilisateur.
theme	string	Thème de l'utilisateur. Valeurs possibles : `default` - *(par défaut)* valeur par défaut du système ; `blue-theme` - Bleu ; `blue-classic-theme` - Bleu (classique) ; `dark-theme` - Sombre ; `dark-blue-theme` - Bleu foncé ; `dark-classic-theme` - Sombre (classique) ; `hc-light` - Clair à contraste élevé ; `hc-dark` - Sombre à contraste élevé.

Propriété	Type	Description
ts_provisioned	timestamp	Heure à laquelle la dernière opération de provisionnement a été effectuée. Comportement de la propriété : - *en lecture seule* - *pris en charge* pour les utilisateurs de type *Super administrateur*
url	string	URL de la page vers laquelle rediriger l'utilisateur après sa connexion.
userdirectoryid	ID	ID de l'annuaire d'utilisateurs auquel l'utilisateur est lié. Utilisé pour le provisionnement (création ou mise à jour), ainsi que pour connecter un utilisateur lié à un annuaire d'utilisateurs. Pour les opérations de connexion, la valeur de cette propriété est prioritaire sur la propriété `userdirectoryid` des groupes d'utilisateurs auxquels l'utilisateur appartient. Valeur par défaut : 0. Comportement de la propriété : - *en lecture seule* - *pris en charge* pour les utilisateurs de type *Super administrateur*
timezone	string	Fuseau horaire de l'utilisateur, par exemple `Europe/London`, `UTC`. Valeur par défaut : `default` - valeur par défaut du système. Pour obtenir la liste complète des fuseaux horaires pris en charge, consultez la documentation PHP.

Média

L'objet média possède les propriétés suivantes.

Property	Type	Description
mediaid	ID	ID du média de l'utilisateur. Comportement de la propriété : - *lecture seule*
mediatypeid	ID	ID du type de média utilisé par le média de l'utilisateur. Comportement de la propriété : - *obligatoire*
sendto	string/array	Adresse, nom d'utilisateur ou autre identifiant du destinataire. Si `type` du type de média est défini sur « Email » ou « Push », les valeurs sont représentées sous forme de tableau. Pour « Push », le tableau peut contenir un caractère générique (`*`) ou un ou plusieurs ID d'appareils mobiles. Si `type` du type de média est défini sur « Script », « SMS » ou « Webhook », la valeur est représentée sous forme de chaîne. Comportement de la propriété : - *obligatoire*
active	integer	Indique si le média est activé. Valeurs possibles : 0 - *(par défaut)* activé ; 1 - désactivé.

Property	Type	Description
severity	integer	Niveaux de gravité des déclencheurs pour lesquels envoyer des notifications. Valeurs bitmap possibles : 1 - Non classé ; 2 - Information ; 4 - Avertissement ; 8 - Moyenne ; 16 - Élevée ; 32 - Désastre. Il s'agit d'un champ de masque de bits ; toute somme de valeurs bitmap possibles est acceptée (par exemple, 48 pour Moyenne, Élevée et Désastre). Valeur par défaut : 63.
period	string	Période pendant laquelle les notifications peuvent être envoyées, sous forme de période ou de macros utilisateur séparées par un point-virgule. Valeur par défaut : 1-7,00:00-24:00.
provisioned	integer	Indique si l'utilisateur a été provisionné. Valeurs possibles : 0 - non provisionné ; 1 - provisionné.
userdirectory_mediaid	ID	ID du mappage de média de l'annuaire des utilisateurs pour le média provisionné. Comportement de la propriété : - *lecture seule* - *pris en charge* pour les utilisateurs de type *Super administrateur*

user.checkAuthentication

Description

`object user.checkAuthentication`

Cette méthode vérifie et prolonge la session utilisateur.

> **Attention:**
> L'appel de la méthode `user.checkAuthentication` à l'aide du paramètre `sessionid` prolonge la session utilisateur par défaut.

Paramètres

La méthode accepte les paramètres suivants.

Parameter	Type	Description
extend	boolean	Indique s'il faut prolonger la session utilisateur. Valeur par défaut: "true". Définir la valeur sur "false" permet de vérifier la session utilisateur sans la prolonger. Comportement du paramètre: - *supported* si `sessionid` est défini

Parameter	Type	Description
sessionid	string	Jeton d'authentification de l'utilisateur. Comportement du paramètre: - *required* si `token` n'est pas défini
token	string	Jeton API de l'utilisateur. Comportement du paramètre: - *required* si `sessionid` n'est pas défini

Valeurs de retour

`(object)` Renvoie un objet contenant des informations sur l'utilisateur.

En plus des propriétés utilisateur standard, les informations suivantes sont renvoyées.

Propriété	Type	Description
auth_type	integer	Authentification par défaut pour l'utilisateur. Consultez la propriété `authentication_type` de l'objet Authentication pour obtenir la liste des valeurs possibles.
debug_mode	integer	Indique si le mode de débogage est activé ou désactivé pour l'utilisateur. Consultez la propriété `debug_mode` de l'objet Groupe d'utilisateurs pour obtenir la liste des valeurs possibles.
deprovisioned	boolean	Indique si l'utilisateur appartient à un groupe d'utilisateurs déprovisionnés.
gui_access	string	Méthode d'authentification de l'utilisateur pour l'interface. Consultez la propriété `gui_access` de l'objet Groupe d'utilisateurs pour obtenir la liste des valeurs possibles.
secret	string	Chaîne aléatoire de 32 caractères. Elle est générée lors de la connexion de l'utilisateur. La propriété `secret` n'est pas renvoyée si la session utilisateur est vérifiée à l'aide d'un jeton d'API.
sessionid	string	Jeton d'authentification, qui doit être utilisé dans les requêtes API suivantes. La propriété `sessionid` n'est pas renvoyée si la session utilisateur est vérifiée à l'aide d'un jeton d'API.
type	integer	Type d'utilisateur. Consultez la propriété `type` de l'objet Role pour obtenir la liste des valeurs possibles.
userip	string	Adresse IP de l'utilisateur.

Exemples

Vérifier l'authentification à l'aide du jeton d'authentification

Vérifiez et prolongez une session utilisateur à l'aide du jeton d'authentification de l'utilisateur, puis renvoyez des informations supplémentaires sur l'utilisateur.

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "user.checkAuthentication",
    "params": {
        "sessionid": "673b8ba11562a35da902c66cf5c23fa2"
    },
```

```
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "userid": "1",
        "username": "Admin",
        "name": "Zabbix",
        "surname": "Administrator",
        "url": "",
        "autologin": "1",
        "autologout": "0",
        "lang": "ru_RU",
        "refresh": "0",
        "theme": "default",
        "attempt_failed": "0",
        "attempt_ip": "127.0.0.1",
        "attempt_clock": "1355919038",
        "rows_per_page": "50",
        "timezone": "Europe/Riga",
        "roleid": "3",
        "userdirectoryid": "0",
        "ts_provisioned": "0",
        "type": 3,
        "userip": "127.0.0.1",
        "debug_mode": 0,
        "gui_access": "0",
        "deprovisioned": false,
        "auth_type": 0,
        "sessionid": "673b8ba11562a35da902c66cf5c23fa2",
        "secret": "0e329b933e46984e49a5c1051ecd0751"
    },
    "id": 1
}
```

Vérifier l'authentification à l'aide d'un jeton API

Vérifiez une session utilisateur à l'aide du jeton API de l'utilisateur et renvoyez des informations supplémentaires sur l'utilisateur.

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "user.checkAuthentication",
    "params": {
        "token": "00aff470e07c12d707e50d98cfe39edef9e6ec349c14728dbdfbc8ddc5ea3eae"
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "userid": "1",
        "username": "Admin",
        "name": "Zabbix",
        "surname": "Administrator",
        "url": "",
        "autologin": "1",
        "autologout": "0",
        "lang": "ru_RU",
```

```
        "refresh": "0",
        "theme": "default",
        "attempt_failed": "0",
        "attempt_ip": "127.0.0.1",
        "attempt_clock": "1355919338",
        "rows_per_page": "50",
        "timezone": "Europe/Riga",
        "roleid": "3",
        "userdirectoryid": "0",
        "ts_provisioned": "0",
        "debug_mode": 0,
        "deprovisioned": false,
        "gui_access": "1",
        "mfaid": 0,
        "auth_type": 0,
        "type": 3,
        "userip": "127.0.0.1"
    },
    "id": 1
}
```


Source

CUser::checkAuthentication() dans *ui/include/classes/api/services/CUser.php*.

user.create

Description

`object user.create(object/array users)`

Cette méthode vous permet de créer de nouveaux utilisateurs.

Note:
Cette méthode est uniquement disponible pour le type d'utilisateur *Super administrateur*. Les autorisations d'appeler cette méthode peuvent être révoquées dans les paramètres des rôles utilisateur. Consultez Rôles utilisateur pour plus d'informations.

Note:
La robustesse du mot de passe utilisateur est validée conformément aux règles de la politique de mots de passe définies par l'API Authentication. Consultez objet authentication pour plus d'informations.

Paramètres

`(object/array)` Utilisateurs à créer.

En plus des propriétés utilisateur standard, la méthode accepte les paramètres suivants.

Paramètre	Type	Description
usrgrps	array	Groupes d'utilisateurs auxquels ajouter l'utilisateur. Les groupes d'utilisateurs doivent avoir uniquement la propriété `usrgrpid` définie.
medias	array	Médias utilisateur à créer.

Valeurs retournées

`(object)` Renvoie un objet contenant les ID des utilisateurs créés sous la propriété `userids`. L'ordre des ID renvoyés correspond à l'ordre des utilisateurs passés.

Exemples

Création d'un utilisateur

Créez un nouvel utilisateur, ajoutez-le à un groupe d'utilisateurs et créez un nouveau média pour lui.

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "user.create",
    "params": {
        "username": "John",
        "passwd": "Doe12345",
        "roleid": "5",
        "usrgrps": [
            {
                "usrgrpid": "7"
            }
        ],
        "medias": [
            {
                "mediatypeid": "1",
                "sendto": [
                    "support@company.com"
                ],
                "active": 0,
                "severity": 63,
                "period": "1-7,00:00-24:00"
            }
        ]
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "userids": [
            "12"
        ]
    },
    "id": 1
}
```

Voir aussi

- Authentification
- Média
- Groupe d'utilisateurs
- Rôle

Source

CUser::create() dans *ui/include/classes/api/services/CUser.php*.

user.delete

Description

`object user.delete(array users)`

Cette méthode vous permet de supprimer des utilisateurs.

> **Note:**
> Cette méthode est disponible uniquement pour le type d'utilisateur *Super admin*. Les autorisations d'appel de cette méthode peuvent être révoquées dans les paramètres du rôle utilisateur. Consultez User roles pour plus d'informations.

Paramètres

`(array)` ID des utilisateurs à supprimer.

Valeurs de retour

`(object)` Renvoie un objet contenant les identifiants des utilisateurs supprimés dans la propriété `userids`.

Exemples

Suppression de plusieurs utilisateurs

Supprimez deux utilisateurs.

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "user.delete",
    "params": [
        "1",
        "5"
    ],
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "userids": [
            "1",
            "5"
        ]
    },
    "id": 1
}
```

Source

CUser::delete() dans *ui/include/classes/api/services/CUser.php*.

user.get

Description

`integer/array user.get(object parameters)`

La méthode permet de récupérer des utilisateurs selon les paramètres fournis.

> **Note:**
> Cette méthode est disponible pour les utilisateurs de tout type. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres des rôles utilisateur. Consultez Rôles utilisateur pour plus d'informations.

Paramètres

`(object)` Paramètres définissant la sortie souhaitée.

La méthode prend en charge les paramètres suivants.

> **Note:**
> Lors de la demande d'informations sur les médias utilisateur, les permissions ou les rôles, les utilisateurs de type *Admin* et *User* ne peuvent récupérer des données que sur leur propre utilisateur. Pour un exemple, voir Récupération des utilisateurs en tant qu'*Admin*.

Paramètre	Type	Description
mediaids	ID/array	Ne renvoyer que les utilisateurs qui utilisent le média indiqué.
mediatypeids	ID/array	Ne renvoyer que les utilisateurs qui utilisent les types de média indiqués.
userids	ID/array	Ne renvoyer que les utilisateurs ayant les ID indiqués.

Paramètre	Type	Description
usrgrpids	ID/array	Ne renvoyer que les utilisateurs qui appartiennent aux groupes d'utilisateurs indiqués.
getAccess	flag	Ajoute des informations supplémentaires sur les permissions de l'utilisateur. Ajoute les propriétés suivantes pour chaque utilisateur : `gui_access` - `(integer)` méthode d'authentification de l'utilisateur dans l'interface. Reportez-vous à la propriété `gui_access` de l'objet groupe d'utilisateurs pour obtenir la liste des valeurs possibles. `debug_mode` - `(integer)` indique si le débogage est activé pour l'utilisateur. Valeurs possibles : 0 - débogage désactivé, 1 - débogage activé. `users_status` - `(integer)` indique si l'utilisateur est désactivé. Valeurs possibles : 0 - utilisateur activé, 1 - utilisateur désactivé.
selectMedias	query	Renvoyer les médias utilisés par l'utilisateur dans la propriété `medias`.
selectMediatypes	query	Renvoyer les types de média utilisés par l'utilisateur dans la propriété `mediatypes`. Voir `mediatype.get` pour les restrictions selon le type d'utilisateur.
selectUsrgrps	query	Renvoyer les groupes d'utilisateurs auxquels l'utilisateur appartient dans la propriété `usrgrps`. Voir `usergroup.get` pour les restrictions selon le type d'utilisateur.
selectRole	query	Renvoyer le rôle de l'utilisateur dans la propriété `role`.
filter	object	Ne renvoyer que les résultats qui correspondent exactement au filtre indiqué. Accepte un objet où les clés sont des noms de propriété et les valeurs sont soit une valeur unique, soit un tableau de valeurs à comparer. Ne prend pas en charge les propriétés de type de données `text` type de données.
output	query	Propriétés possibles de l'objet utilisateur pour les utilisateurs de type *Admin* et *User* lors de la demande de données sur des utilisateurs de leur groupe d'utilisateurs : `userid`, `name`, `surname`, `username`. Propriétés de l'objet utilisateur à renvoyer. Les utilisateurs de type *Admin* et *User* ne peuvent récupérer que les propriétés suivantes : - Pour leur propre utilisateur : `userid`, `attempt_clock`, `attempt_failed`, `attempt_ip`, `autologin`, `autologout`, `lang`, `name`, `provisioned`, `refresh`, `roleid`, `rows_per_page`, `surname`, `theme`, `timezone`, `url`, `username`. - Pour les utilisateurs de leur groupe d'utilisateurs : `userid`, `name`, `surname`, `username`. Par défaut : `extend`.
search	object	Renvoyer les résultats qui correspondent au motif indiqué (insensible à la casse). Accepte un objet où les clés sont des noms de propriété et les valeurs sont des chaînes à rechercher. Si aucune option supplémentaire n'est fournie, cela effectuera une recherche `LIKE "%…%"`. Prend en charge uniquement les propriétés de type de données `string` et `text` type de données. Propriétés possibles de l'objet utilisateur pour les utilisateurs de type *Admin* et *User* lors de la demande de données sur des utilisateurs de leur groupe d'utilisateurs : `name`, `surname`, `username`.

Paramètre	Type	Description
sortfield	string/array	Trier le résultat selon les propriétés indiquées. Valeurs possibles : `userid`, `username`.
countOutput	boolean	Ces paramètres sont décrits dans le commentaire de référence.
editable	boolean	
excludeSearch	boolean	
limit	integer	
preservekeys	boolean	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Valeurs retournées

`(integer/array)` Renvoie soit :

- un tableau d'objets ;
- le nombre d'objets récupérés, si le paramètre `countOutput` a été utilisé.

Exemples

Récupération des utilisateurs

Récupérer tous les utilisateurs configurés.

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "user.get",
    "params": {
        "output": "extend"
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "userid": "1",
            "username": "Admin",
            "name": "Zabbix",
            "surname": "Administrator",
            "url": "",
            "autologin": "1",
            "autologout": "0",
            "lang": "en_US",
            "default_maintenance_period": "1h",
            "refresh": "30s",
            "theme": "default",
            "attempt_failed": "0",
            "attempt_ip": "",
            "attempt_clock": "0",
            "rows_per_page": "50",
            "timezone": "default",
            "roleid": "3",
            "userdirectoryid": "0",
            "ts_provisioned": "0",
            "provisioned": "0"
        },
        {
```

```
            "userid": "2",
            "username": "guest",
            "name": "",
            "surname": "",
            "url": "",
            "autologin": "0",
            "autologout": "15m",
            "lang": "default",
            "default_maintenance_period": "1h",
            "refresh": "30s",
            "theme": "default",
            "attempt_failed": "0",
            "attempt_ip": "",
            "attempt_clock": "0",
            "rows_per_page": "50",
            "timezone": "default",
            "roleid": "4",
            "userdirectoryid": "0",
            "ts_provisioned": "0",
            "provisioned": "0"
        }
    ],
    "id": 1
}
```

Récupération des utilisateurs en tant qu'*Admin*

En tant qu'utilisateur de type *Admin*, récupérez les données détaillées de votre propre utilisateur ainsi que les données limitées des utilisateurs de votre groupe d'utilisateurs.

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "user.get",
    "params": {
        "output": "extend",
        "getAccess": true,
        "selectMedias": "extend",
        "selectMediatypes": "extend",
        "selectUsrgrps": "extend",
        "selectRole": "extend"
    },
    "id": 1
}
```

Réponse :

```
{
  "jsonrpc": "2.0",
  "result": [
    {
      "userid": "1",
      "username": "Admin",
      "name": "Zabbix",
      "surname": "Administrator",
      "url": "",
      "autologin": "1",
      "autologout": "0",
      "lang": "default",
      "default_maintenance_period": "1h",
      "refresh": "30s",
      "theme": "default",
      "attempt_failed": "0",
      "attempt_ip": "",
```

```
    "attempt_clock": "0",
    "rows_per_page": "50",
    "timezone": "default",
    "roleid": "3",
    "userdirectoryid": "0",
    "ts_provisioned": "0",
    "provisioned": "0",
    "gui_access": "1",
    "debug_mode": "0",
    "users_status": "0",
    "usrgrps": [
      {
        "usrgrpid": "7",
        "name": "Zabbix administrators",
        "gui_access": "0",
        "users_status": "0",
        "debug_mode": "0",
        "userdirectoryid": "0",
        "mfa_status": "0",
        "mfaid": "0",
        "proxy_mode": "0",
        "proxy_group_mode": "0"
      },
      {
        "usrgrpid": "13",
        "name": "Internal",
        "gui_access": "1",
        "users_status": "0",
        "debug_mode": "0",
        "userdirectoryid": "0",
        "mfa_status": "0",
        "mfaid": "0",
        "proxy_mode": "0",
        "proxy_group_mode": "0"
      }
    ],
    "medias": [],
    "mediatypes": [],
    "role": {
      "roleid": "3",
      "name": "Super admin role",
      "type": "3",
      "readonly": "1"
    }
  },
  {
    "userid": "2",
    "username": "guest",
    "name": "",
    "surname": "",
    "url": "",
    "autologin": "0",
    "autologout": "15m",
    "lang": "default",
    "default_maintenance_period": "1h",
    "refresh": "30s",
    "theme": "default",
    "attempt_failed": "0",
    "attempt_ip": "",
    "attempt_clock": "0",
    "rows_per_page": "50",
    "timezone": "default",
```

```
      "roleid": "4",
      "userdirectoryid": "0",
      "ts_provisioned": "0",
      "provisioned": "0",
      "gui_access": "1",
      "debug_mode": "0",
      "users_status": "1",
      "usrgrps": [
        {
          "usrgrpid": "8",
          "name": "Guests",
          "gui_access": "0",
          "users_status": "0",
          "debug_mode": "0",
          "userdirectoryid": "0",
          "mfa_status": "0",
          "mfaid": "0",
          "proxy_mode": "0",
          "proxy_group_mode": "0"
        },
        {
          "usrgrpid": "9",
          "name": "Disabled",
          "gui_access": "0",
          "users_status": "1",
          "debug_mode": "0",
          "userdirectoryid": "0",
          "mfa_status": "0",
          "mfaid": "0",
          "proxy_mode": "0",
          "proxy_group_mode": "0"
        },
        {
          "usrgrpid": "13",
          "name": "Internal",
          "gui_access": "1",
          "users_status": "0",
          "debug_mode": "0",
          "userdirectoryid": "0",
          "mfa_status": "0",
          "mfaid": "0",
          "proxy_mode": "0",
          "proxy_group_mode": "0"
        }
      ],
      "medias": [],
      "mediatypes": [],
      "role": {
        "roleid": "4",
        "name": "Guest role",
        "type": "1",
        "readonly": "0"
      }
    }
  ],
  "id": 1
}
```


Récupération des données d'un utilisateur

Récupérer les données de l'utilisateur dont l'ID est « 2 ».

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "user.get",
    "params": {
        "output": ["userid", "username"],
        "selectRole": "extend",
        "userids": "2"
    },
    "id": 1
}
```

Réponse :

```
{
  "jsonrpc": "2.0",
  "result": [
    {
      "userid": "2",
      "username": "guest",
      "role": {
        "roleid": "4",
        "name": "Guest role",
        "type": "1",
        "readonly": "0"
      }
    }
  ],
  "id": 1
}
```

Voir aussi

- Média
- Type de média
- Groupe d'utilisateurs
- Rôle

Source

CUser::get() dans *ui/include/classes/api/services/CUser.php*.

user.login

Description

`string/object user.login(object parameters)`

Cette méthode vous permet de vous connecter à l'API et de générer un jeton d'authentification.

Warning:
Lors de l'utilisation de cette méthode, vous devez également effectuer user.logout afin d'éviter la génération d'un grand nombre d'enregistrements de sessions ouvertes.

Attention:
Cette méthode est uniquement disponible pour les utilisateurs non authentifiés qui n'appartiennent à aucun groupe d'utilisateurs avec l'authentification multifacteur activée.

Paramètres

`(object)` Paramètres contenant le nom d'utilisateur et le mot de passe.

La méthode accepte les paramètres suivants.

Paramètre	Type	Description
password	string	Mot de passe de l'utilisateur. Comportement du paramètre: - *obligatoire*
username	string	Nom d'utilisateur. Comportement du paramètre: - *obligatoire*
userData	flag	Retourner des informations sur l'utilisateur authentifié.

Valeurs de retour

(`string/object`) Si le paramètre `userData` est utilisé, renvoie un objet contenant des informations sur l'utilisateur authentifié.

En plus des propriétés utilisateur standard, les informations suivantes sont renvoyées :

Propriété	Type	Description
auth_type	integer	Authentification par défaut pour l'utilisateur. Voir la propriété `authentication_type` de l'objet Authentication pour la liste des valeurs possibles.
debug_mode	integer	Indique si le mode de débogage est activé ou désactivé pour l'utilisateur. Voir la propriété `debug_mode` de l'objet Groupe d'utilisateurs pour la liste des valeurs possibles.
deprovisioned	boolean	Indique si l'utilisateur appartient à un groupe d'utilisateurs déprovisionnés.
gui_access	string	Méthode d'authentification de l'utilisateur pour l'interface. Voir la propriété `gui_access` de l'objet Groupe d'utilisateurs pour la liste des valeurs possibles.
mfaid	integer	ID de la méthode MFA à utiliser pour l'utilisateur lors de la connexion. Renvoie "0" si la MFA est désactivée globalement ou pour tous les groupes d'utilisateurs auxquels l'utilisateur appartient.
secret	string	Chaîne aléatoire de 32 caractères. Elle est générée lors de la connexion de l'utilisateur.
sessionid	string	Jeton d'authentification, qui doit être utilisé dans les requêtes API suivantes.
type	integer	Type d'utilisateur. Voir la propriété `type` de l'objet Role pour la liste des valeurs possibles.
userip	string	Adresse IP de l'utilisateur.

Note:
Si un utilisateur a été authentifié avec succès après une ou plusieurs tentatives échouées, la méthode renverra les valeurs actuelles des propriétés `attempt_clock`, `attempt_failed` et `attempt_ip`, puis les réinitialisera.

Si le paramètre `userData` n'est pas utilisé, la méthode renvoie un jeton d'authentification requis pour l'authentification.

Exemples

Authentification d'un utilisateur

Authentifiez un utilisateur.

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "user.login",
```

```
    "params": {
        "username": "Admin",
        "password": "zabbix"
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": "0424bd59b807674191e7d77572075f33",
    "id": 1
}
```

Demande d'informations sur l'utilisateur authentifié

Authentifiez-vous et renvoyez des informations supplémentaires sur l'utilisateur.

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "user.login",
    "params": {
        "username": "Admin",
        "password": "zabbix",
        "userData": true
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "userid": "1",
        "username": "Admin",
        "name": "Zabbix",
        "surname": "Administrator",
        "url": "",
        "autologin": "1",
        "autologout": "0",
        "lang": "ru_RU",
        "refresh": "0",
        "theme": "default",
        "attempt_failed": "0",
        "attempt_ip": "127.0.0.1",
        "attempt_clock": "1355919038",
        "rows_per_page": "50",
        "timezone": "Europe/Riga",
        "roleid": "3",
        "userdirectoryid": "0",
        "type": 3,
        "userip": "127.0.0.1",
        "debug_mode": 0,
        "gui_access": "0",
        "mfaid": "1",
        "deprovisioned": false,
        "auth_type": 0,
        "sessionid": "5b56eee8be445e98f0bd42b435736e42",
        "secret": "cd0ba923319741c6586f3d866423a8f4"
    },
    "id": 1
}
```

Voir également

- user.logout

Source

CUser::login() dans *ui/include/classes/api/services/CUser.php*.

user.logout

Description

`string/object user.logout(array)`

Cette méthode vous permet de vous déconnecter de l'API et invalide le jeton d'authentification actuel.

> **Note:**
> Cette méthode est disponible pour les utilisateurs de tout type. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres du rôle utilisateur. Consultez Rôles d'utilisateur pour plus d'informations.

Paramètres

`(array)` La méthode accepte un tableau vide.

Valeurs retournées

`(boolean)` Retourne `true` si l'utilisateur a été déconnecté avec succès.

Exemples

Déconnexion

Se déconnecter de l'API.

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "user.logout",
    "params": [],
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": true,
    "id": 1
}
```

Voir également

- user.login

Source

CUser::login() dans *ui/include/classes/api/services/CUser.php*.

user.provision

Description

`object user.provision(object/array users)`

Cette méthode vous permet de provisionner des utilisateurs LDAP.

> **Note:**
> Cette méthode est disponible uniquement pour le type d'utilisateur *Super admin*. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres du rôle utilisateur. Consultez User roles pour plus d'informations.

Paramètres

`(array)` ID des utilisateurs à provisionner.

Valeurs de retour

`(object)` Renvoie un objet contenant les ID des utilisateurs provisionnés dans la propriété `userids`.

Exemples

Provisionnement de plusieurs utilisateurs

Provisionnez deux utilisateurs.

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "user.provision",
    "params": [
        "1",
        "5"
    ],
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "userids": [
            "1",
            "5"
        ]
    },
    "id": 1
}
```

Source

CUser::provision() dans *ui/include/classes/api/services/CUser.php*.

user.resettotp

Description

`object user.resettotp(object/array users)`

Cette méthode vous permet de réinitialiser les secrets TOTP des utilisateurs.

> **Note:**
> Cette méthode est disponible uniquement pour le type d'utilisateur *Super admin*. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres des rôles utilisateur. Consultez Rôles utilisateur pour plus d'informations.

Paramètres

`(array)` Identifiants des utilisateurs pour lesquels réinitialiser les secrets TOTP.

> **Note:**
> Les sessions utilisateur des utilisateurs spécifiés seront également supprimées (à l'exception de l'utilisateur envoyant la requête).

Valeurs de retour

`(object)` Renvoie un objet contenant les ID des utilisateurs pour lesquels les secrets TOTP ont été réinitialisés, dans la propriété `userids`.

Exemples

Réinitialisation des secrets TOTP pour plusieurs utilisateurs

Réinitialisez les secrets TOTP pour deux utilisateurs.

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "user.resettotp",
    "params": [
        "1",
        "5"
    ],
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "userids": [
            "1",
            "5"
        ]
    },
    "id": 1
}
```

Voir aussi

- Objet MFA

Source

CUser::resettotp() dans *ui/include/classes/api/services/CUser.php*.

user.unblock

Description

`object user.unblock(array userids)`

Cette méthode permet de débloquer des utilisateurs.

> **Note:**
> Cette méthode est disponible uniquement pour le type d'utilisateur *Super admin*. Les autorisations d'appel de cette méthode peuvent être révoquées dans les paramètres du rôle utilisateur. Voir User roles pour plus d'informations.

Paramètres

`(array)` ID des utilisateurs à débloquer.

Valeurs retournées

`(object)` Renvoie un objet contenant les ID des utilisateurs débloqués sous la propriété `userids`.

Exemples

Déblocage de plusieurs utilisateurs

Débloquez deux utilisateurs.

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "user.unblock",
    "params": [
        "1",
        "5"
```

```
    ],
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "userids": [
            "1",
            "5"
        ]
    },
    "id": 1
}
```

Source

CUser::unblock() dans *ui/include/classes/api/services/CUser.php*.

user.update

Description

`object user.update(object/array users)`

Cette méthode vous permet de mettre à jour des utilisateurs existants.

Note:
Cette méthode est disponible pour les utilisateurs de tout type. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres des rôles utilisateur. Consultez Rôles utilisateur pour plus d'informations.

Note:
La robustesse du mot de passe de l'utilisateur est validée conformément aux règles de politique de mot de passe définies par l'API d'authentification. Consultez API d'authentification pour plus d'informations.

Paramètres

`(object/array)` Propriétés de l'utilisateur à mettre à jour.

La propriété `userid` doit être définie pour chaque utilisateur, toutes les autres propriétés sont facultatives. Seules les propriétés transmises seront mises à jour, toutes les autres resteront inchangées.

En plus des propriétés utilisateur standard, la méthode accepte les paramètres suivants.

Paramètre	Type	Description
current_passwd	string	Mot de passe actuel de l'utilisateur. La valeur de ce paramètre peut être une chaîne vide si l'utilisateur est lié à un répertoire d'utilisateurs. Comportement du paramètre: - *écriture seule* - *obligatoire* si `passwd` de l'objet User est défini et que l'utilisateur modifie son propre mot de passe
usrgrps	array	Groupes d'utilisateurs remplaçant les groupes d'utilisateurs existants. Pour les groupes d'utilisateurs, seule la propriété `usrgrpid` doit être définie.
medias	array	Médias utilisateur remplaçant les médias existants non provisionnés. Les médias provisionnés peuvent être omis lors de la mise à jour des médias.

Valeurs retournées

`(object)` Renvoie un objet contenant les ID des utilisateurs mis à jour sous la propriété `userids`.

Exemples

Renommer un utilisateur

Renommez un utilisateur en John Doe.

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "user.update",
    "params": {
        "userid": "1",
        "name": "John",
        "surname": "Doe"
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "userids": [
            "1"
        ]
    },
    "id": 1
}
```

Modification du rôle d'un utilisateur

Modifier le rôle d'un utilisateur.

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "user.update",
    "params": {
        "userid": "12",
        "roleid": "6"
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "userids": [
            "12"
        ]
    },
    "id": 1
}
```

Voir également

- Authentification

Source

CUser::update() dans *ui/include/classes/api/services/CUser.php*.

Vérification de découverte

Cette classe est conçue pour fonctionner avec les vérifications de découverte.

Références d'objet :

- Vérification de découverte

Méthodes disponibles :

- dcheck.get - récupérer les vérifications de découverte

Objet de vérification de découverte

Les objets suivants sont directement liés à l'API `dcheck`.

Vérification de découverte

L'objet de vérification de découverte définit une vérification spécifique effectuée par une règle de découverte réseau. Il possède les propriétés suivantes.

Propriété	Type	Description
dcheckid	ID	ID de la vérification de découverte.
druleid	ID	ID de la règle de découverte à laquelle la vérification appartient.
key_	string	Clé d'élément (si `type` est défini sur "Zabbix agent") ou OID SNMP (si `type` est défini sur "SNMPv1 agent", "SNMPv2 agent" ou "SNMPv3 agent"). Comportement de la propriété : - *obligatoire* si `type` est défini sur "Zabbix agent", "SNMPv1 agent", "SNMPv2 agent" ou "SNMPv3 agent"
ports	string	Une ou plusieurs plages de ports à vérifier, séparées par des virgules. Par défaut : 0. Comportement de la propriété : - *pris en charge* si `type` est défini sur "SSH" (0), "LDAP" (1), "SMTP" (2), "FTP" (3), "HTTP" (4), "POP" (5), "NNTP" (6), "IMAP" (7), "TCP" (8), "Zabbix agent" (9), "SNMPv1 agent" (10), "SNMPv2 agent" (11), "SNMPv3 agent" (13), "HTTPS" (14) ou "Telnet" (15)
snmp_community	string	Communauté SNMP. Comportement de la propriété : - *obligatoire* si `type` est défini sur "SNMPv1 agent" ou "SNMPv2 agent"
snmpv3_authpassphrase	string	Phrase secrète d'authentification. Comportement de la propriété : - *pris en charge* si `type` est défini sur "SNMPv3 agent" et `snmpv3_securitylevel` est défini sur "authNoPriv" ou "authPriv"
snmpv3_authprotocol	integer	Protocole d'authentification. Valeurs possibles : 0 - *(par défaut)* MD5; 1 - SHA1; 2 - SHA224; 3 - SHA256; 4 - SHA384; 5 - SHA512. Comportement de la propriété : - *pris en charge* si `type` est défini sur "SNMPv3 agent" et `snmpv3_securitylevel` est défini sur "authNoPriv" ou "authPriv"

Propriété	Type	Description
snmpv3_contextname	string	Nom de contexte SNMPv3. Comportement de la propriété : - *pris en charge* si `type` est défini sur ”SNMPv3 agent”
snmpv3_privpassphrase	string	Phrase secrète de confidentialité. Comportement de la propriété : - *pris en charge* si `type` est défini sur ”SNMPv3 agent” et `snmpv3_securitylevel` est défini sur ”authPriv”
snmpv3_privprotocol	integer	Protocole de confidentialité. Valeurs possibles : 0 - *(par défaut)* DES; 1 - AES128; 2 - AES192; 3 - AES256; 4 - AES192C; 5 - AES256C. Comportement de la propriété : - *pris en charge* si `type` est défini sur ”SNMPv3 agent” et `snmpv3_securitylevel` est défini sur ”authPriv”
snmpv3_securitylevel	string	Niveau de sécurité. Valeurs possibles : 0 - noAuthNoPriv; 1 - authNoPriv; 2 - authPriv. Comportement de la propriété : - *pris en charge* si `type` est défini sur ”SNMPv3 agent”
snmpv3_securityname	string	Nom de sécurité. Comportement de la propriété : - *pris en charge* si `type` est défini sur ”SNMPv3 agent”
type	integer	Type de vérification. Valeurs possibles : 0 - SSH; 1 - LDAP; 2 - SMTP; 3 - FTP; 4 - HTTP; 5 - POP; 6 - NNTP; 7 - IMAP; 8 - TCP; 9 - Zabbix agent; 10 - SNMPv1 agent; 11 - SNMPv2 agent; 12 - ICMP ping; 13 - SNMPv3 agent; 14 - HTTPS; 15 - Telnet. Comportement de la propriété : - *obligatoire*

Propriété	Type	Description
uniq	integer	Indique s'il faut utiliser cette vérification comme critère d'unicité du périphérique. Une seule vérification unique peut être configurée pour une règle de découverte. Valeurs possibles : 0 - *(par défaut)* ne pas utiliser cette vérification comme critère d'unicité; 1 - utiliser cette vérification comme critère d'unicité. Comportement de la propriété : - *pris en charge* si `type` est défini sur "Zabbix agent", "SNMPv1 agent", "SNMPv2 agent" ou "SNMPv3 agent"
host_source	integer	Source du nom d'hôte. Valeurs possibles : 1 - *(par défaut)* DNS; 2 - IP; 3 - valeur de découverte de cette vérification.
name_source	integer	Source du nom visible. Valeurs possibles : 0 - *(par défaut)* non spécifié; 1 - DNS; 2 - IP; 3 - valeur de découverte de cette vérification.
allow_redirect	integer	Autoriser la situation où la cible interrogée par ICMP ping répond depuis une adresse IP différente. Valeurs possibles : 0 - *(par défaut)* traiter les réponses redirigées comme si l'hôte cible était indisponible (échec); 1 - traiter les réponses redirigées comme si l'hôte cible était disponible (succès). Comportement de la propriété : - *pris en charge* si `type` est défini sur "ICMP ping"

dcheck.get

Description

`integer/array dcheck.get(object parameters)`

La méthode permet de récupérer les vérifications de découverte selon les paramètres fournis.

> **Note:**
> Cette méthode est disponible pour les utilisateurs de tout type. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres du rôle utilisateur. Consultez Rôles d'utilisateur pour plus d'informations.

Paramètres

`(object)` Paramètres définissant la sortie souhaitée.

La méthode prend en charge les paramètres suivants.

Paramètre	Type	Description
dcheckids	ID/array	Retourner uniquement les contrôles de découverte avec les ID donnés.
druleids	ID/array	Retourner uniquement les contrôles de découverte qui appartiennent aux règles de découverte données.
dserviceids	ID/array	Retourner uniquement les contrôles de découverte qui ont détecté les services découverts donnés.

Paramètre	Type	Description
selectDRules	query	Retourner les règles de découverte liées aux contrôles de découverte.
sortfield	string/array	Trier le résultat selon les propriétés données. Valeurs possibles : `dcheckid`, `druleid`.
countOutput	boolean	Ces paramètres sont décrits dans le commentaire de référence.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Valeurs retournées

`(integer/array)` Retourne soit :

- un tableau d'objets ;
- le nombre d'objets récupérés, si le paramètre `countOutput` a été utilisé.

Exemples

Récupérer les vérifications de découverte pour une règle de découverte

Récupérez toutes les vérifications de découverte utilisées par la règle de découverte « 6 ».

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "dcheck.get",
    "params": {
        "output": "extend",
        "dcheckids": "6"
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "dcheckid": "6",
            "druleid": "4",
            "type": "3",
            "key_": "",
            "snmp_community": "",
            "ports": "21",
            "snmpv3_securityname": "",
            "snmpv3_securitylevel": "0",
            "snmpv3_authpassphrase": "",
            "snmpv3_privpassphrase": "",
            "uniq": "0",
            "snmpv3_authprotocol": "0",
            "snmpv3_privprotocol": "0",
            "snmpv3_contextname": "",
            "host_source": "1",
            "name_source": "0",
            "allow_redirect": "0"
```

```
        }
    ],
    "id": 1
}
```

Source

CDCheck::get() dans *ui/include/classes/api/services/CDCheck.php*.

Élément

Cette classe est conçue pour fonctionner avec des éléments.

Références d'objet :

- Élément
  - En-tête HTTP
  - Champ de requête HTTP
- Tag d'élément
- Prétraitement d'élément

Méthodes disponibles :

- item.create - créer de nouveaux éléments
- item.delete - supprimer des éléments
- item.get - récupérer des éléments
- item.update - mettre à jour des éléments

Objet item

Les objets suivants sont directement liés à l'API `item`.

Élément

> **Note:**
> Les éléments web ne peuvent pas être créés, mis à jour ou supprimés directement via l'API Zabbix.

L'objet élément possède les propriétés suivantes.

Property	Type	Description
itemid	ID	ID de l'élément. Comportement de la propriété : - *en lecture seule* - *obligatoire* pour les opérations de mise à jour

Property	Type	Description
delay	string	Intervalle de mise à jour de l'élément. Accepte un nombre de secondes (par exemple, 1, 60, 300), une valeur avec un suffixe temporel (par exemple, 30s, 1m, 2h, 1d) ou, facultativement, un ou plusieurs intervalles personnalisés, tous séparés par des points-virgules. Les intervalles personnalisés peuvent combiner des intervalles flexibles et des intervalles de planification. Accepte les macros utilisateur. Si elle est utilisée, la valeur doit être une macro unique. Plusieurs macros ou des macros combinées à du texte ne sont pas pris en charge. Les intervalles flexibles peuvent être écrits sous la forme de deux macros séparées par une barre oblique (par exemple, {$FLEX_INTERVAL}/{$FLEX_PERIOD}). Exemple : 1h;wd1-5h9-18;{$Macro1}/1-7,00:00-24:00;0/6-7,12:00-24:00;{$Macr Comportement de la propriété : - *obligatoire* si type est défini sur « Zabbix agent » (0), « Simple check » (3), « Zabbix internal » (5), « External check » (10), « Database monitor » (11), « IPMI agent » (12), « SSH agent » (13), « TELNET agent » (14), « Calculated » (15), « JMX agent » (16), « HTTP agent » (19), « SNMP agent » (20), « Script » (21), « Browser » (22), « Telemetry query » (24), ou si type est défini sur « Zabbix agent (active) » (7) et que key_ ne contient pas « mqtt.get »
hostid	ID	ID de l'hôte ou du modèle auquel appartient l'élément. Comportement de la propriété : - *constant* - *obligatoire* pour les opérations de création
interfaceid	ID	ID de l'interface de l'hôte de l'élément. Comportement de la propriété : - *obligatoire* si l'élément appartient à un hôte et que type est défini sur « Zabbix agent », « IPMI agent », « JMX agent », « SNMP trap » ou « SNMP agent » - *pris en charge* si l'élément appartient à un hôte et que type est défini sur « Simple check », « External check », « SSH agent », « TELNET agent » ou « HTTP agent » - *en lecture seule* pour les objets découverts
key_	string	Clé de l'élément. Comportement de la propriété : - *obligatoire* pour les opérations de création - *en lecture seule* pour les objets hérités ou découverts
name	string	Nom de l'élément. Prend en charge les macros utilisateur. Comportement de la propriété : - *obligatoire* pour les opérations de création - *en lecture seule* pour les objets hérités ou découverts
name_resolved	string	Nom de l'élément avec les macros utilisateur résolues. Comportement de la propriété : - *en lecture seule*

Property	Type	Description
type	integer	Type de l'élément. Valeurs possibles : 0 - Zabbix agent ; 2 - Zabbix trapper ; 3 - Simple check ; 5 - Zabbix internal ; 7 - Zabbix agent (active) ; 9 - Web item ; 10 - External check ; 11 - Database monitor ; 12 - IPMI agent ; 13 - SSH agent ; 14 - TELNET agent ; 15 - Calculated ; 16 - JMX agent ; 17 - SNMP trap ; 18 - Dependent item ; 19 - HTTP agent ; 20 - SNMP agent ; 21 - Script ; 22 - Browser ; 24 - Telemetry query. Comportement de la propriété : - *obligatoire* pour les opérations de création - *en lecture seule* pour les objets hérités ou découverts
url	string	Chaîne URL. Prend en charge les macros utilisateur, {HOST.IP}, {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.NAME}, {HOST.PORT}, {ITEM.ID}, {ITEM.KEY}. Comportement de la propriété : - *obligatoire* si `type` est défini sur « HTTP agent » - *en lecture seule* pour les objets hérités ou découverts
value_type	integer	Type d'information de l'élément. Valeurs possibles : 0 - numérique (flottant) ; 1 - caractère ; 2 - journal ; 3 - numérique (non signé) ; 4 - texte ; 5 - binaire ; 6 - JSON. Comportement de la propriété : - *obligatoire* pour les opérations de création - *en lecture seule* pour les objets hérités ou découverts
allow_traps	integer	Autorise le remplissage de la valeur de la même manière que pour un élément trapper. 0 - *(par défaut)* Ne pas accepter les données entrantes ; 1 - Accepter les données entrantes. Comportement de la propriété : - *pris en charge* si `type` est défini sur « HTTP agent » - *en lecture seule* pour les objets découverts

Property	Type	Description
authtype	integer	Méthode d'authentification. Valeurs possibles si `type` est défini sur « SSH agent » : 0 - *(par défaut)* mot de passe ; 1 - clé publique. Valeurs possibles si `type` est défini sur « HTTP agent » : 0 - *(par défaut)* aucune ; 1 - basic ; 2 - NTLM ; 3 - Kerberos ; 4 - Digest. Comportement de la propriété : - *pris en charge* si `type` est défini sur « SSH agent » ou « HTTP agent » - *en lecture seule* pour les objets hérités (si `type` est défini sur « HTTP agent ») ou les objets découverts
description	string	Description de l'élément. Comportement de la propriété : - *en lecture seule* pour les objets découverts
error	string	Texte de l'erreur en cas de problème lors de la mise à jour de la valeur de l'élément. Comportement de la propriété : - *en lecture seule*
flags	integer	Origine de l'élément. Valeurs possibles : 0 - élément standard ; 4 - élément converti à partir d'un prototype. Comportement de la propriété : - *en lecture seule*
follow_redirects	integer	Suit les redirections de la réponse lors de l'interrogation des données. Valeurs possibles : 0 - Ne pas suivre les redirections ; 1 - *(par défaut)* Suivre les redirections. Comportement de la propriété : - *pris en charge* si `type` est défini sur « HTTP agent » - *en lecture seule* pour les objets hérités ou découverts
headers	array	Tableau d'en-têtes envoyés lors de l'exécution d'une requête HTTP. Comportement de la propriété : - *pris en charge* si `type` est défini sur « HTTP agent » - *en lecture seule* pour les objets hérités ou découverts
history	string	Unité de temps pendant laquelle les données d'historique doivent être conservées. Accepte également une macro utilisateur. Valeur par défaut : `31d`. Comportement de la propriété : - *en lecture seule* pour les objets découverts
http_proxy	string	Chaîne de connexion du proxy HTTP(S). Comportement de la propriété : - *pris en charge* si `type` est défini sur « HTTP agent » - *en lecture seule* pour les objets hérités ou découverts

Property	Type	Description
inventory_link	integer	ID du champ d'inventaire de l'hôte alimenté par l'élément. Consultez la page d'inventaire de l'hôte pour obtenir la liste des champs d'inventaire de l'hôte pris en charge et leurs ID. Valeur par défaut : 0. Comportement de la propriété : - *pris en charge* si `value_type` est défini sur « numeric float », « character », « numeric unsigned » ou « text » - *en lecture seule* pour les objets découverts
ipmi_sensor	string	Capteur IPMI. Comportement de la propriété : - *obligatoire* si `type` est défini sur « IPMI agent » et que `key_` n'est pas défini sur « ipmi.get » - *pris en charge* si `type` est défini sur « IPMI agent » - *en lecture seule* pour les objets hérités ou découverts
jmx_endpoint	string	Chaîne de connexion personnalisée de l'agent JMX. Valeur par défaut : service:jmx:rmi:///jndi/rmi://{HOST.CONN}:{HOST.PORT}/jmxrmi Comportement de la propriété : - *pris en charge* si `type` est défini sur « JMX agent » - *en lecture seule* pour les objets découverts
lastclock	timestamp	Moment de la dernière mise à jour de la valeur de l'élément. Par défaut, seules les valeurs datant de moins de 24 heures sont affichées. Vous pouvez prolonger cette période en modifiant la valeur du paramètre *Max history display period* dans la section de menu *Administration > General*. Comportement de la propriété : - *en lecture seule*
lastns	integer	Nanosecondes écoulées depuis la dernière mise à jour de la valeur de l'élément. Par défaut, seules les valeurs datant de moins de 24 heures sont affichées. Vous pouvez prolonger cette période en modifiant la valeur du paramètre *Max history display period* dans la section de menu *Administration > General*. Comportement de la propriété : - *en lecture seule*
lastvalue	string	Dernière valeur de l'élément. Par défaut, seules les valeurs datant de moins de 24 heures sont affichées. Vous pouvez prolonger cette période en modifiant la valeur du paramètre *Max history display period* dans la section de menu *Administration > General*. Comportement de la propriété : - *en lecture seule*
logtimefmt	string	Format de l'heure dans les entrées de journal. Comportement de la propriété : - *pris en charge* si `value_type` est défini sur « log » - *en lecture seule* pour les objets hérités ou découverts

Property	Type	Description
master_itemid	ID	ID de l'élément maître. Comportement de la propriété : - *obligatoire* si `type` est défini sur « Dependent item » - *en lecture seule* pour les objets hérités ou découverts
output_format	integer	Indique si la réponse doit être convertie en JSON. 0 - *(par défaut)* Stocker les données brutes ; 1 - Convertir en JSON. Comportement de la propriété : - *pris en charge* si `type` est défini sur « HTTP agent » - *en lecture seule* pour les objets hérités ou découverts
params	string	Paramètres supplémentaires selon le type d'élément : - script exécuté pour les éléments SSH agent et TELNET agent ; - requête SQL pour les éléments Database monitor ; - formule pour les éléments Calculated ; - script pour les éléments Script et Browser. Comportement de la propriété : - *obligatoire* si `type` est défini sur « Database monitor », « SSH agent », « TELNET agent », « Calculated », « Script » ou « Browser » - *en lecture seule* pour les objets hérités (si `type` est défini sur « Script » ou « Browser ») ou les objets découverts
parameters	object/array	Paramètres supplémentaires si `type` est défini sur « Script » ou « Browser ». Tableau d'objets possédant les propriétés `name` et `value`, où `name` doit être unique. Comportement de la propriété : - *pris en charge* si `type` est défini sur « Script » ou « Browser » - *en lecture seule* pour les objets hérités ou découverts
password	string	Mot de passe d'authentification. Comportement de la propriété : - *obligatoire* si `type` est défini sur « JMX agent » et que `username` est défini - *pris en charge* si `type` est défini sur « Simple check », « SSH agent », « TELNET agent », « Database monitor » ou « HTTP agent » - *en lecture seule* pour les objets hérités (si `type` est défini sur « HTTP agent ») ou les objets découverts
post_type	integer	Type de corps des données POST stocké dans la propriété `posts`. Valeurs possibles : 0 - *(par défaut)* Données brutes ; 2 - Données JSON ; 3 - Données XML. Comportement de la propriété : - *pris en charge* si `type` est défini sur « HTTP agent » - *en lecture seule* pour les objets hérités ou découverts
posts	string	Données du corps de la requête HTTP(S). Comportement de la propriété : - *obligatoire* si `type` est défini sur « HTTP agent » et que `post_type` est défini sur « JSON data » ou « XML data » - *pris en charge* si `type` est défini sur « HTTP agent » et que `post_type` est défini sur « Raw data » - *en lecture seule* pour les objets hérités ou découverts

Property	Type	Description
prevvalue	string	Valeur précédente de l'élément. Par défaut, seules les valeurs datant de moins de 24 heures sont affichées. Vous pouvez prolonger cette période en modifiant la valeur du paramètre *Max history display period* dans la section de menu *Administration > General*. Comportement de la propriété : - *en lecture seule*
privatekey	string	Nom du fichier de clé privée. Comportement de la propriété : - *obligatoire* si `type` est défini sur « SSH agent » et que `authtype` est défini sur « public key » - *en lecture seule* pour les objets découverts
publickey	string	Nom du fichier de clé publique. Comportement de la propriété : - *obligatoire* si `type` est défini sur « SSH agent » et que `authtype` est défini sur « public key » - *en lecture seule* pour les objets découverts
query_fields	array	Tableau de champs de requête envoyés lors de l'exécution d'une requête HTTP. Comportement de la propriété : - *pris en charge* si `type` est défini sur « HTTP agent » - *en lecture seule* pour les objets hérités ou découverts
request_method	integer	Type de méthode de requête. Valeurs possibles : 0 - *(par défaut)* GET ; 1 - POST ; 2 - PUT ; 3 - HEAD. Comportement de la propriété : - *pris en charge* si `type` est défini sur « HTTP agent » - *en lecture seule* pour les objets hérités ou découverts
retrieve_mode	integer	Partie de la réponse à stocker. Valeurs possibles si `request_method` est défini sur « GET », « POST » ou « PUT » : 0 - *(par défaut)* Corps ; 1 - En-têtes ; 2 - Le corps et les en-têtes seront stockés. Valeurs possibles si `request_method` est défini sur « HEAD » : 1 - En-têtes. Comportement de la propriété : - *pris en charge* si `type` est défini sur « HTTP agent » - *en lecture seule* pour les objets hérités ou découverts
snmp_oid	string	OID SNMP. Comportement de la propriété : - *obligatoire* si `type` est défini sur « SNMP agent » - *en lecture seule* pour les objets hérités ou découverts
ssl_cert_file	string	Chemin du fichier de clé SSL publique. Comportement de la propriété : - *pris en charge* si `type` est défini sur « HTTP agent » - *en lecture seule* pour les objets hérités ou découverts

Property	Type	Description
ssl_key_file	string	Chemin du fichier de clé SSL privée. Comportement de la propriété : - *pris en charge* si `type` est défini sur « HTTP agent » - *en lecture seule* pour les objets hérités ou découverts
ssl_key_password	string	Mot de passe du fichier de clé SSL. Comportement de la propriété : - *pris en charge* si `type` est défini sur « HTTP agent » - *en lecture seule* pour les objets hérités ou découverts
state	integer	État de l'élément. Valeurs possibles : 0 - *(par défaut)* normal ; 1 - non pris en charge. Comportement de la propriété : - *en lecture seule*
status	integer	Statut de l'élément. Valeurs possibles : 0 - *(par défaut)* élément activé ; 1 - élément désactivé.
status_codes	string	Plages de codes d'état HTTP requis, séparées par des virgules. Prend également en charge les macros utilisateur dans une liste séparée par des virgules. Exemple : 200,200-{$M},{$M},200-400 Comportement de la propriété : - *pris en charge* si `type` est défini sur « HTTP agent » - *en lecture seule* pour les objets hérités ou découverts
templateid	ID	ID de l'élément du modèle parent. *Conseil* : utilisez la propriété `hostid` pour spécifier le modèle auquel appartient l'élément. Comportement de la propriété : - *en lecture seule*
timeout	string	Délai d'expiration de la requête d'interrogation des données de l'élément. Accepte un nombre de secondes (par exemple, 1, 60, 300) ou une valeur avec un suffixe temporel (par exemple, 300s, 1m). Accepte également les macros utilisateur. Plage de valeurs possibles : 1-600s. Valeur par défaut : "" - utiliser les paramètres du proxy ou les paramètres globaux. Comportement de la propriété : - *pris en charge* si `type` est défini sur « Zabbix agent » (0), « Simple check » (3) et que `key_` ne commence pas par « `vmware.` » ni « `icmpping` », « Zabbix agent (active) » (7), « External check » (10), « Database monitor » (11), « SSH agent » (13), « TELNET agent » (14), « HTTP agent » (19), « SNMP agent » (20) et que `snmp_oid` commence par « `walk[` » ou « `get[` », « Script » (21), « Browser » (22) - *en lecture seule* pour les objets hérités et découverts, - si la valeur par défaut est appliquée et que l'élément appartient à un hôte surveillé par un proxy ou un groupe de proxy inaccessible, le délai d'expiration global est utilisé.

Property	Type	Description
trapper_hosts	string	Hôtes autorisés. Comportement de la propriété : - *en lecture seule* pour les objets découverts - *pris en charge* si `type` est défini sur « Zabbix trapper », ou si `type` est défini sur « HTTP agent » et que `allow_traps` est défini sur « Allow incoming data ». Si la propriété est omise, elle est automatiquement définie sur la macro utilisateur `{$TRAPPER.ALLOWED_HOSTS}` (valeur par défaut). Par défaut, `{$TRAPPER.ALLOWED_HOSTS}` vaut `127.0.0.1,::1`. Si la propriété est définie sur une chaîne vide, les connexions entrantes de tous les hôtes seront refusées.
trends	string	Unité de temps pendant laquelle les données de tendance doivent être conservées. Accepte également une macro utilisateur. Valeur par défaut : `365d`. Comportement de la propriété : - *pris en charge* si `value_type` est défini sur « numeric float » ou « numeric unsigned » - *en lecture seule* pour les objets découverts
units	string	Unités de la valeur. Comportement de la propriété : - *pris en charge* si `value_type` est défini sur « numeric float » ou « numeric unsigned » - *en lecture seule* pour les objets hérités ou découverts
username	string	Nom d'utilisateur pour l'authentification. Comportement de la propriété : - *obligatoire* si `type` est défini sur « SSH agent », « TELNET agent », ou si `type` est défini sur « JMX agent » et que `password` est défini - *pris en charge* si `type` est défini sur « Simple check », « Database monitor » ou « HTTP agent » - *en lecture seule* pour les objets hérités (si `type` est défini sur « HTTP agent ») ou les objets découverts
uuid	string	Identifiant unique universel utilisé pour associer un élément importé à des éléments déjà existants. Généré automatiquement s'il n'est pas fourni. Comportement de la propriété : - *pris en charge* si l'élément appartient à un modèle
valuemapid	ID	ID de la table de correspondance des valeurs associée. Comportement de la propriété : - *pris en charge* si `value_type` est défini sur « numeric float », « character » ou « numeric unsigned » - *en lecture seule* pour les objets hérités ou découverts
verify_host	integer	Indique s'il faut vérifier que le nom d'hôte de la connexion correspond à celui du certificat de l'hôte. Valeurs possibles : 0 - *(par défaut)* Ne pas vérifier ; 1 - Vérifier. Comportement de la propriété : - *pris en charge* si `type` est défini sur « HTTP agent » - *en lecture seule* pour les objets hérités ou découverts

Property	Type	Description
verify_peer	integer	Indique s'il faut vérifier que le certificat de l'hôte est authentique. Valeurs possibles : 0 - *(par défaut)* Ne pas vérifier ; 1 - Vérifier. Comportement de la propriété : - *pris en charge* si `type` est défini sur « HTTP agent » - *en lecture seule* pour les objets hérités ou découverts
query	object	Configuration de la requête de télémétrie. Définit ce que l'élément interroge dans la source de données de télémétrie, la manière dont il filtre les données et la manière dont il agrège le résultat. Comportement de la propriété : - *obligatoire* si `type` est défini sur « Telemetry query » - *en lecture seule* pour les objets hérités ou découverts
time_shift	string	Décalage vers le passé (avant l'heure actuelle) de la fin de la plage temporelle interrogée par l'élément. Accepte un nombre de secondes (par exemple, `1`, `60`, `300`) ou une valeur avec un suffixe temporel (par exemple, `30s`, `1m`, `2h`, `1d`). Accepte également les macros utilisateur. Plage de valeurs possibles : 0-86400s. Valeur par défaut : 15s. Comportement de la propriété : - *pris en charge* si `type` est défini sur « Telemetry query »
lookback_limit	string	Décalage vers le passé (avant l'heure actuelle) jusqu'auquel l'élément recherche des données lorsqu'il n'existe aucune requête précédente à poursuivre (par exemple, lors de la première exécution de l'élément), ou lorsque la requête précédente est antérieure à `delay` (par exemple, parce que le serveur Zabbix était arrêté). Accepte un nombre de secondes (par exemple, `1`, `60`, `300`) ou une valeur avec un suffixe temporel (par exemple, `30s`, `1m`, `2h`, `1d`). Accepte également les macros utilisateur. Plage de valeurs possibles : 1-259200s. Valeur par défaut : 10m. Comportement de la propriété : - *pris en charge* si `type` est défini sur « Telemetry query »
granularity	string	Durée d'un segment en lequel la plage temporelle complète de la requête est divisée. Accepte un nombre de secondes (par exemple, `1`, `60`, `300`) ou une valeur avec un suffixe temporel (par exemple, `30s`, `1m`, `2h`, `1d`). Accepte également les macros utilisateur. Plage de valeurs possibles : 1-86400s. Valeur par défaut : 15s. Comportement de la propriété : - *pris en charge* si `type` est défini sur « Telemetry query »

En-tête HTTP

L’objet d’en-tête possède les propriétés suivantes :

Propriété	Type	Description
name	string	Nom de l’en-tête HTTP. Comportement de la propriété : - *obligatoire*
value	string	Valeur de l’en-tête. Comportement de la propriété : - *obligatoire*

Champ de requête HTTP

L’objet de champ de requête définit un nom et une valeur utilisés pour spécifier un paramètre d’URL. Il possède les propriétés suivantes :

Propriété	Type	Description
name	string	Nom du paramètre. Comportement de la propriété : - *obligatoire*
value	string	Valeur du paramètre. Comportement de la propriété : - *obligatoire*

Requête de télémétrie

L’objet de requête de télémétrie possède les propriétés suivantes.

Property	Type	Description
signal_type	integer	Type de signal interrogé par l’élément. Valeurs possibles : 0 - Traces ; 1 - Métriques ; 2 - Journaux. Comportement de la propriété : - *required*
metric_point_type	integer	Type de point de métrique interrogé par l’élément. Valeurs possibles : 0 - Somme ; 1 - Jauge ; 2 - Histogramme ; 3 - Histogramme exponentiel. Comportement de la propriété : - *required* si `signal_type` est défini sur « Metrics »

Property	Type	Description
columns	array	Tableau de colonnes de données de télémétrie interrogées par l'élément. Peut être vide. Comportement de la propriété : - *required*
aggregated_columns	array	Tableau de colonnes agrégées de données de télémétrie interrogées par l'élément, ainsi que la manière dont il calcule une valeur à partir des enregistrements de colonnes. Comportement de la propriété : - *required*
filter	object	Filtre de requête de télémétrie pour les enregistrements de colonnes de la requête. Comportement de la propriété : - *required*

Colonnes de données de télémétrie

L'objet des colonnes de données de télémétrie possède les propriétés suivantes.

Property	Type	Description
column	string	Nom de la colonne de données de télémétrie interrogée par l'élément. Voir les colonnes de données de télémétrie prises en charge. Pour plusieurs colonnes à valeur unique, chaque valeur de `column` doit être unique. Pour plusieurs colonnes contenant des paires clé-valeur, chaque paire `column` et `attribute_key` doit être unique.
attribute_key	string	Nom de la clé interrogée par l'élément dans la colonne contenant des paires clé-valeur. Pour plusieurs colonnes contenant des paires clé-valeur, chaque paire `column` et `attribute_key` doit être unique. Comportement de la propriété : - *required* si `column` est défini sur une colonne contenant des paires clé-valeur

Colonnes agrégées des données de télémétrie

L'objet des colonnes agrégées des données de télémétrie possède les propriétés suivantes.

Property	Type	Description
column	string	Nom de la colonne de données de télémétrie interrogée par l'élément. Voir colonnes de données de télémétrie prises en charge. Pour plusieurs colonnes, chaque valeur de `column` doit être unique. Cette propriété n'est pas prise en charge si `function` est défini sur "Count". Comportement de la propriété: - *obligatoire* si `function` est défini sur "Min", "Max", "Average", "Sum" ou "Percentile"

Property	Type	Description
function	integer	Fonction appliquée par l'élément aux enregistrements de la colonne. Valeurs possibles : 1 - Min ; 2 - Max ; 3 - Average ; 4 - Count ; 5 - Sum ; 8 - Percentile. Comportement de la propriété: - *obligatoire*
parameters	array	Percentile calculé par l'élément à partir des enregistrements de la colonne. Valeur possible : un nombre de 0 à 100, fourni sous forme de chaîne de caractères. Comportement de la propriété: - *obligatoire* si `function` est défini sur "Percentile"
alias	string	Nom unique de la valeur calculée, utilisé pour l'identifier dans la valeur de l'élément. Chaque valeur de `alias` doit être unique. Comportement de la propriété: - *obligatoire*

Filtre de données de télémétrie

L'objet de filtre de données de télémétrie possède les propriétés suivantes.

Property	Type	Description
conditions	array	Conditions du filtre de données de télémétrie auxquelles un enregistrement d'une colonne de données de télémétrie doit satisfaire. Comportement de la propriété : - *required*
evaltype	integer	Méthode d'évaluation des conditions du filtre. Valeurs possibles : 0 - And/Or ; 1 - And ; 2 - Or ; 3 - Expression personnalisée. Comportement de la propriété : - *read-only*
eval_formula	string	Expression générée utilisée pour évaluer les conditions du filtre. L'expression contient des ID qui font référence à des conditions de filtre spécifiques via leur `formulaid`. La valeur de `eval_formula` est égale à la valeur de `formula` pour les filtres utilisant une expression personnalisée. Comportement de la propriété : - *read-only*

Property	Type	Description
formula	string	Expression définie par l'utilisateur utilisée pour évaluer les conditions des filtres utilisant une expression personnalisée. L'expression doit contenir des ID qui font référence à des conditions de filtre spécifiques via leur `formulaid`. Les ID utilisés dans l'expression doivent correspondre exactement à ceux définis dans les conditions du filtre : aucune condition ne peut rester inutilisée ou être omise. Comportement de la propriété : - *required* si `evaltype` est défini sur « Expression personnalisée »

Conditions de filtrage des données de télémétrie

L'objet des conditions de filtrage des données de télémétrie possède les propriétés suivantes.

Property	Type	Description
column	string	Nom de la colonne de données de télémétrie à laquelle la condition de filtrage s'applique. Voir les colonnes de données de télémétrie prises en charge. Comportement de la propriété : - *obligatoire*
attribute_key	string	Nom de la clé à laquelle la condition s'applique, dans la colonne contenant des paires clé-valeur. Comportement de la propriété : - *obligatoire* si `column` est défini sur une colonne contenant des paires clé-valeur
operator	integer	Opérateur de la condition de filtrage. Valeurs possibles pour les colonnes à valeur unique : 0 - Est égal à ; 1 - N'est pas égal à ; 2 - Contient ; 3 - Ne contient pas. Valeurs possibles pour les colonnes contenant des paires clé-valeur : 0 - Est égal à ; 1 - N'est pas égal à ; 12 - Existe. Comportement de la propriété : - *obligatoire*
value	string	Valeur à comparer. Comportement de la propriété : - *obligatoire* si `operator` est défini sur « Est égal à », « N'est pas égal à », « Contient » ou « Ne contient pas »
formulaid	string	ID unique arbitraire utilisé pour référencer la condition depuis une expression personnalisée. Ne peut contenir que des lettres majuscules. L'ID doit être défini par l'utilisateur lors de la modification des conditions de filtrage, mais sera généré à nouveau lors de leur demande ultérieure. Comportement de la propriété : - *obligatoire* si `evaltype` de l'objet Filtre de requête de télémétrie est défini sur « Expression personnalisée »

Balise d'élément

L'objet de balise d'élément possède les propriétés suivantes.

Propriété	Type	Description
tag	string	Nom de la balise d'élément. Comportement de la propriété : - *obligatoire*
value	string	Valeur de la balise d'élément. Comportement de la propriété : - *lecture seule*.
object	integer	Type d'objet dont la balise a été héritée. Valeurs possibles : 0 - modèle; 1 - hôte. Comportement de la propriété : - *lecture seule*.
objectid	ID	ID de l'objet dont la balise a été héritée. Comportement de la propriété : - *lecture seule*.

Prétraitement de l'élément

L'objet de prétraitement de l'élément possède les propriétés suivantes.

Propriété	Type	Description
type	integer	Type d'option de prétraitement. Valeurs possibles : 1 - Multiplicateur personnalisé ; 2 - Suppression des espaces à droite ; 3 - Suppression des espaces à gauche ; 4 - Suppression des espaces ; 5 - Expression régulière ; 6 - Conversion de booléen en décimal ; 7 - Conversion d'octal en décimal ; 8 - Conversion d'hexadécimal en décimal ; 9 - Modification simple ; 10 - Modification par seconde ; 11 - XPath XML ; 12 - JSONPath ; 13 - Dans l'intervalle ; 14 - Correspond à une expression régulière ; 15 - Ne correspond pas à une expression régulière ; 16 - Vérification des erreurs dans JSON ; 17 - Vérification des erreurs dans XML ; 18 - Vérification des erreurs à l'aide d'une expression régulière ; 19 - Ignorer les valeurs inchangées ; 20 - Ignorer les valeurs inchangées avec pulsation ; 21 - JavaScript ; 22 - Modèle Prometheus ; 23 - Conversion de Prometheus en JSON ; 24 - Conversion de CSV en JSON ; 25 - Remplacement ; 26 - Vérification de la non-prise en charge ; 27 - Conversion de XML en JSON ; 28 - Valeur de parcours SNMP ; 29 - Conversion d'un parcours SNMP en JSON ; 30 - Valeur de récupération SNMP. Comportement de la propriété : - *required*
params	string	Paramètres supplémentaires utilisés par l'option de prétraitement. Plusieurs paramètres sont séparés par le caractère de nouvelle ligne (\n). Si `type` est défini sur « Vérification de la non-prise en charge », les paramètres suivent la syntaxe `<scope>[\n<pattern>]`, où *pattern* est une expression régulière et *scope* est l'une des valeurs suivantes : -1 - correspond à toute erreur ; 0 - vérifie si le message d'erreur correspond à *pattern* ; 1 - vérifie si le message d'erreur ne correspond pas à *pattern*. Comportement de la propriété : - *required* si `type` est défini sur « Multiplicateur personnalisé » (1), « Suppression des espaces à droite » (2), « Suppression des espaces à gauche » (3), « Suppression des espaces » (4), « Expression régulière » (5), « XPath XML » (11), « JSONPath » (12), « Dans l'intervalle » (13), « Correspond à une expression régulière » (14), « Ne correspond pas à une expression régulière » (15), « Vérification des erreurs dans JSON » (16), « Vérification des erreurs dans XML » (17), « Vérification des erreurs à l'aide d'une expression régulière » (18), « Ignorer les valeurs inchangées avec pulsation » (20), « JavaScript » (21), « Modèle Prometheus » (22), « Conversion de Prometheus en JSON » (23), « Conversion de CSV en JSON » (24), « Remplacement » (25), Vérification de la non-prise en charge (26), « Valeur de parcours SNMP » (28), « Conversion d'un parcours SNMP en JSON » (29) ou « Valeur de récupération SNMP » (30)

Propriété	Type	Description
error_handler	integer	Type d'action utilisé en cas d'échec d'une étape de prétraitement. Valeurs possibles : 0 - Le message d'erreur est défini par le serveur Zabbix ; 1 - Ignorer la valeur ; 2 - Définir une valeur personnalisée ; 3 - Définir un message d'erreur personnalisé. Valeurs possibles si `type` est défini sur « Vérification de la non-prise en charge » : 1 - Ignorer la valeur ; 2 - Définir une valeur personnalisée ; 3 - Définir un message d'erreur personnalisé. Comportement de la propriété : - *required* si `type` est défini sur « Multiplicateur personnalisé » (1), « Expression régulière » (5), « Conversion de booléen en décimal » (6), « Conversion d'octal en décimal » (7), « Conversion d'hexadécimal en décimal » (8), « Modification simple » (9), « Modification par seconde » (10), « XPath XML » (11), « JSONPath » (12), « Dans l'intervalle » (13), « Correspond à une expression régulière » (14), « Ne correspond pas à une expression régulière » (15), « Vérification des erreurs dans JSON » (16), « Vérification des erreurs dans XML » (17), « Vérification des erreurs à l'aide d'une expression régulière » (18), « Modèle Prometheus » (22), « Conversion de Prometheus en JSON » (23), « Conversion de CSV en JSON » (24), « Vérification de la non-prise en charge » (26), « Conversion de XML en JSON » (27), « Valeur de parcours SNMP » (28), « Conversion d'un parcours SNMP en JSON » (29) ou « Valeur de récupération SNMP » (30)
error_handler_params	string	Paramètres du gestionnaire d'erreurs. Comportement de la propriété : - *required* si `error_handler` est défini sur « Définir une valeur personnalisée » ou « Définir un message d'erreur personnalisé »

Les paramètres et gestionnaires d'erreurs suivants sont pris en charge pour chaque type de prétraitement.

Type de prétraitement	Nom	Paramètre 1	Paramètre 2	Paramètre 3	Gestionnaires d'erreurs pris en charge
1	Multiplicateur personnalisé	nombre^1, 6^			0, 1, 2, 3
2	Suppression des espaces à droite	liste de caractères[2]			
3	Suppression des espaces à gauche	liste de caractères[2]			
4	Suppression des espaces	liste de caractères[2]			

Type de prétraitement	Nom	Paramètre 1	Paramètre 2	Paramètre 3	Gestionnaires d’erreurs pris en charge
5	Expression régulière	modèle[3]	sortie[2]		0, 1, 2, 3
6	Conversion de booléen en déci-mal				0, 1, 2, 3
7	Conversion d’octal en déci-mal				0, 1, 2, 3
8	Conversion d’hexadécimal en déci-mal				0, 1, 2, 3
9	Modification sim-ple				0, 1, 2, 3
10	Modification par sec-onde				0, 1, 2, 3
11	XPath XML	chemin[4]			0, 1, 2, 3
12	JSONPath	chemin[4]			0, 1, 2, 3
13	Dans l’intervalle	min^1, 6^	max^1, 6^		0, 1, 2, 3
14	Corresponddà une ex-pres-sion régulière	modèle[3]			0, 1, 2, 3
15	Ne cor-re-spond pas à une ex-pres-sion régulière	modèle[3]			0, 1, 2, 3
16	Vérification des er-reurs dans JSON	chemin[4]			0, 1, 2, 3
17	Vérification des er-reurs dans XML	chemin[4]			0, 1, 2, 3

Type de prétraitement	Nom	Paramètre 1	Paramètre 2	Paramètre 3	Gestionnaires d'erreurs pris en charge
18	Vérification des erreurs à l'aide d'une expression régulière	modèle[3]	sortie[2]		0, 1, 2, 3
19	Ignorer les valeurs inchangées				
20	Ignorer les valeurs inchangées avec pulsation	secondes^5, 6^			
21	JavaScript	script[2]			
22	Modèle Prometheus	modèle^6, 7^	`value`, `label`, `function`	sortie^8, 9^	0, 1, 2, 3
23	Conversion de Prometheus en JSON	modèle^6, 7^			0, 1, 2, 3
24	Conversion de CSV en JSON	caractère[2]	caractère[2]	0,1	0, 1, 2, 3
25	Remplacement	chaîne à rechercher[2]	remplacement[2]		
26	Vérification de la non-prise en charge	scope[1]	modèle^3, 6^		1, 2, 3
27	Conversion de XML en JSON				0, 1, 2, 3
28	Valeur de parcours SNMP	OID[2]	Format : 0 - Inchangé 1 - UTF-8 depuis Hex-STRING 2 - MAC depuis Hex-STRING 3 - Entier depuis BITS		0, 1, 2, 3

Type de prétraitement	Nom	Paramètre 1	Paramètre 2	Paramètre 3	Gestionnaires d'erreurs pris en charge
29	Conversion d'un parcours SNMP en JSON[10]	Nom du champ[2]	Préfixe OID[2]	Format : 0 - Inchangé 1 - UTF-8 depuis Hex-STRING 2 - MAC depuis Hex-STRING 3 - Entier depuis BITS	0, 1, 2, 3
30	Valeur de récupération SNMP	Format : 1 - UTF-8 depuis Hex-STRING 2 - MAC depuis Hex-STRING 3 - Entier depuis BITS			0, 1, 2, 3

[1] nombre à virgule flottante (les entiers sont implicitement convertis en valeurs à virgule flottante)
 [2] chaîne
 [3] expression régulière
 [4] JSONPath ou XPath XML
 [5] entier positif (avec prise en charge des suffixes temporels, par exemple 30s, 1m, 2h, 1d)
 [6] macro utilisateur
 [7] modèle Prometheus suivant la syntaxe : `<metric name>{<label name>="<label value>", ...} == <value>`. Chaque composant du modèle Prometheus (métrique, nom d'étiquette, valeur d'étiquette et valeur de métrique) peut être une macro utilisateur.
 [8] sortie Prometheus suivant la syntaxe : `<label name>` (peut être une macro utilisateur) si `label` est sélectionné comme deuxième paramètre.
 [9] L'une des fonctions d'agrégation suivantes : `sum`, `min`, `max`, `avg`, `count` si `function` est sélectionné comme deuxième paramètre.
 [10] Prend en charge plusieurs enregistrements « Nom du champ,Préfixe OID,Format » délimités par un caractère de nouvelle ligne.

item.create

Description

`object item.create(object/array items)`

Cette méthode vous permet de créer de nouveaux éléments.

Note:
Les éléments web ne peuvent pas être créés via l'API Zabbix.

Note:
Cette méthode est disponible uniquement pour les types d'utilisateur *Admin* et *Super admin*. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres des rôles utilisateur. Consultez Rôles utilisateur pour plus d'informations.

Paramètres

`(object/array)` Éléments à créer.

En plus des propriétés standard des éléments, la méthode accepte les paramètres suivants.

Paramètre	Type	Description
preprocessing	array	Options de prétraitement des éléments.
tags	array	Tags d'élément.

Valeurs de retour

`(object)` Renvoie un objet contenant les ID des éléments créés dans la propriété `itemids`. L'ordre des ID renvoyés correspond à l'ordre des éléments transmis.

Exemples

Création d'un élément

Créez un élément numérique d'agent Zabbix avec 2 tags d'élément pour surveiller l'espace disque libre sur l'hôte avec l'ID « 30074 ».

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "item.create",
    "params": {
        "name": "Free disk space on /home/joe/",
        "key_": "vfs.fs.size[/home/joe/,free]",
        "hostid": "30074",
        "type": 0,
        "value_type": 3,
        "interfaceid": "30084",
        "tags": [
            {
                "tag": "component",
                "value": "storage"
            },
            {
                "tag": "equipment",
                "value": "workstation"
            }
        ],
        "delay": "30s"
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "24758"
        ]
    },
    "id": 1
}
```

Création d'un élément d'inventaire d'hôte

Créez un élément d'agent Zabbix pour renseigner le champ d'inventaire « OS » de l'hôte.

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "item.create",
    "params": {
        "name": "uname",
        "key_": "system.uname",
        "hostid": "30021",
        "type": 0,
        "interfaceid": "30007",
        "value_type": 1,
        "delay": "10s",
        "inventory_link": 5
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "24759"
        ]
    },
    "id": 1
}
```

Création d'un élément avec prétraitement

Créez un élément en utilisant un multiplicateur personnalisé.

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "item.create",
    "params": {
        "name": "Device uptime",
        "key_": "sysUpTime",
        "hostid": "10084",
        "type": 20,
        "snmp_oid": "SNMPv2-MIB::sysUpTime.0",
        "value_type": 1,
        "delay": "60s",
        "interfaceid": "83",
        "preprocessing": [
            {
                "type": 1,
                "params": "0.01",
                "error_handler": 1,
                "error_handler_params": ""
            }
        ]
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "44210"
        ]
    },
    "id": 1
}
```

Création d'un élément dépendant

Créez un élément dépendant pour l'élément maître avec l'ID 24759. Seules les dépendances sur le même hôte sont autorisées ; par conséquent, l'élément maître et l'élément dépendant doivent avoir le même hostid.

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "item.create",
    "params": {
      "hostid": "30074",
      "name": "Dependent test item",
      "key_": "dependent.item",
      "type": 18,
```

```
      "master_itemid": "24759",
      "value_type": 2
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "44211"
        ]
    },
    "id": 1
}
```

Créer un élément d'agent HTTP

Créer un élément avec la méthode de requête POST et le prétraitement d'une réponse JSON.

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "item.create",
    "params": {
        "url":"http://127.0.0.1/http.php",
        "query_fields": [
            {
                "name": "mode",
                "value": "json"
            },
            {
                "name": "min",
                "value": "10"
            },
            {
                "name": "max",
                "value": "100"
            }
        ],
        "interfaceid": "1",
        "type": 19,
        "hostid": "10254",
        "delay": "5s",
        "key_": "json",
        "name": "HTTP agent example JSON",
        "value_type": 0,
        "output_format": 1,
        "preprocessing": [
            {
                "type": 12,
                "params": "$.random",
                "error_handler": 0,
                "error_handler_params": ""
            }
        ]
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "23865"
        ]
    },
    "id": 1
}
```

Créer un élément de script

Créez une collecte de données simple à l'aide d'un élément de script.

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "item.create",
    "params": {
        "name": "Script example",
        "key_": "custom.script.item",
        "hostid": "12345",
        "type": 21,
        "value_type": 4,
        "params": "var request = new HttpRequest();\nreturn request.post(\"https://postman-echo.com/post\"
        "parameters": [
            {
                "name": "host",
                "value": "{HOST.CONN}"
            }
        ],
        "timeout": "6s",
        "delay": "30s"
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "23865"
        ]
    },
    "id": 1
}
```

Source

CItem::create() dans *ui/include/classes/api/services/CItem.php*.

item.delete

Description

`object item.delete(array itemIds)`

Cette méthode vous permet de supprimer des éléments.

> **Note:**
> Les éléments web ne peuvent pas être supprimés via l'API Zabbix.

Note:
Cette méthode est disponible uniquement pour les types d'utilisateur *Admin* et *Super admin*. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres du rôle utilisateur. Consultez Rôles d'utilisateur pour plus d'informations.

Paramètres

`(array)` ID des éléments à supprimer.

Valeurs retournées

`(object)` Retourne un objet contenant les identifiants des éléments supprimés sous la propriété `itemids`.

Exemples

Suppression de plusieurs éléments

Supprimez deux éléments.
 Les éléments dépendants et les prototypes d'élément sont supprimés automatiquement si l'élément maître est supprimé.

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "item.delete",
    "params": [
        "22982",
        "22986"
    ],
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "22982",
            "22986"
        ]
    },
    "id": 1
}
```

Source

CItem::delete() dans *ui/include/classes/api/services/CItem.php*.

item.get

Description

`integer/array item.get(object parameters)`

La méthode permet de récupérer des éléments selon les paramètres fournis.

Note:
Cette méthode est disponible pour les utilisateurs de tout type. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres des rôles utilisateur. Consultez Rôles utilisateur pour plus d'informations.

Paramètres

`(object)` Paramètres définissant la sortie souhaitée.

La méthode prend en charge les paramètres suivants.

Paramètre	Type	Description
itemids	ID/array	Renvoyer uniquement les éléments avec les ID donnés.
groupids	ID/array	Renvoyer uniquement les éléments qui appartiennent aux hôtes des groupes donnés.
templateids	ID/array	Renvoyer uniquement les éléments qui appartiennent aux modèles donnés.
hostids	ID/array	Renvoyer uniquement les éléments qui appartiennent aux hôtes donnés.
proxyids	ID/array	Renvoyer uniquement les éléments qui sont surveillés par les proxies donnés.
interfaceids	ID/array	Renvoyer uniquement les éléments qui utilisent les interfaces d'hôte données.
graphids	ID/array	Renvoyer uniquement les éléments qui sont utilisés dans les graphiques donnés.
triggerids	ID/array	Renvoyer uniquement les éléments qui sont utilisés dans les déclencheurs donnés.
webitems	flag	Inclure les éléments web dans le résultat.
inherited	boolean	Si défini sur `true`, renvoyer uniquement les éléments hérités d'un modèle.
inheritedTags	boolean	Renvoyer les éléments qui ont les `tags` donnés également dans l'hôte/modèle/modèles liés. Valeurs possibles : `true` - le modèle/l'hôte/modèles liés doivent également avoir les tags donnés ; `false` - *(par défaut)* les tags du modèle/de l'hôte/des modèles liés sont ignorés.
templated	boolean	Si défini sur `true`, renvoyer uniquement les éléments qui appartiennent à des modèles.
monitored	boolean	Si défini sur `true`, renvoyer uniquement les éléments activés qui appartiennent à des hôtes surveillés.
group	string	Renvoyer uniquement les éléments qui appartiennent à un groupe portant le nom donné.
host	string	Renvoyer uniquement les éléments qui appartiennent à un hôte portant le nom donné.
evaltype	integer	Méthode d'évaluation des tags. Valeurs possibles : 0 - *(par défaut)* Et/Ou ; 2 - Ou.
tags	array	Renvoyer uniquement les éléments avec les tags donnés. Format : `[{"tag": "<tag>", "value": "<value>", "operator": "<operator>"}, ...]`. Un tableau vide renvoie tous les éléments. Valeurs possibles de operator : 0 - *(par défaut)* Contient ; 1 - Égal à ; 2 - Ne contient pas ; 3 - N'est pas égal à ; 4 - Existe ; 5 - N'existe pas.
with_triggers	boolean	Si défini sur `true`, renvoyer uniquement les éléments qui sont utilisés dans des déclencheurs.
selectHosts	query	Renvoyer une propriété `hosts` avec un tableau des hôtes auxquels l'élément appartient.
selectInterfaces	query	Renvoyer une propriété `interfaces` avec un tableau des interfaces d'hôte utilisées par l'élément.
selectTriggers	query	Renvoyer une propriété `triggers` avec les déclencheurs dans lesquels l'élément est utilisé. Prend en charge `count`.

Paramètre	Type	Description
selectGraphs	query	Renvoyer une propriété `graphs` avec les graphiques qui contiennent l'élément. Prend en charge `count`.
selectDiscoveryData	query	Renvoyer une propriété `discoveryData` avec les données de l'objet de découverte de l'élément. L'objet de découverte de l'élément relie un élément découvert à un prototype d'élément à partir duquel il a été découvert. Il possède les propriétés suivantes : `parent_itemid` - `(string)` ID du prototype d'élément à partir duquel l'élément a été créé ; `key_` - `(string)` clé du prototype d'élément ; `status` - `(int)` statut de découverte de l'élément : 0 - *(par défaut)* l'élément est découvert, 1 - l'élément n'est plus découvert ; `ts_delete` - `(timestamp)` heure à laquelle un élément qui n'est plus découvert sera supprimé ; `ts_disable` - `(timestamp)` heure à laquelle un élément qui n'est plus découvert sera désactivé ; `disable_source` - `(int)` indicateur précisant si l'élément a été désactivé par une règle LLD ou manuellement : 0 - *(par défaut)* désactivé automatiquement, 1 - désactivé par une règle LLD.
selectDiscoveryRule	query	Renvoyer une propriété `discoveryRule` avec la règle LLD qui a créé l'élément.
selectInheritedTags	query	Renvoyer une propriété `inheritedTags` avec les tags présents sur le modèle/l'hôte/modèles liés.
selectPreprocessing	query	Renvoyer une propriété `preprocessing` avec les options de prétraitement de l'élément.
selectTags	query	Renvoyer les tags de l'élément dans la propriété `tags`.
selectValueMap	query	Renvoyer une propriété `valuemap` avec la table de correspondance des valeurs de l'élément.
filter	object	Renvoyer uniquement les résultats qui correspondent exactement au filtre donné. Accepte un objet où les clés sont des noms de propriété et les valeurs sont soit une valeur unique, soit un tableau de valeurs à comparer. Ne prend pas en charge les propriétés de type de données `text` data type. Prend en charge des propriétés supplémentaires : `host` - nom technique de l'hôte auquel l'élément appartient.
limitSelects	integer	Limite le nombre d'enregistrements renvoyés par les sous-sélections. S'applique aux sous-sélections suivantes : `selectGraphs` - les résultats seront triés par `name` ; `selectTriggers` - les résultats seront triés par `description`.
sortfield	string/array	Trier le résultat selon les propriétés données. Valeurs possibles : `itemid`, `name`, `key_`, `delay`, `history`, `trends`, `type`, `status`.
countOutput	boolean	Ces paramètres sont décrits dans le commentaire de référence.
editable	boolean	
excludeSearch	boolean	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	

Paramètre	Type	Description
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	
selectItemDiscovery	query	Renvoyer une propriété `itemDiscovery` avec l'objet de découverte de l'élément. L'objet de découverte de l'élément relie l'élément à un prototype d'élément à partir duquel il a été créé. Cette requête est **obsolète** ; veuillez utiliser `selectDiscoveryData` à la place.

Valeurs de retour

`(integer/array)` Renvoie soit :

- un tableau d'objets ;
- le nombre d'objets récupérés, si le paramètre `countOutput` a été utilisé.

Exemples

Recherche d'éléments par clé

Récupérez tous les éléments utilisés dans des déclencheurs pour un ID d'hôte spécifique qui ont le mot « system.cpu » dans la clé de l'élément, puis triez les résultats par nom.

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "item.get",
    "params": {
        "output": "extend",
        "hostids": "10084",
        "with_triggers": true,
        "search": {
            "key_": "system.cpu"
        },
        "sortfield": "name"
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "itemid": "42269",
            "type": "18",
            "snmp_oid": "",
            "hostid": "10084",
            "name": "Utilisation du processeur",
            "key_": "system.cpu.util",
            "delay": "0",
            "history": "7d",
            "trends": "365d",
            "status": "0",
            "value_type": "0",
            "trapper_hosts": "",
            "units": "%",
            "formula": "",
            "logtimefmt": "",
            "templateid": "42267",
            "valuemapid": "0",
            "params": "",
```

```
        "ipmi_sensor": "",
        "authtype": "0",
        "username": "",
        "password": "",
        "publickey": "",
        "privatekey": "",
        "flags": "0",
        "interfaceid": "0",
        "description": "Utilisation du processeur en %.",
        "inventory_link": "0",
        "lifetime": "7d",
        "evaltype": "0",
        "jmx_endpoint": "",
        "master_itemid": "42264",
        "timeout": "",
        "url": "",
        "query_fields": [],
        "posts": "",
        "status_codes": "200",
        "follow_redirects": "1",
        "post_type": "0",
        "http_proxy": "",
        "headers": [],
        "retrieve_mode": "0",
        "request_method": "0",
        "output_format": "0",
        "ssl_cert_file": "",
        "ssl_key_file": "",
        "ssl_key_password": "",
        "verify_peer": "0",
        "verify_host": "0",
        "allow_traps": "0",
        "uuid": "",
        "lifetime_type": "0",
        "enabled_lifetime_type": "2",
        "enabled_lifetime": "0",
        "state": "0",
        "error": "",
        "name_resolved": "Utilisation du processeur",
        "parameters": [],
        "lastclock": "0",
        "lastns": "0",
        "lastvalue": "0",
        "prevvalue": "0"
    },
    {
        "itemid": "42259",
        "type": "0",
        "snmp_oid": "",
        "hostid": "10084",
        "name": "Charge moyenne (moy. sur 15 min)",
        "key_": "system.cpu.load[all,avg15]",
        "delay": "1m",
        "history": "7d",
        "trends": "365d",
        "status": "0",
        "value_type": "0",
        "trapper_hosts": "",
        "units": "",
        "formula": "",
        "logtimefmt": "",
        "templateid": "42219",
```

```
            "valuemapid": "0",
            "params": "",
            "ipmi_sensor": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "flags": "0",
            "interfaceid": "1",
            "description": "",
            "inventory_link": "0",
            "lifetime": "7d",
            "evaltype": "0",
            "jmx_endpoint": "",
            "master_itemid": "0",
            "timeout": "",
            "url": "",
            "query_fields": [],
            "posts": "",
            "status_codes": "200",
            "follow_redirects": "1",
            "post_type": "0",
            "http_proxy": "",
            "headers": [],
            "retrieve_mode": "0",
            "request_method": "0",
            "output_format": "0",
            "ssl_cert_file": "",
            "ssl_key_file": "",
            "ssl_key_password": "",
            "verify_peer": "0",
            "verify_host": "0",
            "allow_traps": "0",
            "uuid": "",
            "lifetime_type": "0",
            "enabled_lifetime_type": "2",
            "enabled_lifetime": "0",
            "state": "0",
            "error": "",
            "name_resolved": "Charge moyenne (moy. sur 15 min)",
            "parameters": [],
            "lastclock": "0",
            "lastns": "0",
            "lastvalue": "0",
            "prevvalue": "0"
        },
        {
            "itemid": "42249",
            "type": "0",
            "snmp_oid": "",
            "hostid": "10084",
            "name": "Charge moyenne (moy. sur 1 min)",
            "key_": "system.cpu.load[all,avg1]",
            "delay": "1m",
            "history": "7d",
            "trends": "365d",
            "status": "0",
            "value_type": "0",
            "trapper_hosts": "",
            "units": "",
            "formula": "",
```

```
            "logtimefmt": "",
            "templateid": "42209",
            "valuemapid": "0",
            "params": "",
            "ipmi_sensor": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "flags": "0",
            "interfaceid": "1",
            "description": "",
            "inventory_link": "0",
            "lifetime": "7d",
            "evaltype": "0",
            "jmx_endpoint": "",
            "master_itemid": "0",
            "timeout": "",
            "url": "",
            "query_fields": [],
            "posts": "",
            "status_codes": "200",
            "follow_redirects": "1",
            "post_type": "0",
            "http_proxy": "",
            "headers": [],
            "retrieve_mode": "0",
            "request_method": "0",
            "output_format": "0",
            "ssl_cert_file": "",
            "ssl_key_file": "",
            "ssl_key_password": "",
            "verify_peer": "0",
            "verify_host": "0",
            "allow_traps": "0",
            "uuid": "",
            "lifetime_type": "0",
            "enabled_lifetime_type": "2",
            "enabled_lifetime": "0",
            "state": "0",
            "error": "",
            "name_resolved": "Charge moyenne (moy. sur 1 min)",
            "parameters": [],
            "lastclock": "0",
            "lastns": "0",
            "lastvalue": "0",
            "prevvalue": "0"
        },
        {
            "itemid": "42257",
            "type": "0",
            "snmp_oid": "",
            "hostid": "10084",
            "name": "Charge moyenne (moy. sur 5 min)",
            "key_": "system.cpu.load[all,avg5]",
            "delay": "1m",
            "history": "7d",
            "trends": "365d",
            "status": "0",
            "value_type": "0",
            "trapper_hosts": "",
```

```
        "units": "",
        "formula": "",
        "logtimefmt": "",
        "templateid": "42217",
        "valuemapid": "0",
        "params": "",
        "ipmi_sensor": "",
        "authtype": "0",
        "username": "",
        "password": "",
        "publickey": "",
        "privatekey": "",
        "flags": "0",
        "interfaceid": "1",
        "description": "",
        "inventory_link": "0",
        "lifetime": "7d",
        "evaltype": "0",
        "jmx_endpoint": "",
        "master_itemid": "0",
        "timeout": "",
        "url": "",
        "query_fields": [],
        "posts": "",
        "status_codes": "200",
        "follow_redirects": "1",
        "post_type": "0",
        "http_proxy": "",
        "headers": [],
        "retrieve_mode": "0",
        "request_method": "0",
        "output_format": "0",
        "ssl_cert_file": "",
        "ssl_key_file": "",
        "ssl_key_password": "",
        "verify_peer": "0",
        "verify_host": "0",
        "allow_traps": "0",
        "uuid": "",
        "lifetime_type": "0",
        "enabled_lifetime_type": "2",
        "enabled_lifetime": "0",
        "state": "0",
        "error": "",
        "name_resolved": "Charge moyenne (moy. sur 5 min)",
        "parameters": [],
        "lastclock": "0",
        "lastns": "0",
        "lastvalue": "0",
        "prevvalue": "0"
    },
    {
        "itemid": "42260",
        "type": "0",
        "snmp_oid": "",
        "hostid": "10084",
        "name": "Nombre de processeurs",
        "key_": "system.cpu.num",
        "delay": "1m",
        "history": "7d",
        "trends": "365d",
        "status": "0",
```

```
            "value_type": "3",
            "trapper_hosts": "",
            "units": "",
            "formula": "",
            "logtimefmt": "",
            "templateid": "42220",
            "valuemapid": "0",
            "params": "",
            "ipmi_sensor": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "flags": "0",
            "interfaceid": "1",
            "description": "",
            "inventory_link": "0",
            "lifetime": "7d",
            "evaltype": "0",
            "jmx_endpoint": "",
            "master_itemid": "0",
            "timeout": "",
            "url": "",
            "query_fields": [],
            "posts": "",
            "status_codes": "200",
            "follow_redirects": "1",
            "post_type": "0",
            "http_proxy": "",
            "headers": [],
            "retrieve_mode": "0",
            "request_method": "0",
            "output_format": "0",
            "ssl_cert_file": "",
            "ssl_key_file": "",
            "ssl_key_password": "",
            "verify_peer": "0",
            "verify_host": "0",
            "allow_traps": "0",
            "uuid": "",
            "lifetime_type": "0",
            "enabled_lifetime_type": "2",
            "enabled_lifetime": "0",
            "state": "0",
            "error": "",
            "name_resolved": "Nombre de processeurs",
            "parameters": [],
            "lastclock": "0",
            "lastns": "0",
            "lastvalue": "0",
            "prevvalue": "0"
        }
    ],
    "id": 1
}
```


Recherche d'éléments dépendants par clé

Récupérez tous les éléments dépendants de l'hôte avec l'ID "10116" dont la clé contient le mot "apache".

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "item.get",
    "params": {
        "output": "extend",
        "hostids": "10116",
        "search": {
            "key_": "apache"
        },
        "filter": {
            "type": 18
        }
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "itemid": "25550",
            "type": "18",
            "snmp_oid": "",
            "hostid": "10116",
            "name": "Days",
            "key_": "apache.status.uptime.days",
            "delay": "0",
            "history": "90d",
            "trends": "365d",
            "status": "0",
            "value_type": "3",
            "trapper_hosts": "",
            "units": "",
            "formula": "",
            "logtimefmt": "",
            "templateid": "0",
            "valuemapid": "0",
            "params": "",
            "ipmi_sensor": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "flags": "0",
            "interfaceid": "0",
            "description": "",
            "inventory_link": "0",
            "lifetime": "7d",
            "evaltype": "0",
            "jmx_endpoint": "",
            "master_itemid": "25545",
            "timeout": "",
            "url": "",
            "query_fields": [],
            "posts": "",
            "status_codes": "200",
            "follow_redirects": "1",
            "post_type": "0",
            "http_proxy": "",
            "headers": [],
```

```
            "retrieve_mode": "0",
            "request_method": "0",
            "output_format": "0",
            "ssl_cert_file": "",
            "ssl_key_file": "",
            "ssl_key_password": "",
            "verify_peer": "0",
            "verify_host": "0",
            "allow_traps": "0",
            "uuid": "",
            "lifetime_type": "0",
            "enabled_lifetime_type": "2",
            "enabled_lifetime": "0",
            "state": "0",
            "error": "",
            "name_resolved": "Days",
            "parameters": [],
            "lastclock": "0",
            "lastns": "0",
            "lastvalue": "0",
            "prevvalue": "0"
        },
        {
            "itemid": "25555",
            "type": "18",
            "snmp_oid": "",
            "hostid": "10116",
            "name": "Hours",
            "key_": "apache.status.uptime.hours",
            "delay": "0",
            "history": "90d",
            "trends": "365d",
            "status": "0",
            "value_type": "3",
            "trapper_hosts": "",
            "units": "",
            "formula": "",
            "logtimefmt": "",
            "templateid": "0",
            "valuemapid": "0",
            "params": "",
            "ipmi_sensor": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "flags": "0",
            "interfaceid": "0",
            "description": "",
            "inventory_link": "0",
            "lifetime": "7d",
            "evaltype": "0",
            "jmx_endpoint": "",
            "master_itemid": "25545",
            "timeout": "",
            "url": "",
            "query_fields": [],
            "posts": "",
            "status_codes": "200",
            "follow_redirects": "1",
            "post_type": "0",
```

```
            "http_proxy": "",
            "headers": [],
            "retrieve_mode": "0",
            "request_method": "0",
            "output_format": "0",
            "ssl_cert_file": "",
            "ssl_key_file": "",
            "ssl_key_password": "",
            "verify_peer": "0",
            "verify_host": "0",
            "allow_traps": "0",
            "uuid": "",
            "lifetime_type": "0",
            "enabled_lifetime_type": "2",
            "enabled_lifetime": "0",
            "state": "0",
            "error": "",
            "name_resolved": "Hours",
            "parameters": [],
            "lastclock": "0",
            "lastns": "0",
            "lastvalue": "0",
            "prevvalue": "0"
        }
    ],
    "id": 1
}
```


Rechercher un élément d'agent HTTP

Recherchez un élément d'agent HTTP avec un type de corps POST XML pour un ID d'hôte spécifique.

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "item.get",
    "params": {
        "hostids": "10255",
        "filter": {
            "type": 19,
            "post_type": 3
        }
    },
    "id": 1
}
```

Réponse :


```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "itemid": "28252",
            "type": "19",
            "snmp_oid": "",
            "hostid": "10255",
            "name": "template item",
            "key_": "ti",
            "delay": "30s",
            "history": "90d",
            "trends": "365d",
            "status": "0",
            "value_type": "3",
            "trapper_hosts": "",
```

```
            "units": "",
            "formula": "",
            "logtimefmt": "",
            "templateid": "0",
            "valuemapid": "0",
            "params": "",
            "ipmi_sensor": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "flags": "0",
            "interfaceid": "0",
            "description": "",
            "inventory_link": "0",
            "lifetime": "7d",
            "evaltype": "0",
            "jmx_endpoint": "",
            "master_itemid": "0",
            "timeout": "",
            "url": "localhost",
            "query_fields": [
                {
                    "name": "mode",
                    "value": "xml"
                }
            ],
            "posts": "<body>\r\n<![CDATA[{$MACRO}<foo></bar>]]>\r\n</body>",
            "status_codes": "200",
            "follow_redirects": "0",
            "post_type": "3",
            "http_proxy": "",
            "headers": [],
            "retrieve_mode": "1",
            "request_method": "3",
            "output_format": "0",
            "ssl_cert_file": "",
            "ssl_key_file": "",
            "ssl_key_password": "",
            "verify_peer": "0",
            "verify_host": "0",
            "allow_traps": "0",
            "uuid": "",
            "lifetime_type": "0",
            "enabled_lifetime_type": "2",
            "enabled_lifetime": "0",
            "state": "0",
            "error": "",
            "name_resolved": "template item",
            "parameters": [],
            "lastclock": "0",
            "lastns": "0",
            "lastvalue": "",
            "prevvalue": ""
        }
    ],
    "id": 1
}
```


Récupération des éléments avec des règles de prétraitement

Récupérez tous les éléments et leurs règles de prétraitement pour un ID d'hôte spécifique.

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "item.get",
    "params": {
        "output": ["itemid", "name", "key_"],
        "selectPreprocessing": "extend",
        "hostids": "10254"
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemid": "23865",
        "name": "HTTP agent example JSON",
        "key_": "json",
        "preprocessing": [
            {
                "type": "12",
                "params": "$.random",
                "error_handler": "1",
                "error_handler_params": ""
            }
        ]
    },
    "id": 1
}
```

Voir aussi

- Règle de découverte
- Graphique
- Hôte
- Interface d'hôte
- Déclencheur

Source

CItem::get() dans *ui/include/classes/api/services/CItem.php*.

item.update

Description

`object item.update(object/array items)`

Cette méthode vous permet de mettre à jour des éléments existants.

> **Note:**
> Les éléments web ne peuvent pas être mis à jour via l'API Zabbix.

> **Note:**
> Cette méthode est disponible uniquement pour les types d'utilisateur *Admin* et *Super admin*. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres des rôles utilisateur. Consultez Rôles utilisateur pour plus d'informations.

Paramètres

`(object/array)` Propriétés des éléments à mettre à jour.

La propriété `itemid` doit être définie pour chaque élément, toutes les autres propriétés sont facultatives. Seules les propriétés transmises seront mises à jour, toutes les autres resteront inchangées.

En plus des propriétés standard des éléments, la méthode accepte les paramètres suivants.

Paramètre	Type	Description
preprocessing	array	Options de prétraitement des éléments pour remplacer les options de prétraitement actuelles. Comportement des paramètres: - *lecture seule* pour les objets hérités ou les objets découverts
tags	array	Tags d'élément. Comportement des paramètres: - *lecture seule* pour les objets découverts

Valeurs retournées

`(object)` Retourne un objet contenant les identifiants des éléments mis à jour sous la propriété `itemids`.

Exemples

Activation d'un élément

Activez un élément, c'est-à-dire définissez son statut sur « 0 ».

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "item.update",
    "params": {
        "itemid": "10092",
        "status": 0
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "10092"
        ]
    },
    "id": 1
}
```

Mettre à jour un élément dépendant

Mettez à jour le nom de l'élément dépendant et l'ID de l'élément maître. Seules les dépendances sur le même hôte sont autorisées, par conséquent l'élément maître et l'élément dépendant doivent avoir le même hostid.

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "item.update",
    "params": {
        "name": "Dependent item updated name",
        "master_itemid": "25562",
        "itemid": "189019"
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "189019"
        ]
    },
    "id": 1
}
```

Mettre à jour un élément d'agent HTTP

Activer l'interception des valeurs de l'élément.

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "item.update",
    "params": {
        "itemid": "23856",
        "allow_traps": 1
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "23856"
        ]
    },
    "id": 1
}
```

Mise à jour d'un élément avec le prétraitement

Mettez à jour un élément avec la règle de prétraitement d'élément « In range ».

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "item.update",
    "params": {
        "itemid": "23856",
        "preprocessing": [
            {
                "type": 13,
                "params": "\n100",
                "error_handler": 1,
                "error_handler_params": ""
            }
        ]
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "23856"
```

```
        ]
    },
    "id": 1
}
```

Mise à jour d'un élément de script

Mettez à jour un élément de script avec un script différent et supprimez les paramètres inutiles qui étaient utilisés par le script précédent.

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "item.update",
    "params": {
        "itemid": "23865",
        "parameters": [],
        "params": "Zabbix.log(3, 'Log test');\nreturn 1;"
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "23865"
        ]
    },
    "id": 1
}
```

Source

CItem::update() dans *ui/include/classes/api/services/CItem.php*.

Élément de graphique

Cette classe est conçue pour fonctionner avec les éléments de graphique.

Références d'objet :

- Élément de graphique

Méthodes disponibles :

- graphitem.get - récupérer des éléments de graphique

Objet d'élément de graphique

Les objets suivants sont directement liés à l'API `graphitem`.

Élément de graphique

> **Note:**
> Les éléments de graphique ne peuvent être modifiés que via l'API `graph`.

L'objet élément de graphique possède les propriétés suivantes.

Propriété	Type	Description
gitemid	ID	ID de l'élément de graphique. Comportement de la propriété: - *lecture seule*
color	string	Couleur de dessin de l'élément de graphique sous forme de code couleur hexadécimal. Comportement de la propriété: - *obligatoire* pour les opérations de création
itemid	ID	ID de l'élément. Comportement de la propriété: - *obligatoire* pour les opérations de création
calc_fnc	integer	Valeur de l'élément qui sera affichée. Valeurs possibles : 1 - valeur minimale ; 2 - *(par défaut)* valeur moyenne ; 4 - valeur maximale ; 7 - toutes les valeurs ; 9 - dernière valeur, utilisée uniquement par les graphiques en secteurs et éclatés.
drawtype	integer	Style de tracé de l'élément de graphique. Valeurs possibles : 0 - *(par défaut)* ligne ; 1 - zone remplie ; 2 - ligne en gras ; 3 - point ; 4 - ligne en pointillés ; 5 - ligne en dégradé.
graphid	ID	ID du graphique auquel appartient l'élément de graphique.
sortorder	integer	Position de l'élément dans le graphique. Par défaut : commence à "0" et augmente de un à chaque entrée.
type	integer	Type d'élément de graphique. Valeurs possibles : 0 - *(par défaut)* simple ; 2 - somme du graphique, utilisée uniquement par les graphiques en secteurs et éclatés.
yaxisside	integer	Côté du graphique où l'échelle Y de l'élément de graphique sera dessinée. Valeurs possibles : 0 - *(par défaut)* côté gauche ; 1 - côté droit.

graphitem.get

Description

`integer/array graphitem.get(object parameters)`

La méthode permet de récupérer les éléments de graphique selon les paramètres fournis.

> **Note:**
> Cette méthode est disponible pour les utilisateurs de tout type. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres des rôles utilisateur. Consultez Rôles utilisateur pour plus d'informations.

Paramètres

`(object)` Paramètres définissant la sortie souhaitée.

La méthode prend en charge les paramètres suivants.

Paramètre	Type	Description
graphids	ID/array	Retourner uniquement les éléments de graphique qui appartiennent aux graphiques donnés.
itemids	ID/array	Retourner uniquement les éléments de graphique avec les ID d'élément donnés.
type	integer	Retourner uniquement les éléments de graphique du type donné. Consultez la page de l'objet d'élément de graphique pour obtenir la liste des types d'éléments de graphique pris en charge.
selectGraphs	query	Retourner une propriété `graphs` avec un tableau des graphiques auxquels l'élément appartient.
sortfield	string/array	Trier le résultat selon les propriétés données. Valeurs possibles : `gitemid`.
countOutput	boolean	Ces paramètres sont décrits dans le commentaire de référence.
editable	boolean	
limit	integer	
output	query	
preservekeys	boolean	
sortorder	string/array	

Valeurs retournées

`(integer/array)` Retourne soit :

- un tableau d'objets ;
- le nombre d'objets récupérés, si le paramètre `countOutput` a été utilisé.

Exemples

Récupération des éléments de graphique à partir d'un graphique

Récupérez tous les éléments de graphique utilisés dans un graphique avec des informations supplémentaires sur l'élément et l'hôte.

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "graphitem.get",
    "params": {
        "output": "extend",
        "graphids": "387"
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "gitemid": "1242",
            "graphid": "387",
            "itemid": "22665",
            "drawtype": "1",
            "sortorder": "1",
            "color": "FF5555",
            "yaxisside": "0",
            "calc_fnc": "2",
            "type": "0"
```

```
        },
        {
            "gitemid": "1243",
            "graphid": "387",
            "itemid": "22668",
            "drawtype": "1",
            "sortorder": "2",
            "color": "55FF55",
            "yaxisside": "0",
            "calc_fnc": "2",
            "type": "0"
        },
        {
            "gitemid": "1244",
            "graphid": "387",
            "itemid": "22671",
            "drawtype": "1",
            "sortorder": "3",
            "color": "009999",
            "yaxisside": "0",
            "calc_fnc": "2",
            "type": "0"
        }
    ],
    "id": 1
}
```


Voir également

- Graphique

Source

CGraphItem::get() dans *ui/include/classes/api/services/CGraphItem.php*.

Événement

Cette classe est conçue pour fonctionner avec les événements.

Références d'objet :

- Event
- Event tag
- Media type URL

Méthodes disponibles :

- event.get - récupérer des événements
- event.acknowledge - acquitter des événements

Objet événement

Les objets suivants sont directement liés à l'API `event`.

Événement

> **Note:**
> Les événements sont créés par le serveur Zabbix et ne peuvent pas être modifiés via l'API.

L'objet événement possède les propriétés suivantes.

Property	Type	Description
eventid	ID	ID de l'événement.

Property	Type	Description
source	integer	Type de l'événement. Valeurs possibles : 0 - événement créé par un déclencheur ; 1 - événement créé par une règle de découverte ; 2 - événement créé par l'autoréenregistrement actif de l'agent ; 3 - événement interne ; 4 - événement créé lors de la mise à jour de l'état du service.
object	integer	Type d'objet lié à l'événement. Valeurs possibles si `source` est défini sur « événement créé par un déclencheur » : 0 - déclencheur. Valeurs possibles si `source` est défini sur « événement créé par une règle de découverte » : 1 - hôte découvert ; 2 - service découvert. Valeurs possibles si `source` est défini sur « événement créé par l'autoréenregistrement actif de l'agent » : 3 - hôte autoréenregistré. Valeurs possibles si `source` est défini sur « événement interne » : 0 - déclencheur ; 4 - élément ; 5 - règle LLD. Valeurs possibles si `source` est défini sur « événement créé lors de la mise à jour de l'état du service » : 6 - service.
objectid	ID	ID de l'objet lié.
acknowledged	integer	Indique si l'événement a été acquitté.
clock	timestamp	Heure de création de l'événement.
ns	integer	Nanosecondes au moment de la création de l'événement.
name	string	Nom résolu de l'événement.
value	integer	État de l'objet lié. Valeurs possibles si `source` est défini sur « événement créé par un déclencheur » ou « événement créé lors de la mise à jour de l'état du service » : 0 - OK ; 1 - problème. Valeurs possibles si `source` est défini sur « événement créé par une règle de découverte » : 0 - hôte ou service opérationnel ; 1 - hôte ou service indisponible ; 2 - hôte ou service découvert ; 3 - hôte ou service perdu. Valeurs possibles si `source` est défini sur « événement interne » : 0 - état « normal » ; 1 - état « inconnu » ou « non pris en charge ». Property behavior : - *supported* si `source` est défini sur « événement créé par un déclencheur », « événement créé par une règle de découverte », « événement interne » ou « événement créé lors de la mise à jour de l'état du service »

Property	Type	Description
severity	integer	Gravité actuelle de l'événement. Valeurs possibles : 0 - non classé ; 1 - information ; 2 - avertissement ; 3 - moyen ; 4 - élevé ; 5 - désastre.
r_eventid	ID	ID de l'événement de récupération.
c_eventid	ID	ID de l'événement utilisé pour remplacer (fermer) l'événement actuel selon une règle de corrélation globale. Voir `correlationid` pour identifier la règle de corrélation exacte. Ce paramètre n'est défini que lorsque l'événement est fermé par une règle de corrélation globale.
cause_eventid	ID	ID de l'événement cause.
correlationid	ID	ID de la règle de corrélation qui a généré la fermeture du problème. Ce paramètre n'est défini que lorsque l'événement est fermé par une règle de corrélation globale.
userid	ID	ID de l'utilisateur qui a fermé l'événement (si l'événement a été fermé manuellement).
suppressed	integer	Indique si l'événement est supprimé. Valeurs possibles : 0 - l'événement est dans un état normal ; 1 - l'événement est supprimé.
opdata	string	Données opérationnelles avec macros développées.
urls	array	URL de type de média actives.

Tag d'événement

L'objet de tag d'événement possède les propriétés suivantes.

Propriété	Type	Description
tag	string	Nom du tag d'événement.
value	string	Valeur du tag d'événement.

URL du type de média

L'objet URL du type de média possède les propriétés suivantes.

Propriété	Type	Description
name	string	Nom de l'URL définie pour le type de média.
url	string	Valeur de l'URL définie pour le type de média.

Les résultats contiendront des entrées uniquement pour les types de média actifs avec une entrée de menu d'événement activée. La macro utilisée dans les propriétés sera développée, mais si l'une des propriétés contient une macro non développée, les deux propriétés seront exclues des résultats. Pour les macros prises en charge, voir *Macros prises en charge*.

event.acknowledge

Description

`object event.acknowledge(object/array parameters)`

Cette méthode vous permet de mettre à jour des événements. Les actions de mise à jour suivantes peuvent être effectuées :

- Fermer l'événement. Si l'événement est déjà résolu, cette action sera ignorée.
- Reconnaître l'événement. Si l'événement est déjà reconnu, cette action sera ignorée.
- Annuler la reconnaissance de l'événement. Si l'événement n'est pas reconnu, cette action sera ignorée.

- Ajouter un message.
- Modifier la gravité de l'événement. Si l'événement a déjà la même gravité, cette action sera ignorée.
- Supprimer la suppression de l'événement. Si l'événement est déjà supprimé, cette action sera ignorée.
- Annuler la suppression de l'événement. Si l'événement n'est pas supprimé, cette action sera ignorée.
- Modifier le rang de l'événement en cause. Si l'événement est déjà classé comme cause, cette action sera ignorée.
- Modifier le rang de l'événement en symptôme. Si l'événement est déjà classé comme symptôme, cette action sera ignorée.

Attention:
Seuls les événements de déclencheur peuvent être mis à jour.
 Seuls les événements de problème peuvent être mis à jour.
 Des droits de lecture/écriture sur le déclencheur sont requis pour fermer l'événement ou modifier sa gravité.
 Pour fermer un événement, la fermeture manuelle doit être autorisée dans le déclencheur.

Note:
Cette méthode est disponible pour les utilisateurs de tout type. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres du rôle utilisateur. Voir Rôles utilisateur pour plus d'informations.

Paramètres

`(object/array)` Paramètres contenant les ID des événements et les opérations de mise à jour à effectuer.

Parameter	Type	Description
eventids	ID/array	ID des événements à acquitter. Comportement du paramètre : - *requis*
action	integer	Action(s) de mise à jour de l'événement. Valeurs bitmap possibles : 1 - fermer le problème ; 2 - acquitter l'événement ; 4 - ajouter un message ; 8 - modifier la sévérité ; 16 - annuler l'acquittement de l'événement ; 32 - supprimer l'événement ; 64 - annuler la suppression de l'événement ; 128 - modifier le classement de l'événement en cause ; 256 - modifier le classement de l'événement en symptôme. Il s'agit d'un champ de masque de bits ; toute somme de valeurs bitmap possibles est acceptée (par exemple, 6 pour acquitter un événement et laisser un message, ou 34 pour acquitter et supprimer un événement). Comportement du paramètre : - *requis*
cause_eventid	ID	ID de l'événement cause. Comportement du paramètre : - *requis* si `action` contient le bit « modifier le classement de l'événement en symptôme »
message	string	Texte du message. Comportement du paramètre : - *requis* si `action` contient le bit « ajouter un message »

Parameter	Type	Description
severity	integer	Nouvelle sévérité des événements. Valeurs possibles : 0 - non classifié ; 1 - information ; 2 - avertissement ; 3 - moyenne ; 4 - élevée ; 5 - désastre. Comportement du paramètre : - *requis* si `action` contient le bit « modifier la sévérité »
suppress_until	integer	Horodatage Unix jusqu'auquel l'événement doit être supprimé. Si la valeur est définie sur « 0 », la suppression sera indéfinie. Comportement du paramètre : - *requis* si `action` contient le bit « supprimer l'événement »

Valeurs de retour

`(object)` Renvoie un objet contenant les ID des événements mis à jour dans la propriété `eventids`.

Exemples

Accusé de réception d'un événement

Accusez réception d'un événement unique et laissez un message.

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "event.acknowledge",
    "params": {
        "eventids": "20427",
        "action": 6,
        "message": "Problème résolu."
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "eventids": [
            "20427"
        ]
    },
    "id": 1
}
```

Modification de la sévérité d'un événement

Modifiez la sévérité de plusieurs événements et laissez un message.

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "event.acknowledge",
    "params": {
        "eventids": ["20427", "20428"],
        "action": 12,
```

```
        "message": "Maintenance required to fix it.",
        "severity": 4
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "eventids": [
            "20427",
            "20428"
        ]
    },
    "id": 1
}
```

Suppression d'un événement indéfiniment

Supprimer la suppression d'un seul événement sans date d'expiration.

Request:

```
{
    "jsonrpc": "2.0",
    "method": "event.acknowledge",
    "params": {
        "eventids": "20427",
        "action": 32,
        "suppress_until": 0
    },
    "id": 1
}
```

Response:

```
{
    "jsonrpc": "2.0",
    "result": {
        "eventids": [
            20427
        ]
    },
    "id": 1
}
```

Annulation de la suppression d'un événement

Annulez la suppression d'un seul événement précédemment supprimé.

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "event.acknowledge",
    "params": {
        "eventids": "20427",
        "action": 64
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
```

```
        "eventids": [
            20427
        ]
    },
    "id": 1
}
```

Reconnaître et supprimer un déclencheur

Reconnaître un ou plusieurs événements, ajouter un message et les supprimer dans la même requête.

Request:

```
{
    "jsonrpc": "2.0",
    "method": "event.acknowledge",
    "params": {
        "eventids": [
            "20427",
            "20428"
        ],
        "action": 38,
        "message": "Suppressing during planned maintenance.",
        "suppress_until": 0
    },
    "id": 1
}
```

Response:

```
{
    "jsonrpc": "2.0",
    "result": {
        "eventids": [
            20427,
            20428
        ]
    },
    "id": 1
}
```

Source

CEvent::acknowledge() dans *ui/include/classes/api/services/CEvent.php*.

event.get

Description

`integer/array event.get(object parameters)`

La méthode permet de récupérer des événements selon les paramètres fournis.

Attention:
Cette méthode peut renvoyer des événements d'une entité supprimée si ces événements n'ont pas encore été supprimés par le housekeeper.

Note:
Cette méthode est disponible pour les utilisateurs de tout type. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres du rôle utilisateur. Voir User roles pour plus d'informations.

Paramètres

`(object)` Paramètres définissant la sortie souhaitée.

La méthode prend en charge les paramètres suivants.

Parameter	Type	Description
eventids	ID/array	Ne retourner que les événements avec les IDs donnés.
groupids	ID/array	Ne retourner que les événements créés par des objets appartenant aux groupes d'hôtes donnés.
hostids	ID/array	Ne retourner que les événements créés par des objets appartenant aux hôtes donnés.
objectids	ID/array	Ne retourner que les événements créés par les objets donnés.
source	integer	Ne retourner que les événements avec le type donné. Consultez la page de l'objet event pour la liste des types d'événements pris en charge. Par défaut: 0 - événements de déclencheur.
object	integer	Ne retourner que les événements créés par des objets du type donné. Consultez la page de l'objet event pour la liste des types d'objets pris en charge. Par défaut: 0 - déclencheur.
acknowledged	boolean	Si défini sur `true`, ne retourner que les événements reconnus.
action	integer	Ne retourner que les événements pour lesquels les actions de mise à jour d'événement données ont été exécutées. Pour plusieurs actions, utilisez une somme de valeurs bitmap acceptables quelconques comme masque binaire (par exemple, 34 pour reconnaître et supprimer l'événement).
action_userids	ID/array	Ne retourner que les événements avec les IDs donnés des utilisateurs ayant exécuté les actions de mise à jour d'événement.
suppressed	boolean	Si défini sur `true`, ne retourner que les événements supprimés.
symptom	boolean	Si défini sur `true`, ne retourner que les événements symptôme.
severities	integer/array	Ne retourner que les événements avec les niveaux de gravité donnés. S'applique uniquement si object est trigger.
trigger_severities	integer/array	Ne retourner que les événements avec les niveaux de gravité de déclencheur donnés. S'applique uniquement si object est trigger.
evaltype	integer	Méthode d'évaluation des tags. Valeurs possibles: 0 - *(par défaut)* Et/Ou; 2 - Ou.
tags	array	Ne retourner que les événements avec les tags donnés. Format: `[{"tag": "<tag>", "value": "<value>", "operator": "<operator>"}, ...]`. Un tableau vide retourne tous les événements. Valeurs possibles de operator: 0 - *(par défaut)* Contient; 1 - Égal à; 2 - Ne contient pas; 3 - N'est pas égal à; 4 - Existe; 5 - N'existe pas.
eventid_from	string	Ne retourner que les événements avec des IDs supérieurs ou égaux à l'ID donné.
eventid_till	string	Ne retourner que les événements avec des IDs inférieurs ou égaux à l'ID donné.
time_from	timestamp	Ne retourner que les événements créés à partir de l'heure donnée, ou après.
time_till	timestamp	Ne retourner que les événements créés à l'heure donnée, ou avant.
problem_time_from	timestamp	Retourne uniquement les événements qui étaient à l'état de problème à partir de `problem_time_from`, quel que soit leur état actuel. S'applique uniquement lorsque la source est un événement de déclencheur et que l'objet est un déclencheur. Ce paramètre est obligatoire si `problem_time_till` est spécifié.

Parameter	Type	Description
problem_time_till	timestamp	Retourne uniquement les événements qui étaient à l'état de problème jusqu'à `problem_time_till`, quel que soit leur état actuel. S'applique uniquement lorsque la source est un événement de déclencheur et que l'objet est un déclencheur. Ce paramètre est obligatoire si `problem_time_from` est spécifié.
value	integer/array	Ne retourner que les événements avec les valeurs données.
selectAcknowledges	query	Retourne une propriété `acknowledges` avec les mises à jour d'événement. Les mises à jour d'événement sont triées par ordre chronologique inverse. L'objet de mise à jour d'événement possède les propriétés suivantes: `acknowledgeid` - `(ID)` ID de la reconnaissance; `userid` - `(ID)` ID de l'utilisateur qui a mis à jour l'événement; `clock` - `(timestamp)` heure à laquelle l'événement a été mis à jour; `message` - `(string)` texte du message; `action` - `(integer)` action de mise à jour effectuée, voir `event.acknowledge`; `old_severity` - `(integer)` gravité de l'événement avant cette action de mise à jour; `new_severity` - `(integer)` gravité de l'événement après cette action de mise à jour; `suppress_until` - `(timestamp)` heure jusqu'à laquelle l'événement sera supprimé; `taskid` - `(ID)` ID de la tâche si l'événement actuel est en cours de changement de rang; `maintenanceid` - `(ID)` ID de la maintenance; `username` - `(string)` nom d'utilisateur de l'utilisateur qui a mis à jour l'événement; `name` - `(string)` prénom de l'utilisateur qui a mis à jour l'événement; `surname` - `(string)` nom de famille de l'utilisateur qui a mis à jour l'événement. Prend en charge `count`.
selectAlerts	query	Retourne une propriété `alerts` avec les alertes générées par l'événement. Les alertes sont triées par ordre chronologique inverse.
selectHosts	query	Retourne une propriété `hosts` avec les hôtes contenant l'objet qui a créé l'événement. Pris en charge uniquement pour les événements générés par des déclencheurs, des éléments ou des règles LLD.
selectRelatedObject	query	Retourne une propriété `relatedObject` avec l'objet qui a créé l'événement. Le type d'objet retourné dépend du type d'événement.
selectSuppressionData	query	Retourne une propriété `suppression_data` avec la liste des maintenances actives et des suppressions manuelles: `maintenanceid` - `(ID)` ID de la maintenance; `userid` - `(ID)` ID de l'utilisateur qui a supprimé l'événement; `suppress_until` - `(integer)` heure jusqu'à laquelle l'événement est supprimé.
selectTags	query	Retourne une propriété `tags` avec les tags de l'événement.
filter	object	Ne retourner que les résultats qui correspondent exactement au filtre donné. Accepte un objet, où les clés sont des noms de propriétés et les valeurs sont soit une seule valeur, soit un tableau de valeurs à comparer. Ne prend pas en charge les propriétés de type de données data type `text`.

Parameter	Type	Description
sortfield	string/array	Trier le résultat selon les propriétés données. Valeurs possibles: `eventid`, `objectid`, `clock`. Valeurs possibles lorsqu'il est utilisé avec `groupBy`: `objectid`. Valeurs possibles lorsqu'il est utilisé avec `countOutput` et `groupBy`: `objectid`, `rowscount`.
groupBy	string/array	Grouper les résultats selon les propriétés données. Les propriétés spécifiées seront retournées dans les résultats. Valeurs possibles: `objectid`.
countOutput	boolean	Ces paramètres sont décrits dans le commentaire de référence.
editable	boolean	
excludeSearch	boolean	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Valeurs de retour

`(integer/array)` Renvoie soit :

- un tableau d'objets
- le nombre d'objets récupérés, si le paramètre `countOutput` a été utilisé, mais que le paramètre `groupBy` n'a pas été utilisé
- un tableau d'objets avec les résultats d'agrégation, si le paramètre `groupBy` a été utilisé

Exemples

Récupération des événements de déclencheur

Récupérez les événements les plus récents du déclencheur "22395".

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "event.get",
    "params": {
        "output": "extend",
        "selectAcknowledges": "extend",
        "selectSuppressionData": "extend",
        "selectTags": "extend",
        "objectids": "22395",
        "sortfield": ["clock", "eventid"],
        "sortorder": "DESC"
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "eventid": "20",
            "source": "0",
            "object": "0",
            "objectid": "22395",
```

```
            "clock": "1728658089",
            "value": "0",
            "acknowledged": "0",
            "ns": "461809482",
            "name": "La charge moyenne est trop élevée (charge par CPU supérieure à 1.5 pendant 5m)",
            "severity": "0",
            "r_eventid": "0",
            "c_eventid": "0",
            "correlationid": "0",
            "userid": "0",
            "suppressed": "0",
            "cause_eventid": "0",
            "acknowledges": [],
            "opdata": "Charges moyennes(1m 5m 15m): (1.6328 3.0522 2.2515), nombre de CPU : 2",
            "suppression_data": [],
            "tags": [
                {
                    "tag": "scope",
                    "value": "capacity"
                },
                {
                    "tag": "scope",
                    "value": "performance"
                },
                {
                    "tag": "component",
                    "value": "cpu"
                },
                {
                    "tag": "class",
                    "value": "os"
                },
                {
                    "tag": "target",
                    "value": "linux"
                }
            ],
            "urls": []
        },
        {
            "eventid": "4",
            "source": "0",
            "object": "0",
            "objectid": "22395",
            "clock": "1728657737",
            "value": "1",
            "acknowledged": "1",
            "ns": "460759366",
            "name": "La charge moyenne est trop élevée (charge par CPU supérieure à 1.5 pendant 5m)",
            "severity": "3",
            "r_eventid": "20",
            "c_eventid": "0",
            "correlationid": "0",
            "userid": "0",
            "suppressed": "1",
            "cause_eventid": "0",
            "acknowledges": [
                {
                    "acknowledgeid": "1",
                    "userid": "1",
                    "clock": "1728657938",
                    "message": "Testing environment. Please disregard this alert.",
```

```
                    "action": "38",
                    "old_severity": "0",
                    "new_severity": "0",
                    "suppress_until": "1728744338",
                    "taskid": "0",
                    "maintenanceid": "0",
                    "username": "Admin",
                    "name": "Zabbix",
                    "surname": "Administrator"
                }
            ],
            "opdata": "Charges moyennes(1m 5m 15m): (1.6328 3.0522 2.2515), nombre de CPU : 2",
            "suppression_data": [
                {
                    "maintenanceid": "0",
                    "suppress_until": "1728744338",
                    "userid": "1"
                }
            ],
            "tags": [
                {
                    "tag": "scope",
                    "value": "capacity"
                },
                {
                    "tag": "scope",
                    "value": "performance"
                },
                {
                    "tag": "component",
                    "value": "cpu"
                },
                {
                    "tag": "class",
                    "value": "os"
                },
                {
                    "tag": "target",
                    "value": "linux"
                }
            ],
            "urls": []
        }
    ],
    "id": 1
}
```


Récupération des événements par période

Récupérez tous les événements créés entre les 17 et 18 octobre 2012, dans l'ordre chronologique inverse.

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "event.get",
    "params": {
        "output": "extend",
        "time_from": "1350432000",
        "time_till": "1350518400",
        "sortfield": ["clock", "eventid"],
        "sortorder": "DESC"
    },
    "id": 1
```

```
}
```

Réponse :


```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "eventid": "20617",
            "source": "0",
            "object": "0",
            "objectid": "14282",
            "clock": "1350477816",
            "value": "1",
            "acknowledged": "0",
            "ns": "0",
            "name": "Moins de 25 % libres dans le cache d'historique",
            "severity": "3",
            "r_eventid": "0",
            "c_eventid": "0",
            "correlationid": "0",
            "userid": "0",
            "cause_eventid": "0",
            "opdata": "",
            "suppressed": "0",
            "urls": []
        },
        {
            "eventid": "20616",
            "source": "0",
            "object": "0",
            "objectid": "14281",
            "clock": "1350477814",
            "value": "0",
            "acknowledged": "0",
            "ns": "0",
            "name": "Les processus Zabbix trapper sont occupés à plus de 75 %",
            "severity": "0",
            "r_eventid": "0",
            "c_eventid": "0",
            "correlationid": "0",
            "userid": "0",
            "cause_eventid": "0",
            "opdata": "",
            "suppressed": "0",
            "urls": []
        },
        {
            "eventid": "20615",
            "source": "0",
            "object": "0",
            "objectid": "14281",
            "clock": "1350477541",
            "value": "1",
            "acknowledged": "0",
            "ns": "0",
            "name": "Les processus Zabbix trapper sont occupés à plus de 75 %",
            "severity": "3",
            "r_eventid": "20616",
            "c_eventid": "0",
            "correlationid": "0",
            "userid": "0",
            "cause_eventid": "0",
```

```
            "opdata": "",
            "suppressed": "0",
            "urls": []
        }
    ],
    "id": 1
}
```

Récupération des événements reconnus par un utilisateur spécifié

Récupération des événements reconnus par l'utilisateur avec l'ID=10

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "event.get",
    "params": {
        "output": "extend",
        "action": 2,
        "action_userids": [10],
        "selectAcknowledges": ["userid", "action"],
        "sortfield": ["eventid"],
        "sortorder": "DESC"
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "eventid": "503",
            "source": "0",
            "object": "0",
            "objectid": "23162",
            "clock": "1747212236",
            "value": "1",
            "acknowledged": "1",
            "ns": "413470863",
            "name": "Number of installed packages has been changed",
            "severity": "2",
            "r_eventid": "0",
            "c_eventid": "0",
            "correlationid": "0",
            "userid": "0",
            "suppressed": "0",
            "cause_eventid": "0",
            "acknowledges": [
                {
                    "userid": "10",
                    "action": "2"
                }
            ],
            "opdata": "",
            "urls": []
        }
    ],
    "id": 1
}
```

Récupération des principaux déclencheurs avec le nombre d'événements de problème

Récupérez les 5 principaux déclencheurs ayant les niveaux de gravité « Warning », « Average », « High » ou « Disaster », ainsi que

le nombre d'événements de problème sur une période donnée.

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "event.get",
    "params": {
        "countOutput": true,
        "groupBy": "objectid",
        "source": 0,
        "object": 0,
        "value": 1,
        "time_from": 1672531200,
        "time_till": 1677628800,
        "trigger_severities": [2, 3, 4, 5],
        "sortfield": ["rowscount"],
        "sortorder": "DESC",
        "limit": 5
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "objectid": "232124",
            "rowscount": "27"
        },
        {
            "objectid": "29055",
            "rowscount": "23"
        },
        {
            "objectid": "253731",
            "rowscount": "18"
        },
        {
            "objectid": "254062",
            "rowscount": "11"
        },
        {
            "objectid": "23216",
            "rowscount": "7"
        }
    ],
    "id": 1
}
```

Voir aussi

- Alerte
- Élément
- Hôte
- Règle LLD
- Service
- Déclencheur

Source

CEvent::get() dans *ui/include/classes/api/services/CEvent.php*.

Notation Signature de la méthode

Chaque page de méthode d'API contient la signature de la méthode.

Exemple : `object action.create(object/array actions)`

Signature	Description
`object`	Le type de données renvoyé par la méthode.
`action.create`	Le nom complet de la méthode.
`object/array`	Le type de données accepté par la méthode.
`actions`	Un bref résumé de ce que la méthode accepte.

Types de données

L'API Zabbix prend en charge les types de données suivants en entrée :

Type	Description
ID	Un identifiant unique utilisé pour référencer une entité.
boolean	Une valeur booléenne (`true` ou `false`).
flag	Une valeur considérée comme `true` si elle est transmise et différente de `null` ; sinon, la valeur est considérée comme `false`.
integer	Un nombre entier.
float	Un nombre à virgule flottante.
string	Une chaîne de caractères.
text	Une chaîne de texte plus longue.
timestamp	Un horodatage Unix.
array	Une séquence ordonnée de valeurs (un tableau simple).
object	Un tableau associatif.
query	Une valeur qui définit les données à renvoyer. La valeur peut être définie comme un tableau de noms de propriétés (pour ne renvoyer que des propriétés spécifiques), ou comme l'une des valeurs prédéfinies suivantes : `extend` - renvoie toutes les propriétés de l'objet ; `count` - renvoie le nombre d'enregistrements récupérés, pris en charge uniquement par certaines sous-sélections.

Attention:
L'API Zabbix renvoie toujours uniquement des valeurs sous forme de chaînes ou de tableaux.

Comportement des propriétés

Certaines propriétés de l'objet sont marquées par de courtes étiquettes décrivant leur comportement. Les étiquettes suivantes sont utilisées :

- ***lecture seule*** - la valeur de la propriété est définie automatiquement et ne peut pas être définie ni modifiée par l'utilisateur, même dans certaines conditions spécifiques (par ex., *lecture seule* pour les objets hérités ou les objets découverts) ;
- ***écriture seule*** - la valeur de la propriété peut être définie, mais ne peut pas être consultée ensuite ;
- ***constante*** - la valeur de la propriété peut être définie lors de la création d'un objet, mais ne peut pas être modifiée par la suite ;
- ***prise en charge*** - la valeur de la propriété n'est pas obligatoire, mais elle peut être définie dans certaines conditions spécifiques (par ex., *prise en charge* si `type` est défini sur "Simple check", "External check", "SSH agent", "TELNET agent" ou "HTTP agent") ; notez toutefois que les propriétés *prises en charge* peuvent toujours être définies sur leurs valeurs par défaut quelles que soient les conditions ;
- ***obligatoire*** - la valeur de la propriété doit être définie pour toutes les opérations (à l'exception des opérations get) ou dans certaines conditions spécifiques (par ex., *obligatoire* pour les opérations de création ; *obligatoire* si `operationtype` est défini sur "global script" et que `opcommand_hst` n'est pas défini).

Note:
Pour les opérations de mise à jour, une propriété est considérée comme « définie » lorsqu'elle est définie pendant l'opération de mise à jour.

Les propriétés qui ne sont pas marquées par des étiquettes sont facultatives.

Comportement des paramètres

Certains paramètres d'opération sont marqués par de courtes étiquettes afin de décrire leur comportement pour l'opération. Les étiquettes suivantes sont utilisées :

- ***lecture seule*** - la valeur du paramètre est définie automatiquement et ne peut pas être définie ni modifiée par l'utilisateur, même dans certaines conditions spécifiques (par exemple, *lecture seule* pour les objets hérités ou les objets découverts) ;
- ***écriture seule*** - la valeur du paramètre peut être définie, mais ne peut pas être consultée ensuite ;
- ***pris en charge*** - la valeur du paramètre n'est pas obligatoire, mais peut être définie dans certaines conditions spécifiques (par exemple, *pris en charge* si `operating_mode` de l'objet Proxy est défini sur « passive proxy ») ; notez toutefois que les paramètres *pris en charge* peuvent toujours être définis sur leurs valeurs par défaut, quelles que soient les conditions ;
- ***obligatoire*** - la valeur du paramètre doit être définie.

Les paramètres qui ne sont pas marqués par des étiquettes sont facultatifs.

Valeur d'ID réservée « 0 » La valeur d'ID réservée « 0 » peut être utilisée pour filtrer des éléments et pour supprimer des objets référencés. Par exemple, pour supprimer un proxy référencé d'un hôte, proxyid doit être défini sur 0 (« proxyid »: « 0 ») ou, pour filtrer les hôtes surveillés par le serveur, l'option proxyids doit être définie sur 0 (« proxyids »: « 0 »).

Paramètres communs des méthodes "get" Les paramètres suivants sont pris en charge par toutes les méthodes `get` :

Paramètre	Type	Description
countOutput	boolean	Renvoyer le nombre d'enregistrements dans le résultat au lieu des données réelles.
editable	boolean	Si défini sur `true`, renvoyer uniquement les objets pour lesquels l'utilisateur dispose des droits d'écriture. Par défaut : `false`.
excludeSearch	boolean	Renvoyer les résultats qui ne correspondent pas aux critères indiqués dans le paramètre `search`.
filter	object	Renvoyer uniquement les résultats qui correspondent exactement au filtre donné. Accepte un objet, où les clés sont des noms de propriétés (par exemple, les propriétés de l'objet Host dans `host.get`, les propriétés de l'objet Item dans `item.get`, etc.), et les valeurs sont soit une valeur unique, soit un tableau de valeurs à comparer. Ne prend pas en charge les propriétés du type de données `text`. Notez que certaines méthodes ont une fonctionnalité spécifique pour ce paramètre, décrite sur la page de la méthode (par exemple, le paramètre `filter` dans host.get prend également en charge les propriétés de l'interface de l'hôte).
limit	integer	Limiter le nombre d'enregistrements renvoyés.
output	query	Propriétés de l'objet à renvoyer. Notez que l'ID de l'objet (c'est-à-dire `hostid`, `itemid`, etc.) est toujours inclus dans la réponse, même s'il n'est pas spécifié dans le paramètre `output`. Par défaut : `extend`.
preservekeys	boolean	Utiliser les ID comme clés dans le tableau résultant.

Paramètre	Type	Description
search	object	Renvoyer les résultats qui correspondent au modèle donné (sans distinction de casse). Accepte un objet, où les clés sont des noms de propriétés (par exemple, les propriétés de l'objet Host dans `host.get`, les propriétés de l'objet Item dans `item.get`, etc.), et les valeurs sont des chaînes à rechercher. Si aucune option supplémentaire n'est fournie, cela effectuera une recherche `LIKE "%…%"`. Prend en charge uniquement les propriétés des types de données `string` et `text`. Notez que certaines méthodes ont une fonctionnalité spécifique pour ce paramètre, décrite sur la page de la méthode (par exemple, le paramètre `search` dans host.get prend également en charge les propriétés de l'interface de l'hôte).
searchByAny	boolean	Si défini sur `true`, renvoyer les résultats qui correspondent à l'un des critères indiqués dans le paramètre `filter` ou `search`, au lieu de tous les critères. Par défaut : `false`.
searchWildcardsEnabled	boolean	Si défini sur `true`, active l'utilisation de "*" comme caractère générique dans le paramètre `search`. Par défaut : `false`.
sortfield	string/array	Trier le résultat selon les propriétés indiquées. Reportez-vous à la description d'une méthode get API spécifique pour obtenir la liste des propriétés pouvant être utilisées pour le tri. Les macros ne sont pas développées avant le tri. Si aucune valeur n'est spécifiée, les données seront renvoyées sans tri.
sortorder	string/array	Ordre de tri. Si un tableau est transmis, chaque valeur sera associée à la propriété correspondante indiquée dans le paramètre `sortfield`. Valeurs possibles : `ASC` - *(par défaut)* croissant ; `DESC` - décroissant.
startSearch	boolean	Le paramètre `search` comparera le début des champs, c'est-à-dire effectuera à la place une recherche `LIKE "…%"`. Ignoré si `searchWildcardsEnabled` est défini sur `true`.

Indicateurs d'origine des entités Les méthodes `get` renvoient une propriété `flags` pour les entités liées à la découverte de bas niveau (règle LLD/prototype de règle LLD, élément/prototype d'élément, etc.). Cette propriété est utile pour indiquer si l'entité a été découverte ou non, car la modification des entités découvertes est limitée.

La propriété `flags` renvoie un résultat basé sur une combinaison (opération « + ») de ces valeurs :

Value	Description
0	Entité de base (élément, déclencheur, graphe, hôte)
1	Règle de découverte de bas niveau
2	Tout prototype (prototype d'élément, prototype de déclencheur, prototype de règle LLD, etc.)
4	Entité découverte (élément, déclencheur, graphe, hôte, règle LLD découverts)

La valeur **combinée** renvoyée par la propriété `flags` peut être :

Value	Combination of	Description
0	0	Entité simple (élément, déclencheur, graphe, hôte).
2	2	Prototype d'entité (prototype d'élément, prototype de déclencheur, etc.).
6	2+4	Élément, déclencheur, graphe, hôte découverts (convertis à partir d'un prototype).
1	1	Règle de découverte de bas niveau.
3	1+2	Prototype de règle de découverte de bas niveau.

Value	Combination of	Description
5	1+4	Règle de découverte de bas niveau découverte (convertie à partir d'un prototype).
7	1+2+4	Prototype de règle de découverte de bas niveau découvert.

Exemples Vérification des permissions de l'utilisateur

L'utilisateur a-t-il la permission d'écrire sur les hôtes dont les noms commencent par "MySQL" ou "Linux" ?

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "host.get",
    "params": {
        "countOutput": true,
        "search": {
            "host": ["MySQL", "Linux"]
        },
        "editable": true,
        "startSearch": true,
        "searchByAny": true
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": "0",
    "id": 1
}
```

Note:
Un résultat égal à zéro signifie qu'aucun hôte ne dispose de permissions de lecture/écriture.

Comptage des non-correspondances

Comptez le nombre d'hôtes dont les noms ne contiennent pas la sous-chaîne "ubuntu"

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "host.get",
    "params": {
        "countOutput": true,
        "search": {
            "host": "ubuntu"
        },
        "excludeSearch": true
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": "44",
    "id": 1
}
```

Recherche d'hôtes à l'aide de caractères génériques

Recherchez les hôtes dont le nom contient le mot ”server” et qui ont des ports d’interface ”10050” ou ”10071”. Triez le résultat par nom d’hôte dans l’ordre décroissant et limitez-le à 5 hôtes.

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "host.get",
    "params": {
        "output": ["hostid", "host"],
        "selectInterfaces": ["port"],
        "filter": {
            "port": ["10050", "10071"]
        },
        "search": {
            "host": "*server*"
        },
        "searchWildcardsEnabled": true,
        "searchByAny": true,
        "sortfield": "host",
        "sortorder": "DESC",
        "limit": 5
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "hostid": "50003",
            "host": "WebServer-Tomcat02",
            "interfaces": [
                {
                    "port": "10071"
                }
            ]
        },
        {
            "hostid": "50005",
            "host": "WebServer-Tomcat01",
            "interfaces": [
                {
                    "port": "10071"
                }
            ]
        },
        {
            "hostid": "50004",
            "host": "WebServer-Nginx",
            "interfaces": [
                {
                    "port": "10071"
                }
            ]
        },
        {
            "hostid": "99032",
            "host": "MySQL server 01",
            "interfaces": [
                {
                    "port": "10050"
                }
```

```
            ]
        },
        {
            "hostid": "99061",
            "host": "Linux server 01",
            "interfaces": [
                {
                    "port": "10050"
                }
            ]
        }
    ],
    "id": 1
}
```

Recherche d'hôtes à l'aide de caractères génériques avec "preservekeys"

Si vous ajoutez le paramètre "preservekeys" à la requête précédente, le résultat est renvoyé sous forme de tableau associatif, où les clés sont les identifiants des objets.

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "host.get",
    "params": {
        "output": ["hostid", "host"],
        "selectInterfaces": ["port"],
        "filter": {
            "port": ["10050", "10071"]
        },
        "search": {
            "host": "*server*"
        },
        "searchWildcardsEnabled": true,
        "searchByAny": true,
        "sortfield": "host",
        "sortorder": "DESC",
        "limit": 5,
        "preservekeys": true
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "50003": {
            "hostid": "50003",
            "host": "WebServer-Tomcat02",
            "interfaces": [
                {
                    "port": "10071"
                }
            ]
        },
        "50005": {
            "hostid": "50005",
            "host": "WebServer-Tomcat01",
            "interfaces": [
                {
                    "port": "10071"
                }
```

```
            ]
        },
        "50004": {
            "hostid": "50004",
            "host": "WebServer-Nginx",
            "interfaces": [
                {
                    "port": "10071"
                }
            ]
        },
        "99032": {
            "hostid": "99032",
            "host": "MySQL server 01",
            "interfaces": [
                {
                    "port": "10050"
                }
            ]
        },
        "99061": {
            "hostid": "99061",
            "host": "Linux server 01",
            "interfaces": [
                {
                    "port": "10050"
                }
            ]
        }
    },
    "id": 1
}
```

Annexe 2. Modifications de la version 7.4 à la version 8.0

Changements incompatibles

Autres modifications et corrections de bugs

Annexe 3. Modifications dans la version 8.0

8.0.1 Cette version n'est pas encore publiée.

Extensions

Aperçu Bien que Zabbix offre une multitude de fonctionnalités, il reste toujours possible d'y ajouter des capacités supplémentaires. Les extensions constituent un moyen pratique de modifier et d'améliorer les capacités de supervision de Zabbix sans modifier son code source.

Vous pouvez étendre les fonctionnalités de Zabbix soit en utilisant les options d'extension intégrées (éléments trapper, paramètres utilisateur, etc.), soit en utilisant ou en créant des extensions personnalisées (modules chargeables, plugins, etc.).

Cette section fournit un aperçu avec des références vers toutes les options permettant d'étendre Zabbix.

Collecte de données avec des commandes personnalisées Éléments trapper

Les éléments trapper sont des éléments qui acceptent des données entrantes au lieu de les interroger. Les éléments trapper sont utiles pour envoyer des données spécifiques au serveur Zabbix ou au proxy, par exemple des données périodiques de disponibilité et de performance dans le cas de scripts utilisateur de longue durée. L'envoi de données au serveur Zabbix ou au proxy est possible à l'aide de l'utilitaire Zabbix sender ou du protocole Zabbix sender. L'envoi de données au serveur Zabbix est également possible à l'aide de la méthode API `history.push`.

Vérifications externes

Une vérification externe est un élément permettant d'exécuter des contrôles en lançant un exécutable, par exemple un script shell ou un binaire.

Les vérifications externes sont exécutées par le serveur Zabbix ou le proxy (lorsque l'hôte est supervisé par un proxy), et ne nécessitent pas qu'un agent soit exécuté sur l'hôte supervisé.

Paramètres utilisateur

Un paramètre utilisateur est une commande définie par l'utilisateur (associée à une clé définie par l'utilisateur) qui, lorsqu'elle est exécutée, peut récupérer les données dont vous avez besoin depuis l'hôte sur lequel l'agent Zabbix est en cours d'exécution. Les paramètres utilisateur sont utiles pour configurer des éléments d'agent ou d'agent 2 qui ne sont pas prédéfinis dans Zabbix.

Éléments `system.run[]` de l'agent Zabbix

`system.run[]` de l'agent Zabbix élément est un élément correspondant à une commande définie par l'utilisateur (associée à une clé prédéfinie `system.run[]`, par exemple, `system.run[myscript.sh]`) qui peut être exécutée sur l'hôte où l'agent Zabbix est exécuté.

Note:
Les éléments `system.run[]` sont désactivés par défaut et, s'ils sont utilisés, doivent être activés (autorisés) et définis dans le fichier de configuration de l'agent Zabbix ou de Zabbix agent 2 (paramètre de configuration `AllowKey` ou `AllowKeyRegexp`).

Attention:
Les commandes définies par l'utilisateur dans des éléments tels que les vérifications externes, les paramètres utilisateur et les éléments d'agent Zabbix `system.run[]` sont exécutées par l'utilisateur du système d'exploitation utilisé pour exécuter les composants Zabbix. Pour exécuter ces commandes, cet utilisateur doit disposer des autorisations nécessaires.

Éléments de type HTTP agent

Un élément de type HTTP agent est un élément permettant d'exécuter des requêtes de données via HTTP/HTTPS. Les éléments de type HTTP agent sont utiles pour envoyer des requêtes à des points de terminaison HTTP afin de récupérer des données auprès de services tels qu'*Elasticsearch* et *OpenWeatherMap*, pour vérifier l'état de l'API Zabbix ou celui d'un serveur web Apache ou NGINX, etc. Les éléments de type HTTP agent (avec la réception des données activée) peuvent également fonctionner comme des éléments trapper.

Éléments de script

Un élément de script est un élément permettant d'exécuter du code JavaScript défini par l'utilisateur afin de récupérer des données via HTTP/HTTPS. Les éléments de script sont utiles lorsque les fonctionnalités fournies par les éléments d'agent HTTP ne suffisent pas. Par exemple, dans des scénarios exigeants de collecte de données nécessitant plusieurs étapes ou une logique complexe, un élément de script peut être configuré pour effectuer un appel HTTP, puis traiter les données reçues, puis transmettre la valeur transformée à un second appel HTTP.

Note:
Les éléments d'agent HTTP et les éléments de script sont pris en charge par le serveur et le proxy Zabbix, et ne nécessitent pas qu'un agent soit exécuté sur l'hôte surveillé.

Extensions avancées Modules chargeables

Les modules chargeables, écrits en C, constituent une option polyvalente et axée sur les performances pour étendre les fonctionnalités des composants Zabbix (server, proxy, agent) sur les plateformes UNIX. Un module chargeable est essentiellement une bibliothèque partagée utilisée par le démon Zabbix et chargée au démarrage. La bibliothèque doit contenir certaines fonctions, afin qu'un processus Zabbix puisse détecter que le fichier est bien un module qu'il peut charger et utiliser.

Les modules chargeables présentent un certain nombre d'avantages, notamment la possibilité d'ajouter de nouvelles métriques ou d'implémenter toute autre logique (par exemple, l'export des données d'historique de Zabbix), d'excellentes performances, ainsi

que la possibilité de développer, d'utiliser et de partager les fonctionnalités qu'ils fournissent. Cela contribue à une maintenance sans difficulté et permet de fournir plus facilement de nouvelles fonctionnalités, indépendamment du code de base de Zabbix.

Les modules chargeables sont particulièrement utiles dans une configuration de supervision complexe. Lors de la supervision de systèmes embarqués, avec un grand nombre de paramètres supervisés ou des scripts lourds avec une logique complexe ou un long temps de démarrage, des extensions telles que les paramètres utilisateur, les éléments d'agent Zabbix `system.run[]` et les vérifications externes auront un impact sur les performances. Les modules chargeables offrent un moyen d'étendre les fonctionnalités de Zabbix sans sacrifier les performances.

Plugins

Les plugins étendent les capacités de supervision de Zabbix agent 2. Écrits dans le langage de programmation Go, ils offrent une alternative aux modules chargeables (écrits en C).

Il est possible de développer votre propre plugin (collecteur de données indépendant) pour Zabbix agent 2. Voir le centre de développement.

Personnalisation des alertes Webhooks

Un webhook est un type de média Zabbix qui offre la possibilité d'étendre les capacités d'alerte de Zabbix à des logiciels externes tels que des systèmes de helpdesk, des chats ou des messageries. À l'instar des éléments script, les webhooks sont utiles pour effectuer des appels HTTP à l'aide de code JavaScript personnalisé, par exemple pour envoyer des notifications vers différentes plateformes telles que Microsoft Teams, Discord et Jira. Il est également possible de renvoyer certaines données (par exemple, concernant les tickets de helpdesk créés) qui sont ensuite affichées dans Zabbix.

Les webhooks existants sont disponibles dans le dépôt Git de Zabbix. Pour le développement de webhooks personnalisés, consultez les Directives de développement des webhooks.

Scripts d'alerte

Un script d'alerte est un type de média de Zabbix qui offre la possibilité de créer une autre méthode (script) pour gérer les alertes Zabbix. Les scripts d'alerte sont utiles si les types de média existants pour l'envoi d'alertes dans Zabbix ne vous conviennent pas.

Personnalisation du frontend Thèmes personnalisés

Il est possible de modifier l'apparence visuelle de l'interface web de Zabbix en utilisant des thèmes personnalisés. Consultez les instructions sur la création et l'application de vos propres thèmes.

Modules d'interface web

Les modules d'interface web offrent une possibilité d'étendre les fonctionnalités de l'interface web de Zabbix en ajoutant des modules tiers ou en développant les vôtres. Avec les modules d'interface web, vous pouvez ajouter de nouveaux éléments de menu, leurs vues respectives, actions, etc.

Scripts globaux Un script global est un ensemble de commandes défini par l'utilisateur qui peut être exécuté sur une cible de supervision (par l'interpréteur shell (/bin/sh)), selon la portée configurée et les autorisations de l'utilisateur. Les scripts globaux peuvent être configurés pour les actions suivantes :

- Opération d'action
- Action manuelle sur l'hôte
- Action manuelle sur l'événement

Les scripts globaux sont utiles dans de nombreux cas. Par exemple, s'ils sont configurés pour des opérations d'action ou des actions manuelles sur l'hôte, vous pouvez utiliser des scripts globaux pour exécuter automatiquement ou manuellement des commandes distantes telles que le redémarrage d'une application (serveur web, middleware, CRM, etc.) ou la libération d'espace disque (suppression des fichiers anciens, nettoyage de `/tmp`, etc.). Autre exemple : s'ils sont configurés pour des actions manuelles sur les événements, vous pouvez utiliser des scripts globaux pour gérer des tickets d'incident dans des systèmes externes.

Les scripts globaux peuvent être exécutés par le serveur Zabbix, le proxy ou l'agent.

> **Attention:**
> Les commandes définies par l'utilisateur sont exécutées par l'utilisateur du système d'exploitation utilisé pour exécuter les composants Zabbix. Pour exécuter ces commandes, cet utilisateur doit disposer des autorisations nécessaires.

API Zabbix L'API Zabbix est une API basée sur HTTP qui fait partie du frontend Zabbix. Avec l'API Zabbix, vous pouvez effectuer l'une des opérations suivantes :

- Récupérer et modifier par programmation la configuration de Zabbix.
- Importer et exporter la configuration de Zabbix.

- Accéder aux données historiques et aux tendances de Zabbix.
- Configurer des applications pour qu'elles fonctionnent avec Zabbix.
- Intégrer Zabbix à des logiciels tiers.
- Automatiser les tâches de routine.

L'API Zabbix se compose d'un grand nombre de méthodes, regroupées nominalement en API distinctes. Chaque méthode exécute une tâche spécifique. Pour connaître les méthodes disponibles, ainsi qu'un aperçu des fonctions fournies par l'API Zabbix, consultez la Référence des méthodes de l'API Zabbix.

Modules chargeables

Vue d'ensemble

Les modules chargeables offrent une option axée sur les performances pour étendre les fonctionnalités de Zabbix.

Vous pouvez étendre les fonctionnalités de Zabbix de nombreuses façons, par exemple avec les paramètres utilisateur, les vérifications externes et les éléments d'agent Zabbix system.run. Ces méthodes fonctionnent très bien, mais présentent un inconvénient majeur, à savoir fork(). Zabbix doit créer un nouveau processus par fork chaque fois qu'il traite une métrique utilisateur, ce qui n'est pas bon pour les performances. Ce n'est généralement pas un gros problème, mais cela peut devenir un problème sérieux lors de la supervision de systèmes embarqués, lorsqu'un grand nombre de paramètres sont supervisés ou lorsque des scripts lourds avec une logique complexe ou un long temps de démarrage sont utilisés.

La prise en charge des modules chargeables permet d'étendre l'agent, le serveur et le proxy Zabbix sans sacrifier les performances.

Un module chargeable est essentiellement une bibliothèque partagée utilisée par le démon Zabbix et chargée au démarrage. La bibliothèque doit contenir certaines fonctions, afin qu'un processus Zabbix puisse détecter que le fichier est bien un module qu'il peut charger et utiliser.

Les modules chargeables présentent un certain nombre d'avantages. D'excellentes performances et la possibilité d'implémenter n'importe quelle logique sont très importantes, mais l'avantage le plus important est peut-être la possibilité de développer, d'utiliser et de partager des modules Zabbix. Cela contribue à une maintenance sans difficulté et aide à fournir de nouvelles fonctionnalités plus facilement et indépendamment de la base de code principale de Zabbix.

La licence des modules et leur distribution sous forme binaire sont régies par la licence AGPL-3.0 (les modules sont liés à Zabbix à l'exécution et utilisent les en-têtes Zabbix ; l'ensemble du code Zabbix est sous licence AGPL-3.0 depuis Zabbix 7.0). La compatibilité binaire n'est pas garantie par Zabbix.

La stabilité de l'API des modules est garantie pendant un cycle de version LTS (Long Term Support) de Zabbix. La stabilité de l'API Zabbix n'est pas garantie (techniquement, il est possible d'appeler des fonctions internes de Zabbix depuis un module, mais rien ne garantit que de tels modules fonctionneront).

API de module

Pour qu'une bibliothèque partagée soit traitée comme un module Zabbix, elle doit implémenter et exporter plusieurs fonctions. Il existe actuellement six fonctions dans l'API de module Zabbix, dont une seule est obligatoire et les cinq autres sont facultatives.

Interface obligatoire

La seule fonction obligatoire est `zbx_module_api_version()` :

```
int zbx_module_api_version(void);
```

Cette fonction doit renvoyer la version de l'API implémentée par ce module et, pour que le module puisse être chargé, cette version doit correspondre à la version de l'API du module prise en charge par Zabbix. La version de l'API du module prise en charge par Zabbix est `ZBX_MODULE_API_VERSION`. Cette fonction doit donc renvoyer cette constante. L'ancienne constante `ZBX_MODULE_API_VERSION_ONE`, utilisée à cette fin, est désormais définie comme étant égale à `ZBX_MODULE_API_VERSION` afin de préserver la compatibilité des sources, mais son utilisation n'est pas recommandée.

Interface facultative

Les fonctions facultatives sont `zbx_module_init()`, `zbx_module_item_list()`, `zbx_module_item_timeout()`, `zbx_module_histor`
et `zbx_module_uninit()` :

```
int zbx_module_init(void);
```

Cette fonction doit effectuer l'initialisation nécessaire du module, le cas échéant. En cas de succès, elle doit renvoyer `ZBX_MODULE_OK`. Sinon, elle doit renvoyer `ZBX_MODULE_FAIL`. Dans ce dernier cas, Zabbix ne démarrera pas.

```
ZBX_METRIC  *zbx_module_item_list(void);
```

Cette fonction doit renvoyer une liste des éléments pris en charge par le module. Chaque élément est défini dans une structure ZBX_METRIC, voir la section ci-dessous pour plus de détails. La liste se termine par une structure ZBX_METRIC dont le champ "key" vaut NULL.

```
void    zbx_module_item_timeout(int timeout);
```

Si le module exporte zbx_module_item_list(), cette fonction est utilisée par Zabbix pour spécifier les paramètres de délai d'attente dans le fichier de configuration Zabbix que les vérifications d'éléments implémentées par le module doivent respecter. Ici, le paramètre "timeout" est exprimé en secondes.

```
ZBX_HISTORY_WRITE_CBS   zbx_module_history_write_cbs(void);
```

Cette fonction doit renvoyer les fonctions de rappel que le serveur Zabbix utilisera pour exporter l'historique des différents types de données. Les fonctions de rappel sont fournies sous forme de champs de la structure ZBX_HISTORY_WRITE_CBS; les champs peuvent être NULL si le module ne s'intéresse pas à l'historique d'un certain type.

```
int zbx_module_uninit(void);
```

Cette fonction doit effectuer la désinitialisation nécessaire, le cas échéant, comme la libération des ressources allouées, la fermeture des descripteurs de fichiers, etc.

Toutes les fonctions sont appelées une fois au démarrage de Zabbix lors du chargement du module, à l'exception de zbx_module_uninit(), qui est appelée une fois à l'arrêt de Zabbix lors du déchargement du module.

Définition des éléments

Chaque élément est défini dans une structure ZBX_METRIC :

```
typedef struct
{
    char        *key;
    unsigned    flags;
    int     (*function)();
    char        *test_param;
}
ZBX_METRIC;
```

Ici, **key** est la clé de l'élément (par exemple, "dummy.random"), **flags** vaut soit CF_HAVEPARAMS soit 0 (selon que l'élément accepte ou non des paramètres), **function** est une fonction C qui implémente l'élément (par exemple, zbx_module_dummy_random), et **test_param** est la liste des paramètres à utiliser lorsque l'agent Zabbix est démarré avec l'option -p (par exemple, "1,1000", peut être NULL). Un exemple de définition peut ressembler à ceci :

```
static ZBX_METRIC keys[] =
{
    { "dummy.random", CF_HAVEPARAMS, zbx_module_dummy_random, "1,1000" },
    { NULL }
}
```

Chaque fonction qui implémente un élément doit accepter deux paramètres pointeur, le premier de type AGENT_REQUEST et le second de type AGENT_RESULT :

```
int zbx_module_dummy_random(AGENT_REQUEST *request, AGENT_RESULT *result)
{
    ...

    SET_UI64_RESULT(result, from + rand() % (to - from + 1));

    return SYSINFO_RET_OK;
}
```

Ces fonctions doivent renvoyer SYSINFO_RET_OK si la valeur de l'élément a été obtenue avec succès. Sinon, elles doivent renvoyer SYSINFO_RET_FAIL. Voir l'exemple du module "dummy" ci-dessous pour plus de détails sur la manière d'obtenir des informations à partir de AGENT_REQUEST et de définir des informations dans AGENT_RESULT.

Fourniture de callbacks d'export de l'historique

> **Attention:**
> L'export de l'historique via un module n'est plus pris en charge par le proxy Zabbix.

Le module peut spécifier des fonctions pour exporter les données d'historique par type: Numeric (float), Numeric (unsigned), Character, Text et Log:

```
typedef struct
{
    void    (*history_float_cb)(const ZBX_HISTORY_FLOAT *history, int history_num);
    void    (*history_integer_cb)(const ZBX_HISTORY_INTEGER *history, int history_num);
    void    (*history_string_cb)(const ZBX_HISTORY_STRING *history, int history_num);
    void    (*history_text_cb)(const ZBX_HISTORY_TEXT *history, int history_num);
    void    (*history_log_cb)(const ZBX_HISTORY_LOG *history, int history_num);
}
ZBX_HISTORY_WRITE_CBS;
```

Chacun d'eux doit prendre comme arguments le tableau `history` contenant `history_num` éléments. Selon le type de données d'historique à exporter, `history` est respectivement un tableau des structures suivantes:

```
typedef struct
{
    zbx_uint64_t    itemid;
    int     clock;
    int     ns;
    double      value;
}
ZBX_HISTORY_FLOAT;

typedef struct
{
    zbx_uint64_t    itemid;
    int     clock;
    int     ns;
    zbx_uint64_t    value;
}
ZBX_HISTORY_INTEGER;

typedef struct
{
    zbx_uint64_t    itemid;
    int     clock;
    int     ns;
    const char  *value;
}
ZBX_HISTORY_STRING;

typedef struct
{
    zbx_uint64_t    itemid;
    int     clock;
    int     ns;
    const char  *value;
}
ZBX_HISTORY_TEXT;

typedef struct
{
    zbx_uint64_t    itemid;
    int     clock;
    int     ns;
    const char  *value;
    const char  *source;
    int     timestamp;
    int     logeventid;
    int     severity;
}
ZBX_HISTORY_LOG;
```

Les callbacks seront utilisés par les processus de synchronisation de l'historique du serveur Zabbix à la fin de la procédure de synchronisation de l'historique, après que les données ont été écrites dans la base de données Zabbix et enregistrées dans le cache des valeurs.

> **Attention:**
> En cas d'erreur interne dans le module d'export de l'historique, il est recommandé que le module soit écrit de manière à ne pas bloquer l'ensemble de la supervision jusqu'à sa récupération, mais à rejeter les données à la place et à permettre au serveur Zabbix de continuer à fonctionner.

Compilation des modules

Les modules sont actuellement destinés à être compilés à l'intérieur de l'arborescence des sources de Zabbix, car l'API des modules dépend de certaines structures de données définies dans les en-têtes de Zabbix.

L'en-tête le plus important pour les modules chargeables est `include/module.h`, qui définit ces structures de données. Les autres en-têtes système nécessaires au bon fonctionnement de `include/module.h` sont `stdlib.h` et `stdint.h`.

Fort de ces informations, tout est prêt pour compiler le module. Le module doit inclure `stdlib.h`, `stdint.h` et `module.h`, et le script de compilation doit s'assurer que ces fichiers se trouvent dans le chemin d'inclusion. Voir l'exemple du module "dummy" ci-dessous pour plus de détails.

Un autre en-tête utile est `include/zbxcommon.h`, qui définit la fonction `zabbix_log()`, pouvant être utilisée à des fins de journalisation et de débogage.

Paramètres de configuration

L'agent Zabbix, le serveur et le proxy prennent en charge deux paramètres pour gérer les modules :

- `LoadModulePath` - chemin complet vers l'emplacement des modules chargeables
- `LoadModule` - module(s) à charger au démarrage.

Les modules doivent se trouver dans un répertoire spécifié par `LoadModulePath`, ou le chemin doit précéder le nom du module. Si le chemin qui précède est absolu (commence par /), `LoadModulePath` est ignoré. Plusieurs paramètres `LoadModule` sont autorisés.

Par exemple, pour étendre l'agent Zabbix, nous pouvons ajouter les paramètres suivants :

```
LoadModulePath=/usr/local/lib/zabbix/agent/
LoadModule=mariadb.so
LoadModule=apache.so
LoadModule=kernel.so
LoadModule=/usr/local/lib/zabbix/dummy.so
```

Au démarrage de l'agent, les modules mariadb.so, apache.so et kernel.so seront chargés depuis le répertoire `/usr/local/lib/zabbix/agen` tandis que dummy.so sera chargé depuis `/usr/local/lib/zabbix`. L'agent ne démarrera pas si un module est manquant, si les autorisations sont incorrectes ou si une bibliothèque partagée n'est pas un module Zabbix.

Configuration de l'interface

Les modules chargeables sont pris en charge par Zabbix agent, serveur et proxy. Par conséquent, le type d'élément dans l'interface Zabbix dépend de l'endroit où le module est chargé. Si le module est chargé dans l'agent, alors le type d'élément doit être "Zabbix agent" ou "Zabbix agent (active)". Si le module est chargé dans le serveur ou le proxy, alors le type d'élément doit être "Simple check".

L'export de l'historique via les modules Zabbix ne nécessite aucune configuration de l'interface. Si le module est chargé avec succès par le serveur et fournit la fonction `zbx_module_history_write_cbs()` qui renvoie au moins une fonction de rappel non NULL, alors l'export de l'historique sera activé automatiquement.

Module dummy

Zabbix inclut un module d'exemple écrit en langage C. Le module se trouve dans `src/modules/dummy` :

```
alex@alex:~trunk/src/modules/dummy$ ls -l
-rw-rw-r-- 1 alex alex 9019 Apr 24 17:54 dummy.c
-rw-rw-r-- 1 alex alex   67 Apr 24 17:54 Makefile
-rw-rw-r-- 1 alex alex  245 Apr 24 17:54 README
```

Le module est bien documenté ; il peut être utilisé comme modèle pour vos propres modules.

Après avoir exécuté `./configure` à la racine de l'arborescence des sources de Zabbix, comme décrit ci-dessus, exécutez simplement `make` afin de compiler `dummy.so`.

```c
/*
** Zabbix
** Copyright (C) 2001-2020 Zabbix SIA
**
** This program is free software; you can redistribute it and/or modify
** it under the terms of the GNU General Public License as published by
** the Free Software Foundation; either version 2 of the License, or
** (at your option) any later version.
**
** This program is distributed in the hope that it will be useful,
** but WITHOUT ANY WARRANTY; without even the implied warranty of
** MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
** GNU General Public License for more details.
**
** You should have received a copy of the GNU General Public License
** along with this program; if not, write to the Free Software
** Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA  02110-1301, USA.
**/

###include <stdlib.h>
###include <string.h>
###include <time.h>
###include <stdint.h>

###include "module.h"

/* la variable conserve le paramètre de délai d'attente pour le traitement des éléments */
static int  item_timeout = 0;

/* le module DOIT définir les fonctions internes comme statiques et utiliser une convention de nommage dif
/* symboles internes de Zabbix (zbx_*) et des fonctions de l'API des modules chargeables (zbx_module_*) af
static int  dummy_ping(AGENT_REQUEST *request, AGENT_RESULT *result);
static int  dummy_echo(AGENT_REQUEST *request, AGENT_RESULT *result);
static int  dummy_random(AGENT_REQUEST *request, AGENT_RESULT *result);

static ZBX_METRIC keys[] =
/*  KEY         FLAG        FUNCTION    TEST PARAMETERS */
{
    {"dummy.ping",      0,      dummy_ping, NULL},
    {"dummy.echo",      CF_HAVEPARAMS,  dummy_echo, "a message"},
    {"dummy.random",    CF_HAVEPARAMS,  dummy_random,   "1,1000"},
    {NULL}
};

/******************************************************************************
 *                                                                            *
 * Function: zbx_module_api_version                                           *
 *                                                                            *
 * Purpose: returns version number of the module interface                    *
 *                                                                            *
 * Return value: ZBX_MODULE_API_VERSION - version of module.h module is       *
 *               compiled with, in order to load module successfully Zabbix   *
 *               MUST be compiled with the same version of this header file   *
 *                                                                            *
 ******************************************************************************/
int zbx_module_api_version(void)
{
    return ZBX_MODULE_API_VERSION;
}

/******************************************************************************
 *                                                                            *
```

```
 * Function: zbx_module_item_timeout                                          *
 *                                                                            *
 * Purpose: set timeout value for processing of items                         *
 *                                                                            *
 * Parameters: timeout - timeout in seconds, 0 - no timeout set               *
 *                                                                            *
 ******************************************************************************/
void    zbx_module_item_timeout(int timeout)
{
    item_timeout = timeout;
}

/******************************************************************************
 *                                                                            *
 * Function: zbx_module_item_list                                             *
 *                                                                            *
 * Purpose: returns list of item keys supported by the module                 *
 *                                                                            *
 * Return value: list of item keys                                            *
 *                                                                            *
 ******************************************************************************/
ZBX_METRIC  *zbx_module_item_list(void)
{
    return keys;
}

static int  dummy_ping(AGENT_REQUEST *request, AGENT_RESULT *result)
{
    SET_UI64_RESULT(result, 1);

    return SYSINFO_RET_OK;
}

static int  dummy_echo(AGENT_REQUEST *request, AGENT_RESULT *result)
{
    char    *param;

    if (1 != request->nparam)
    {
        /* set optional error message */
        SET_MSG_RESULT(result, strdup("Invalid number of parameters."));
        return SYSINFO_RET_FAIL;
    }

    param = get_rparam(request, 0);

    SET_STR_RESULT(result, strdup(param));

    return SYSINFO_RET_OK;
}

/******************************************************************************
 *                                                                            *
 * Function: dummy_random                                                     *
 *                                                                            *
 * Purpose: a main entry point for processing of an item                      *
 *                                                                            *
 * Parameters: request - structure that contains item key and parameters      *
 *              request->key - item key without parameters                    *
 *              request->nparam - number of parameters                        *
 *              request->params[N-1] - pointers to item key parameters        *
 *              request->types[N-1] - item key parameters types:              *
```

```
 *                  REQUEST_PARAMETER_TYPE_UNDEFINED (key parameter is empty) *
 *                  REQUEST_PARAMETER_TYPE_ARRAY (array)                      *
 *                  REQUEST_PARAMETER_TYPE_STRING (quoted or unquoted string) *
 *                                                                            *
 *             result - structure that will contain result                    *
 *                                                                            *
 * Return value: SYSINFO_RET_FAIL - function failed, item will be marked      *
 *                                  as not supported by zabbix                *
 *               SYSINFO_RET_OK - success                                     *
 *                                                                            *
 * Comment: get_rparam(request, N-1) can be used to get a pointer to the Nth  *
 *          parameter starting from 0 (first parameter). Make sure it exists  *
 *          by checking value of request->nparam.                             *
 *          In the same manner get_rparam_type(request, N-1) can be used to   *
 *          get a parameter type.                                             *
 *                                                                            *
 ******************************************************************************/
static int  dummy_random(AGENT_REQUEST *request, AGENT_RESULT *result)
{
    char    *param1, *param2;
    int from, to;

    if (2 != request->nparam)
    {
        /* set optional error message */
        SET_MSG_RESULT(result, strdup("Invalid number of parameters."));
        return SYSINFO_RET_FAIL;
    }

    param1 = get_rparam(request, 0);
    param2 = get_rparam(request, 1);

    /* there is no strict validation of parameters and types for simplicity sake */
    from = atoi(param1);
    to = atoi(param2);

    if (from > to)
    {
        SET_MSG_RESULT(result, strdup("Invalid range specified."));
        return SYSINFO_RET_FAIL;
    }

    SET_UI64_RESULT(result, from + rand() % (to - from + 1));

    return SYSINFO_RET_OK;
}

/******************************************************************************
 *                                                                            *
 * Function: zbx_module_init                                                  *
 *                                                                            *
 * Purpose: the function is called on agent startup                           *
 *          It should be used to call any initialization routines             *
 *                                                                            *
 * Return value: ZBX_MODULE_OK - success                                      *
 *               ZBX_MODULE_FAIL - module initialization failed               *
 *                                                                            *
 * Comment: the module won't be loaded in case of ZBX_MODULE_FAIL             *
 *                                                                            *
 ******************************************************************************/
int zbx_module_init(void)
{
```

```
    /* initialization for dummy.random */
    srand(time(NULL));

    return ZBX_MODULE_OK;
}

/******************************************************************************
 *                                                                            *
 * Function: zbx_module_uninit                                                *
 *                                                                            *
 * Purpose: the function is called on agent shutdown                          *
 *          It should be used to cleanup used resources if there are any      *
 *                                                                            *
 * Return value: ZBX_MODULE_OK - success                                      *
 *               ZBX_MODULE_FAIL - function failed                            *
 *                                                                            *
 ******************************************************************************/
int zbx_module_uninit(void)
{
    return ZBX_MODULE_OK;
}

/******************************************************************************
 *                                                                            *
 * Functions: dummy_history_float_cb                                          *
 *            dummy_history_integer_cb                                        *
 *            dummy_history_string_cb                                         *
 *            dummy_history_text_cb                                           *
 *            dummy_history_log_cb                                            *
 *                                                                            *
 * Purpose: callback functions for storing historical data of types float,    *
 *          integer, string, text and log respectively in external storage    *
 *                                                                            *
 * Parameters: history     - array of historical data                         *
 *             history_num - number of elements in history array              *
 *                                                                            *
 ******************************************************************************/
static void dummy_history_float_cb(const ZBX_HISTORY_FLOAT *history, int history_num)
{
    int i;

    for (i = 0; i < history_num; i++)
    {
        /* do something with history[i].itemid, history[i].clock, history[i].ns, history[i].value, ... */
    }
}

static void dummy_history_integer_cb(const ZBX_HISTORY_INTEGER *history, int history_num)
{
    int i;

    for (i = 0; i < history_num; i++)
    {
        /* do something with history[i].itemid, history[i].clock, history[i].ns, history[i].value, ... */
    }
}

static void dummy_history_string_cb(const ZBX_HISTORY_STRING *history, int history_num)
{
    int i;

    for (i = 0; i < history_num; i++)
```

```
    {
        /* do something with history[i].itemid, history[i].clock, history[i].ns, history[i].value, ... */
    }
}

static void dummy_history_text_cb(const ZBX_HISTORY_TEXT *history, int history_num)
{
    int i;

    for (i = 0; i < history_num; i++)
    {
        /* do something with history[i].itemid, history[i].clock, history[i].ns, history[i].value, ... */
    }
}

static void dummy_history_log_cb(const ZBX_HISTORY_LOG *history, int history_num)
{
    int i;

    for (i = 0; i < history_num; i++)
    {
        /* do something with history[i].itemid, history[i].clock, history[i].ns, history[i].value, ... */
    }
}

/******************************************************************************
 *                                                                            *
 * Function: zbx_module_history_write_cbs                                     *
 *                                                                            *
 * Purpose: returns a set of module functions Zabbix will call to export      *
 *          different types of historical data                                *
 *                                                                            *
 * Return value: structure with callback function pointers (can be NULL if    *
 *               module is not interested in data of certain types)           *
 *                                                                            *
 ******************************************************************************/
ZBX_HISTORY_WRITE_CBS   zbx_module_history_write_cbs(void)
{
    static ZBX_HISTORY_WRITE_CBS    dummy_callbacks =
    {
        dummy_history_float_cb,
        dummy_history_integer_cb,
        dummy_history_string_cb,
        dummy_history_text_cb,
        dummy_history_log_cb,
    };

    return dummy_callbacks;
}
```

Le module exporte trois nouveaux éléments :

- `dummy.ping` - renvoie toujours '1'
- `dummy.echo[param1]` - renvoie le premier paramètre tel quel, par exemple `dummy.echo[ABC]` renverra ABC
- `dummy.random[param1, param2]` - renvoie un nombre aléatoire dans l'intervalle param1-param2, par exemple `dummy.random[1,1000000]`

Limitations

La prise en charge des modules chargeables est implémentée uniquement pour la plateforme Unix. Cela signifie qu'elle ne fonctionne pas pour les agents Windows.

Dans certains cas, un module peut avoir besoin de lire des paramètres de configuration liés au module depuis `zabbix_agentd.conf`. Ce n'est pas pris en charge actuellement. Si vous avez besoin que votre module utilise certains paramètres de configuration, vous devriez probablement implémenter l'analyse d'un fichier de configuration spécifique au module.

Modules de l'interface

Aperçu

Il est possible d'améliorer les fonctionnalités de l'interface Zabbix en ajoutant des modules tiers ou en développant vos propres modules, sans avoir à modifier le code source de Zabbix.

Notez que le code du module s'exécutera avec les mêmes privilèges que le code source de Zabbix. Cela signifie :

- Les modules tiers peuvent être dangereux. Vous devez faire confiance aux modules que vous installez.
- Des erreurs dans le code d'un module tiers peuvent faire planter l'interface. Si cela se produit, supprimez simplement le code du module de l'interface. Dès que vous rechargez l'interface Zabbix, vous verrez une note indiquant que certains modules sont absents. Allez dans Administration des modules (dans *Administration* > *Général* > *Modules*) et cliquez à nouveau sur *Analyser le répertoire* pour supprimer de la base de données les modules inexistants.

Installation

Veuillez toujours lire le manuel d'installation du module concerné. Il est recommandé d'installer les nouveaux modules un par un afin de détecter facilement les échecs.

Juste avant d'installer un module :

- Assurez-vous d'avoir téléchargé le module depuis une source fiable. L'installation d'un code malveillant peut entraîner des conséquences, telles qu'une perte de données.
- Différentes versions du même module (même ID) peuvent être installées en parallèle, mais une seule version peut être activée à la fois.

Étapes pour installer un module :

- Décompressez votre module dans son propre dossier dans le répertoire `modules` de l'installation de l'interface Zabbix (par exemple, `zabbix/ui/modules`).
- Assurez-vous que le dossier de votre module contient au moins le fichier `manifest.json`.
- Accédez à Administration des modules et cliquez sur le bouton *Scanner le répertoire*.
- Le nouveau module apparaîtra dans la liste avec sa version, son auteur, sa description et son statut.
- Activez le module en cliquant sur son statut.

Dépannage :

Problem	Solution
Module did not appear in the list	Si votre module n'apparaît pas dans la liste, assurez-vous que `manifest.json` existe dans `modules/your-module/`. Si ce n'est pas le cas, vous avez peut-être décompressé le module dans le mauvais répertoire. Si c'est le cas, le module n'est peut-être pas compatible avec votre version de Zabbix. Vérifiez également que l'utilisateur du serveur web dispose d'au moins des droits de lecture et de recherche (`r-x`) sur le dossier du module et tous ses sous-répertoires, ainsi que des droits de lecture (`r--`) sur tous les fichiers qu'ils contiennent.
Frontend crashed	Le code du module n'est pas compatible avec la version actuelle de Zabbix ou avec la configuration du serveur. Veuillez supprimer les fichiers du module et recharger l'interface. Vous verrez un message indiquant que certains modules sont absents. Accédez à Administration des modules et cliquez à nouveau sur *Scanner le répertoire* pour supprimer de la base de données les modules inexistants.
Error message about identical namespace, ID or actions appears	Le nouveau module a tenté d'enregistrer un espace de noms, un ID ou des actions déjà enregistrés par d'autres modules activés. Désactivez le module en conflit (mentionné dans le message d'erreur) avant d'activer le nouveau.
Technical error messages appear	Signalez les erreurs au développeur du module.

Développement de modules

Pour plus d'informations sur le développement de modules personnalisés, consultez le centre de développement.

Annexes

Aperçu Cette section contient des documents de référence, des paramètres de configuration, des spécifications techniques et des guides de procédure qui complètent le manuel principal.

Référence essentielle

- Installation et configuration - création de la base de données, installation et configuration du serveur et de l'agent.
- Configuration des processus - paramètres de configuration pour le serveur Zabbix, le proxy, l'agent, et plus encore.
- Éléments - référence des clés d'élément et des types de données pris en charge.
- Macros - syntaxe et utilisation des macros intégrées et utilisateur.

Installation et configuration

Cette section couvre la création de la base de données, la configuration de la sécurité, les procédures de configuration spécifiques à la plateforme et la configuration des fonctionnalités.

Veuillez utiliser la barre latérale pour trouver la procédure adaptée à votre environnement.

Création de la base de données

Vue d'ensemble

Une base de données Zabbix doit être créée lors de l'installation du serveur Zabbix ou du proxy.

Cette section fournit des instructions pour créer une base de données Zabbix. Un ensemble distinct d'instructions est disponible pour chaque base de données prise en charge.

> **Note:**
> Pour améliorer la sécurité de la base de données en créant des rôles/utilisateurs de base de données avec des privilèges minimaux, consultez les bonnes pratiques de création de base de données pour chaque base de données prise en charge :

> - MySQL/MariaDB
> - PostgreSQL/TimescaleDB
>
> Pour configurer des connexions TLS sécurisées, consultez Connexion sécurisée à la base de données.

UTF-8 est le seul encodage pris en charge par Zabbix. Il est connu pour fonctionner sans aucune faille de sécurité. Les utilisateurs doivent savoir qu'il existe des problèmes de sécurité connus lors de l'utilisation de certains autres encodages. Pour passer à UTF-8, consultez Réparation du jeu de caractères et du classement de la base de données Zabbix. Voir aussi Limites du filtrage avec les classements utf8mb4.

> **Note:**
> Si vous installez depuis le dépôt Git Zabbix, vous devez exécuter la commande suivante avant de passer aux étapes suivantes :

 `make dbschema`

MySQL/MariaDB

Le jeu de caractères utf8mb4 avec la collation utf8mb4_bin est pris en charge pour que le serveur/proxy Zabbix fonctionne correctement avec les bases de données MySQL/MariaDB.
 Pour éviter des problèmes ultérieurs, il est fortement recommandé d'utiliser le jeu de caractères utf8mb4 avec la collation utf8mb4_bin pour les nouvelles installations comme pour les installations existantes. Pour convertir les bases de données utilisant le jeu de caractères utf8mb3 et la collation utf8mb3_bin vers le jeu de caractères utf8mb4 et la collation utf8mb4_bin, suivez les instructions de la page Réparer le jeu de caractères et la collation de la base de données Zabbix.

Les déclencheurs déterministes doivent être créés lors de l'importation du schéma. Avec MySQL et MariaDB, cela nécessite de définir `GLOBAL log_bin_trust_function_creators = 1` si la journalisation binaire est activée, qu'aucun privilège de superutilisateur n'est disponible et que `log_bin_trust_function_creators = 1` n'est pas défini dans le fichier de configuration MySQL.

Si vous installez Zabbix à partir de paquets ou de conteneurs, consultez les instructions correspondantes.

Si vous installez Zabbix à partir des sources :

- Créez et configurez une base de données et un utilisateur. Pour des raisons de sécurité, utilisez un mot de passe fort et unique pour l'utilisateur de la base de données Zabbix.

```
mysql -uroot -p<password>

mysql> create database zabbix character set utf8mb4 collate utf8mb4_bin;
mysql> create user 'zabbix'@'localhost' identified by '<password>';
mysql> grant all privileges on zabbix.* to 'zabbix'@'localhost';
mysql> SET GLOBAL log_bin_trust_function_creators = 1;
mysql> quit;
```

- Importez les données dans la base de données et définissez le jeu de caractères utf8mb4 comme valeur par défaut (en supposant que vous vous trouvez dans le répertoire racine des sources Zabbix). Pour une base de données de proxy Zabbix, seul `schema.sql` doit être importé (ni images.sql ni data.sql).

```
cd database/mysql
mysql -uzabbix -p<password> zabbix < schema.sql
#### stop here if you are creating database for Zabbix proxy
mysql -uzabbix -p<password> zabbix < images.sql
mysql -uzabbix -p<password> --default-character-set=utf8mb4 zabbix < data.sql
```

`log_bin_trust_function_creators` peut être désactivé une fois le schéma importé avec succès :

```
mysql -uroot -p<password>

mysql> SET GLOBAL log_bin_trust_function_creators = 0;
mysql> quit;
```

PostgreSQL

Vous devez disposer d'un utilisateur de base de données ayant les autorisations nécessaires pour créer des objets de base de données.

Si vous installez Zabbix à partir de packages ou de conteneurs, consultez les instructions correspondantes.

Si vous installez Zabbix à partir des sources :

- Créez un utilisateur de base de données.

La commande shell suivante créera l'utilisateur `zabbix`. Spécifiez un mot de passe lorsque vous y êtes invité, puis saisissez-le à nouveau (notez que l'on peut d'abord vous demander le mot de passe sudo) :

```
sudo -u postgres createuser --pwprompt zabbix
```

- Créez une base de données.

La commande shell suivante créera la base de données `zabbix` (dernier paramètre) avec l'utilisateur créé précédemment comme propriétaire (`-O zabbix`).

```
sudo -u postgres createdb -O zabbix -E Unicode -T template0 zabbix
```

- Importez le schéma initial et les données (en supposant que vous vous trouvez dans le répertoire racine des sources de Zabbix). Pour une base de données de proxy Zabbix, seul `schema.sql` doit être importé (ni `images.sql` ni `data.sql`).

```
cd database/postgresql
cat schema.sql | sudo -u zabbix psql zabbix
#### stop here if you are creating database for Zabbix proxy
cat images.sql | sudo -u zabbix psql zabbix
cat data.sql | sudo -u zabbix psql zabbix
```

Attention:
Les commandes ci-dessus sont fournies à titre d'exemple et fonctionneront dans la plupart des installations GNU/Linux. Vous pouvez utiliser d'autres commandes selon la configuration de votre système/base de données, par exemple :

 `psql -U <username>`

 Si vous rencontrez des difficultés lors de la configuration de la base de données, veuillez consulter votre administrateur de base de données.

TimescaleDB

Les instructions pour créer et configurer TimescaleDB sont fournies dans une section distincte.

Elasticsearch et ClickHouse

Zabbix peut également stocker les données d'historique dans Elasticsearch ou ClickHouse. Pour les instructions de configuration, consultez les pages liées.

SQLite

L'utilisation de SQLite est prise en charge uniquement pour le **proxy Zabbix** !

La base de données sera automatiquement créée si elle n'existe pas.

Retournez à la section d'installation.

Réparation du jeu de caractères et de la collation de la base de données Zabbix

MySQL/MariaDB

Historiquement, MySQL et ses dérivés utilisaient 'utf8' comme alias de utf8mb3 - l'implémentation 3 octets de MySQL du standard UTF8, qui en utilise 4. À partir de MySQL 8.0.28 et MariaDB 10.6.1, le jeu de caractères 'utf8mb3' est obsolète et, à terme, sa prise en charge sera supprimée, tandis que 'utf8' deviendra une référence à 'utf8mb4'. Depuis Zabbix 6.0, 'utf8mb4' est pris en charge. Pour éviter de futurs problèmes, il est fortement recommandé d'utiliser 'utf8mb4'. Un autre avantage du passage à 'utf8mb4' est la prise en charge des caractères Unicode supplémentaires.

> **Warning:**
> Comme les versions antérieures à Zabbix 6.0 ne connaissent pas utf8mb4, assurez-vous de d'abord mettre à niveau le serveur Zabbix et le schéma de base de données vers la version 6.0.x ou ultérieure avant d'exécuter la conversion utf8mb4.

1. Vérifiez le jeu de caractères et la collation de la base de données.

Par exemple :

```
mysql> SELECT @@character_set_database, @@collation_database;
+--------------------------+----------------------+
| @@character_set_database | @@collation_database |
+--------------------------+----------------------+
| latin2                   | latin2 _general_ci   |
+--------------------------+----------------------+
```

Ou :

```
mysql> SELECT @@character_set_database, @@collation_database;
+--------------------------+----------------------+
| @@character_set_database | @@collation_database |
+--------------------------+----------------------+
| utf8                     | utf8_bin             |
+--------------------------+----------------------+
```

Comme on le voit, le jeu de caractères ici n'est pas 'utf8mb4' et la collation n'est pas 'utf8mb4_bin', il faut donc les corriger.

2. Arrêtez Zabbix.

3. Créez une copie de sauvegarde de la base de données !

4. Corrigez le jeu de caractères et la collation au niveau de la base de données :

```
alter database <your DB name> character set utf8mb4 collate utf8mb4_bin;
```

Valeurs corrigées :

```
mysql> SELECT @@character_set_database, @@collation_database;
+--------------------------+----------------------+
| @@character_set_database | @@collation_database |
+--------------------------+----------------------+
| utf8mb4                  | utf8mb4_bin          |
+--------------------------+----------------------+
```

5. Chargez le script pour corriger le jeu de caractères et la collation au niveau des tables et des colonnes :

```
mysql <your DB name> < utf8mb4_convert.sql
```

6. Exécutez le script :

```
                SET @ZABBIX_DATABASE = '<your DB name>';
Si MariaDB →    set innodb_strict_mode = OFF;
                CALL zbx_convert_utf8();
Si MariaDB →    set innodb_strict_mode = ON;
                drop procedure zbx_convert_utf8;
```

Veuillez noter que 'utf8mb4' est censé consommer légèrement plus d'espace disque.

7. Si aucune erreur ne s'est produite, vous pouvez créer une copie de sauvegarde de la base de données avec la base de données corrigée.

8. Démarrez Zabbix.

Mise à niveau de la base de données vers des clés primaires

Vue d'ensemble

Cette section fournit des instructions pour mettre manuellement à niveau les tables des installations existantes vers des clés primaires.

La mise à niveau vers des clés primaires optimise la manière dont les données sont indexées et consultées, ce qui peut accélérer les requêtes et économiser de l'espace. Elle améliore également la gestion des données et la synchronisation dans les configurations en cluster, ce qui facilite la montée en charge et aide à garantir la fiabilité du système, même si certains serveurs tombent en panne.

> **Attention:**
> Les instructions fournies sur cette page sont conçues pour des utilisateurs avancés et peuvent devoir être adaptées à votre configuration spécifique. La mise à niveau vers des clés primaires peut prendre du temps et consommer beaucoup de ressources. Assurez-vous qu'un espace disque libre suffisant est disponible ; selon la taille de votre base de données et les données stockées, le processus peut nécessiter jusqu'à 2,5 fois l'espace actuellement utilisé par les tables d'historique.

Les clés primaires sont utilisées pour toutes les tables dans les nouvelles installations depuis Zabbix 6.0.

Il n'existe pas de mise à niveau automatique de la base de données vers des clés primaires ; toutefois, les installations existantes peuvent être mises à niveau manuellement **après** la mise à niveau du serveur Zabbix vers la version 6.0 ou ultérieure.

> **Attention:**
> Depuis Zabbix 7.0, la mise à niveau des tables vers des clés primaires met également à niveau les tables pour utiliser des types de données en double précision.

 Si vous utilisez Zabbix 7.0 (ou une version ultérieure), les tables utilisent déjà la double précision. Cependant, les instructions de cette page peuvent toujours être suivies pour mettre à niveau les tables vers des clés primaires sans affecter les tables qui utilisent déjà la double précision.

 Si vous utilisez Zabbix 6.4 (ou une version antérieure), envisagez de mettre d'abord les tables à niveau vers la double précision. Pour plus d'informations, consultez Mise à niveau vers des valeurs numériques à plage étendue dans la documentation Zabbix 7.0.

Des instructions sont disponibles pour :

- MariaDB/MySQL
- PostgreSQL
- TimescaleDB

Remarques importantes

Pour effectuer la mise à niveau de la base de données :

1. Arrêtez le serveur Zabbix.

Il est fortement recommandé d'arrêter le serveur Zabbix pendant la durée de la mise à niveau. Cependant, si cela est absolument nécessaire, vous pouvez effectuer la mise à niveau pendant que le serveur est en cours d'exécution (uniquement pour MySQL, MariaDB et PostgreSQL sans TimescaleDB).

2. Sauvegardez votre base de données.
3. Installez le dernier paquet zabbix-sql-scripts compatible avec votre version de Zabbix (par exemple, pour RHEL : `dnf install zabbix-sql-scripts`).
4. Exécutez les scripts pour votre base de données.
5. Démarrez le serveur Zabbix.

Warning:
Exécutez les scripts uniquement pour la base de données du serveur. Le proxy ne bénéficiera pas de cette mise à niveau.

Si la base de données utilise des partitions, contactez l'administrateur de la base de données ou le support Zabbix pour obtenir de l'aide.

Les fichiers CSV peuvent être supprimés après une mise à niveau réussie vers des clés primaires.

En option, le frontend Zabbix peut être basculé en mode maintenance.

MariaDB/MySQL

L'exportation et l'importation doivent être effectuées dans tmux/screen afin de garantir que la session ne soit pas interrompue.

Voir aussi : Notes importantes

MySQL 8.4+ avec mysqlsh

Cette méthode peut être utilisée avec un serveur Zabbix en cours d'exécution, mais il est recommandé d'arrêter le serveur pendant la mise à niveau. MySQL Shell (*mysqlsh*) doit être installé et capable de se connecter à la base de données.

- Connectez-vous à la console MySQL en tant que root (recommandé) ou en tant que tout utilisateur disposant des privilèges FILE.
- Démarrez MySQL avec la variable local_infile activée.
- Renommez les anciennes tables et créez les nouvelles tables en exécutant `history_upgrade_prepare.sql`.

```
mysql -uzabbix -p<password> zabbix < /usr/share/zabbix/sql-scripts/mysql/option-patches/history_upgrade_pr
```

- Exportez et importez les données.

Connectez-vous via mysqlsh. Si vous utilisez une connexion par socket, il peut être nécessaire de spécifier le chemin.

```
sudo mysqlsh -uroot -S /run/mysqld/mysqld.sock --no-password -Dzabbix
```

Passez en mode JavaScript à l'aide de :

```
\js
```

Exécutez ensuite le code ci-dessous (CSVPATH peut être modifié selon les besoins) :

```
CSVPATH="/var/lib/mysql-files";

util.exportTable("history_old", CSVPATH + "/history.csv", { dialect: "csv" });
util.importTable(CSVPATH + "/history.csv", {"dialect": "csv", "table": "history" });

util.exportTable("history_uint_old", CSVPATH + "/history_uint.csv", { dialect: "csv" });
util.importTable(CSVPATH + "/history_uint.csv", {"dialect": "csv", "table": "history_uint" });

util.exportTable("history_str_old", CSVPATH + "/history_str.csv", { dialect: "csv" });
util.importTable(CSVPATH + "/history_str.csv", {"dialect": "csv", "table": "history_str" });

util.exportTable("history_log_old", CSVPATH + "/history_log.csv", { dialect: "csv" });
util.importTable(CSVPATH + "/history_log.csv", {"dialect": "csv", "table": "history_log" });

util.exportTable("history_text_old", CSVPATH + "/history_text.csv", { dialect: "csv" });
util.importTable(CSVPATH + "/history_text.csv", {"dialect": "csv", "table": "history_text" });
```

Si vous recevez le message "JavaScript is not supported", votre installation de MySQL Shell ne prend pas en charge JS. Dans ce cas, installez le package officiel MySQL Shell d'Oracle (ou compilez-le à partir des sources) afin que le mode JavaScript soit activé.

- Suivez les instructions post-migration pour supprimer les anciennes tables.

MariaDB/MySQL 8.4+ sans mysqlsh

Cette méthode de mise à niveau prend plus de temps et ne doit être utilisée que si une mise à niveau avec *mysqlsh* n'est pas possible.

Mise à niveau des tables

- Connectez-vous à la console MySQL en tant que root (recommandé) ou en tant qu'utilisateur disposant des privilèges FILE.

- Si vous effectuez la migration avec un serveur Zabbix en cours d'exécution, démarrez MySQL avec la variable local_infile activée.
- Renommez les anciennes tables et créez les nouvelles en exécutant `history_upgrade_prepare.sql` :

```
mysql -uzabbix -p<password> zabbix < /usr/share/zabbix/sql-scripts/mysql/option-patches/history_upgrade_pr
```

Migration avec serveur arrêté

`max_execution_time` (dans MySQL) ou `max_statement_time` (dans MariaDB) doit être désactivé avant la migration des données afin d'éviter un dépassement de délai pendant la migration.

Pour MySQL :

```
SET @@max_execution_time=0;
```

Pour MariaDB :

```
SET @@max_statement_time=0;
```

```
INSERT IGNORE INTO history SELECT * FROM history_old;
INSERT IGNORE INTO history_uint SELECT * FROM history_uint_old;
INSERT IGNORE INTO history_str SELECT * FROM history_str_old;
INSERT IGNORE INTO history_log SELECT * FROM history_log_old;
INSERT IGNORE INTO history_text SELECT * FROM history_text_old;
```

Suivez les instructions post-migration pour supprimer les anciennes tables.

Migration avec serveur en cours d'exécution

Vérifiez pour quels chemins l'import/export est activé :

```
mysql> SELECT @@secure_file_priv;
+-----------------------+
| @@secure_file_priv    |
+-----------------------+
| /var/lib/mysql-files/ |
+-----------------------+
```

Si la valeur de *secure_file_priv* est un chemin vers un répertoire, l'export/import sera effectué pour les fichiers de ce répertoire. Dans ce cas, modifiez en conséquence les chemins vers les fichiers dans les requêtes ou définissez la valeur *secure_file_priv* sur une chaîne vide pendant la durée de la mise à niveau.

Si la valeur de *secure_file_priv* est vide, l'export/import peut être effectué depuis n'importe quel emplacement.

Si la valeur de *secure_file_priv* est NULL, définissez-la sur le chemin qui contient les données exportées des tables ('/var/lib/mysql-files/' dans l'exemple ci-dessus).

Pour plus d'informations, consultez la documentation MySQL ou la documentation MariaDB.

`max_execution_time` (dans MySQL) ou `max_statement_time` (dans MariaDB) doit être désactivé avant l'export des données afin d'éviter un dépassement de délai pendant l'export.

Pour MySQL :

```
SET @@max_execution_time=0;
```

Pour MariaDB :

```
SET @@max_statement_time=0;
```

```
SELECT * INTO OUTFILE '/var/lib/mysql-files/history.csv' FIELDS TERMINATED BY ',' ESCAPED BY '"' LINES TER
LOAD DATA INFILE '/var/lib/mysql-files/history.csv' IGNORE INTO TABLE history FIELDS TERMINATED BY ',' ESC

SELECT * INTO OUTFILE '/var/lib/mysql-files/history_uint.csv' FIELDS TERMINATED BY ',' ESCAPED BY '"' LINE
LOAD DATA INFILE '/var/lib/mysql-files/history_uint.csv' IGNORE INTO TABLE history_uint FIELDS TERMINATED

SELECT * INTO OUTFILE '/var/lib/mysql-files/history_str.csv' FIELDS TERMINATED BY ',' ESCAPED BY '"' LINES
LOAD DATA INFILE '/var/lib/mysql-files/history_str.csv' IGNORE INTO TABLE history_str FIELDS TERMINATED BY

SELECT * INTO OUTFILE '/var/lib/mysql-files/history_log.csv' FIELDS TERMINATED BY ',' ESCAPED BY '"' LINES
LOAD DATA INFILE '/var/lib/mysql-files/history_log.csv' IGNORE INTO TABLE history_log FIELDS TERMINATED BY
```

```
SELECT * INTO OUTFILE '/var/lib/mysql-files/history_text.csv' FIELDS TERMINATED BY ',' ESCAPED BY '"' LINE
LOAD DATA INFILE '/var/lib/mysql-files/history_text.csv' IGNORE INTO TABLE history_text FIELDS TERMINATED
```

Suivez les instructions post-migration pour supprimer les anciennes tables.

PostgreSQL

L'exportation et l'importation doivent être effectuées dans tmux/screen afin de garantir que la session ne soit pas interrompue. Pour les installations avec TimescaleDB, ignorez cette section et passez à PostgreSQL + TimescaleDB.

Voir aussi : Notes importantes

Mise à niveau des tables

- Renommez les tables à l'aide de `history_upgrade_prepare.sql` :

```
sudo -u zabbix psql zabbix < /usr/share/zabbix/sql-scripts/postgresql/option-patches/history_upgrade_prepa
```

Migration avec serveur arrêté

- Exportez l'historique actuel, importez-le dans la table temporaire, puis insérez les données dans les nouvelles tables tout en ignorant les doublons :

```
INSERT INTO history SELECT * FROM history_old ON CONFLICT (itemid,clock,ns) DO NOTHING;

INSERT INTO history_uint SELECT * FROM history_uint_old ON CONFLICT (itemid,clock,ns) DO NOTHING;

INSERT INTO history_str SELECT * FROM history_str_old ON CONFLICT (itemid,clock,ns) DO NOTHING;

INSERT INTO history_log SELECT * FROM history_log_old ON CONFLICT (itemid,clock,ns) DO NOTHING;

INSERT INTO history_text SELECT * FROM history_text_old ON CONFLICT (itemid,clock,ns) DO NOTHING;
```

Voir les conseils pour améliorer les performances de INSERT : PostgreSQL: Bulk Loading Huge Amounts of Data, Checkpoint Distance and Amount of WAL.

- Suivez les instructions post-migration pour supprimer les anciennes tables.

Migration avec serveur en cours d'exécution

- Exportez l'historique actuel, importez-le dans la table temporaire, puis insérez les données dans les nouvelles tables en ignorant les doublons :

```
\copy history_old TO '/tmp/history.csv' DELIMITER ',' CSV
CREATE TEMP TABLE temp_history (
    itemid                   bigint                                    NOT NULL,
    clock                    integer         DEFAULT '0'               NOT NULL,
    value                    DOUBLE PRECISION DEFAULT '0.0000'          NOT NULL,
    ns                       integer         DEFAULT '0'               NOT NULL
);
\copy temp_history FROM '/tmp/history.csv' DELIMITER ',' CSV
INSERT INTO history SELECT * FROM temp_history ON CONFLICT (itemid,clock,ns) DO NOTHING;

\copy history_uint_old TO '/tmp/history_uint.csv' DELIMITER ',' CSV
CREATE TEMP TABLE temp_history_uint (
    itemid                   bigint                                    NOT NULL,
    clock                    integer         DEFAULT '0'               NOT NULL,
    value                    numeric(20)     DEFAULT '0'               NOT NULL,
    ns                       integer         DEFAULT '0'               NOT NULL
);
\copy temp_history_uint FROM '/tmp/history_uint.csv' DELIMITER ',' CSV
INSERT INTO history_uint SELECT * FROM temp_history_uint ON CONFLICT (itemid,clock,ns) DO NOTHING;

\copy history_str_old TO '/tmp/history_str.csv' DELIMITER ',' CSV
CREATE TEMP TABLE temp_history_str (
    itemid                   bigint                                    NOT NULL,
    clock                    integer         DEFAULT '0'               NOT NULL,
    value                    varchar(255)    DEFAULT ''                NOT NULL,
    ns                       integer         DEFAULT '0'               NOT NULL
```

```
);
\copy temp_history_str FROM '/tmp/history_str.csv' DELIMITER ',' CSV
INSERT INTO history_str (itemid,clock,value,ns) SELECT * FROM temp_history_str ON CONFLICT (itemid,clock,n

\copy history_log_old TO '/tmp/history_log.csv' DELIMITER ',' CSV
CREATE TEMP TABLE temp_history_log (
    itemid                   bigint                                    NOT NULL,
    clock                    integer         DEFAULT '0'               NOT NULL,
    timestamp                integer         DEFAULT '0'               NOT NULL,
    source                   varchar(64)     DEFAULT ''                NOT NULL,
    severity                 integer         DEFAULT '0'               NOT NULL,
    value                    text            DEFAULT ''                NOT NULL,
    logeventid               integer         DEFAULT '0'               NOT NULL,
    ns                       integer         DEFAULT '0'               NOT NULL
);
\copy temp_history_log FROM '/tmp/history_log.csv' DELIMITER ',' CSV
INSERT INTO history_log SELECT * FROM temp_history_log ON CONFLICT (itemid,clock,ns) DO NOTHING;

\copy history_text_old TO '/tmp/history_text.csv' DELIMITER ',' CSV
CREATE TEMP TABLE temp_history_text (
    itemid                   bigint                                    NOT NULL,
    clock                    integer         DEFAULT '0'               NOT NULL,
    value                    text            DEFAULT ''                NOT NULL,
    ns                       integer         DEFAULT '0'               NOT NULL
);
\copy temp_history_text FROM '/tmp/history_text.csv' DELIMITER ',' CSV
INSERT INTO history_text SELECT * FROM temp_history_text ON CONFLICT (itemid,clock,ns) DO NOTHING;
```

- Suivez les instructions post-migration pour supprimer les anciennes tables.

PostgreSQL + TimescaleDB

L'exportation et l'importation doivent être effectués dans tmux/screen afin de garantir que la session ne soit pas interrompue. Le serveur Zabbix doit être arrêté pendant la mise à niveau.

Voir aussi: Notes importantes

- Renommez les tables à l'aide de `history_upgrade_prepare.sql`.
  - Si la compression est activée (dans l'installation par défaut), exécutez le script depuis `/usr/share/zabbix/sql-scripts/post`

    ```
    cat /usr/share/zabbix/sql-scripts/postgresql/timescaledb/option-patches/with-compression/history_
    ```

  - Si la compression est désactivée, exécutez le script depuis `/usr/share/zabbix/sql-scripts/postgresql/timescaledb/`

    ```
    cat /usr/share/zabbix/sql-scripts/postgresql/timescaledb/option-patches/without-compression/histo
    ```

- Exécutez les scripts de migration des hypertables TimescaleDB en fonction des paramètres de compression:
  - Si la compression est activée (dans l'installation par défaut), exécutez les scripts depuis `/usr/share/zabbix/sql-scripts/po`

    ```
    cat /usr/share/zabbix/sql-scripts/postgresql/timescaledb/option-patches/with-compression/history_
    cat /usr/share/zabbix/sql-scripts/postgresql/timescaledb/option-patches/with-compression/history_
    cat /usr/share/zabbix/sql-scripts/postgresql/timescaledb/option-patches/with-compression/history_
    cat /usr/share/zabbix/sql-scripts/postgresql/timescaledb/option-patches/with-compression/history_
    cat /usr/share/zabbix/sql-scripts/postgresql/timescaledb/option-patches/with-compression/history_
    cat /usr/share/zabbix/sql-scripts/postgresql/timescaledb/option-patches/with-compression/trends_u
    ```

  - Si la compression est désactivée, exécutez les scripts depuis `/usr/share/zabbix/sql-scripts/postgresql/timescaledb`

    ```
    cat /usr/share/zabbix/sql-scripts/postgresql/timescaledb/option-patches/without-compression/histo
    cat /usr/share/zabbix/sql-scripts/postgresql/timescaledb/option-patches/without-compression/histo
    cat /usr/share/zabbix/sql-scripts/postgresql/timescaledb/option-patches/without-compression/histo
    cat /usr/share/zabbix/sql-scripts/postgresql/timescaledb/option-patches/without-compression/histo
    cat /usr/share/zabbix/sql-scripts/postgresql/timescaledb/option-patches/without-compression/histo
    cat /usr/share/zabbix/sql-scripts/postgresql/timescaledb/option-patches/without-compression/trend
    ```

Voir aussi: Conseils pour améliorer les performances d'INSERT.

- Suivez les instructions post-migration pour supprimer les anciennes tables.

Post-migration

Pour toutes les bases de données, une fois la migration terminée, procédez comme suit :

- Vérifiez que tout fonctionne comme prévu.
- Supprimez les anciennes tables :

```
DROP TABLE history_old;
DROP TABLE history_uint_old;
DROP TABLE history_str_old;
DROP TABLE history_log_old;
DROP TABLE history_text_old;
```

- Pour TimescaleDB, supprimez également l'ancienne table suivante :

```
DROP TABLE trends_old;
```

Voir aussi

- Préparation de la table auditlog pour le partitionnement

Préparation de la table auditlog pour le partitionnement

Présentation

Certaines bases de données (par exemple, MySQL) nécessitent que la colonne de partitionnement fasse partie de la contrainte unique de la table. Par conséquent, pour partitionner la table `auditlog` par heure, la clé primaire doit être modifiée de `auditid` à une clé composite `auditid` + `clock`.

Cette section fournit des instructions pour modifier la clé primaire de la table `auditlog`.

> **Attention:**
> Les instructions fournies sur cette page sont destinées aux utilisateurs avancés. Notez que ces instructions peuvent devoir être ajustées en fonction de votre configuration spécifique. La modification de la clé primaire peut également être incompatible avec les futurs correctifs de mise à niveau, il peut donc être nécessaire de gérer manuellement les futures mises à niveau.

 La modification de la clé primaire peut être une opération gourmande en ressources qui prend beaucoup de temps en fonction de la taille de la table `auditlog`. Il est recommandé d'arrêter le serveur Zabbix et de basculer l'interface Zabbix en mode maintenance pendant la durée de la modification. Cependant, si cela est absolument nécessaire, il existe un moyen de modifier la clé primaire sans interruption de service (voir ci-dessous).

Le partitionnement de la table `auditlog` peut améliorer, par exemple, la gestion interne dans les grandes configurations. Bien que Zabbix gestion interne ne puisse actuellement pas tirer parti des tables partitionnées (à l'exception de TimescaleDB), vous pouvez désactiver la gestion interne Zabbix et supprimer des partitions à l'aide de scripts.

Depuis Zabbix 7.0, la table `auditlog` de TimescaleDB a été convertie en hypertable, ce qui permet au gestionnaire de supprimer des données par blocs. Pour mettre à niveau la table `auditlog` existante vers une hypertable, voir Mise à niveau du schéma TimescaleDB.

MySQL

Notes importantes sur la reconstruction des index

MySQL reconstruit automatiquement les index de la clé primaire lors de l'opération `ALTER TABLE`. Cependant, il est fortement recommandé de reconstruire également manuellement les index avec l'instruction `OPTIMIZE TABLE` afin de garantir des performances optimales de la base de données.

La reconstruction des index peut nécessiter temporairement autant d'espace disque supplémentaire que celui utilisé par la table elle-même. Pour obtenir la taille actuelle des données et des index, vous pouvez exécuter les instructions suivantes :

```
ANALYZE TABLE auditlog;
SHOW TABLE STATUS LIKE 'auditlog';
```

Si l'espace disque disponible est une préoccupation, suivez les instructions de Modification de la clé primaire sans interruption. D'autres options sont également disponibles :

- Augmenter le paramètre MySQL `sort_buffer_size` peut aider à réduire l'utilisation de l'espace disque lors de la reconstruction manuelle des index. Cependant, la modification de cette variable peut avoir un impact sur l'utilisation globale de la mémoire de la base de données.
- Envisagez de libérer de l'espace en supprimant des données potentiellement inutiles.
- Envisagez de réduire le paramètre *Période de stockage des données* housekeeper avant d'exécuter le housekeeper.

Modification de la clé primaire avec interruption

1. Supprimez la clé primaire actuelle de la table `auditlog` et ajoutez la nouvelle clé primaire.

```
ALTER TABLE auditlog DROP PRIMARY KEY, ADD PRIMARY KEY (auditid, clock);
```

2. Reconstruisez les index (facultatif, mais fortement recommandé, voir Notes importantes sur la reconstruction des index).

```
OPTIMIZE TABLE auditlog;
```

Modification de la clé primaire sans interruption

La méthode manuelle de modification de la clé primaire est décrite ici. Vous pouvez également utiliser la boîte à outils pt-online-schema-change de Percona. Cette boîte à outils effectue automatiquement les actions suivantes, tout en minimisant l'espace utilisé pour modifier la table `auditlog`.

1. Créez une nouvelle table avec la nouvelle clé primaire et créez les index.

```
CREATE TABLE `auditlog_new` (
  `auditid`            varchar(25)                         NOT NULL,
  `userid`             bigint unsigned                     NULL,
  `username`           varchar(100)    DEFAULT ''          NOT NULL,
  `clock`              integer         DEFAULT '0'         NOT NULL,
  `ip`                 varchar(39)     DEFAULT ''          NOT NULL,
  `action`             integer         DEFAULT '0'         NOT NULL,
  `resourcetype`       integer         DEFAULT '0'         NOT NULL,
  `resourceid`         bigint unsigned                     NULL,
  `resource_cuid`      varchar(25)                         NULL,
  `resourcename`       varchar(255)    DEFAULT ''          NOT NULL,
  `recordsetid`        varchar(25)                         NOT NULL,
  `details`            longtext                            NOT NULL,
  PRIMARY KEY (auditid,clock)
) ENGINE=InnoDB;
CREATE INDEX `auditlog_1` ON `auditlog_new` (`userid`,`clock`);
CREATE INDEX `auditlog_2` ON `auditlog_new` (`clock`);
CREATE INDEX `auditlog_3` ON `auditlog_new` (`resourcetype`,`resourceid`);
```

2. Permutez les tables.

```
RENAME TABLE auditlog TO auditlog_old, auditlog_new TO auditlog;
```

3. Copiez les données de l'ancienne table vers la nouvelle table.

```
INSERT INTO auditlog SELECT * FROM auditlog_old;
```

Cela peut être effectué par lots (plusieurs instructions `INSERT INTO` avec des clauses `WHERE clock` selon les besoins) afin d'éviter une consommation excessive de ressources.

4. Supprimez l'ancienne table.

```
DROP TABLE auditlog_old;
```

PostgreSQL

Notes importantes sur la reconstruction des index

PostgreSQL reconstruit automatiquement les index de la clé primaire pendant l'opération `ALTER TABLE`. Cependant, il est fortement recommandé de reconstruire également manuellement les index avec l'instruction `REINDEX TABLE CONCURRENTLY` afin de garantir des performances optimales de la base de données.

La reconstruction des index peut nécessiter temporairement jusqu'à trois fois l'espace disque actuellement utilisé par les index. Pour obtenir la taille actuelle des index, vous pouvez exécuter la requête suivante :

```
SELECT pg_size_pretty(pg_indexes_size('auditlog'));
```

Si l'espace disque disponible est une préoccupation, suivez les instructions de Modification de la clé primaire sans interruption. D'autres options sont également disponibles :

- Augmenter le paramètre PostgreSQL `maintenance_work_mem` peut aider à réduire l'utilisation de l'espace disque lors de la reconstruction manuelle des index. Cependant, la modification de cette variable peut avoir un impact sur l'utilisation globale de la mémoire de la base de données.

- Si vous disposez d'un autre disque ou d'un tablespace avec plus d'espace disponible, vous pouvez envisager de modifier l'emplacement de stockage temporaire pour la reconstruction de l'index. Vous pouvez définir le paramètre PostgreSQL `temp_tablespaces` pour spécifier un autre tablespace pour les objets temporaires.
- Envisagez de libérer de l'espace en supprimant des données potentiellement inutiles.
- Envisagez de réduire le paramètre *Période de stockage des données* housekeeper avant d'exécuter le housekeeper.

Modification de la clé primaire avec interruption

1. Supprimez la clé primaire actuelle de la table `auditlog` et ajoutez la nouvelle clé primaire.

```
ALTER TABLE auditlog DROP CONSTRAINT auditlog_pkey;
ALTER TABLE auditlog ADD PRIMARY KEY (auditid,clock);
```

2. Reconstruisez les index (facultatif, mais fortement recommandé, voir Notes importantes sur la reconstruction des index).

```
REINDEX TABLE CONCURRENTLY auditlog;
```

Modification de la clé primaire sans interruption

La méthode manuelle de modification de la clé primaire est décrite ici. Sinon, l'extension `pg_repack` peut être envisagée pour créer une nouvelle table, copier les données et permuter les tables.

1. Créez une nouvelle table avec la nouvelle clé primaire et créez les index.

```
CREATE TABLE auditlog_new (
  auditid               varchar(25)                             NOT NULL,
  userid                bigint                                  NULL,
  username              varchar(100)    DEFAULT ''              NOT NULL,
  clock                 integer         DEFAULT '0'             NOT NULL,
  ip                    varchar(39)     DEFAULT ''              NOT NULL,
  action                integer         DEFAULT '0'             NOT NULL,
  resourcetype          integer         DEFAULT '0'             NOT NULL,
  resourceid            bigint                                  NULL,
  resource_cuid         varchar(25)                             NULL,
  resourcename          varchar(255)    DEFAULT ''              NOT NULL,
  recordsetid           varchar(25)                             NOT NULL,
  details               text            DEFAULT ''              NOT NULL,
  PRIMARY KEY (auditid,clock)
);
CREATE INDEX auditlog_new_1 ON auditlog_new (userid,clock);
CREATE INDEX auditlog_new_2 ON auditlog_new (clock);
CREATE INDEX auditlog_new_3 ON auditlog_new (resourcetype,resourceid);
```

2. Permutez les tables.

```
ALTER TABLE auditlog RENAME TO auditlog_old;
ALTER TABLE auditlog_new RENAME TO auditlog;
```

3. Copiez les données de l'ancienne table vers la nouvelle table.

```
INSERT INTO auditlog SELECT * FROM auditlog_old;
```

Cela peut être fait par lots (plusieurs instructions `INSERT INTO` avec des clauses `WHERE clock` selon les besoins) afin d'éviter une consommation excessive de ressources.

4. Supprimez l'ancienne table.

```
DROP TABLE auditlog_old;
```

Voir aussi

- Mise à niveau de la base de données vers des clés primaires

Configuration de TimescaleDB

Aperçu

Zabbix prend en charge TimescaleDB, une solution de base de données basée sur PostgreSQL qui partitionne automatiquement les données en segments temporels afin d'offrir de meilleures performances à grande échelle.

Warning:
Actuellement, TimescaleDB n'est pas pris en charge par le proxy Zabbix.

Les instructions de cette page peuvent être utilisées dans les scénarios suivants :

- Création d'une base de données TimescaleDB ou migration de tables PostgreSQL existantes vers TimescaleDB (voir Configuration).
- Mise à niveau du schéma d'une base de données TimescaleDB existante lors de la mise à niveau de Zabbix (voir Mise à niveau du schéma TimescaleDB).

Configuration

Prérequis : extension TimescaleDB d'une version prise en charge installée sur le serveur de base de données. Pour les instructions d'installation, voir la documentation TimescaleDB.

Warning:
Avant d'installer TimescaleDB, installez une version PostgreSQL prise en charge depuis le dépôt PostgreSQL officiel.

Activez l'extension TimescaleDB pour la base de données spécifique en exécutant :

```
echo "CREATE EXTENSION IF NOT EXISTS timescaledb CASCADE;" | sudo -u postgres psql zabbix
```

L'exécution de cette commande nécessite des privilèges d'administrateur de base de données.

Note:
Si vous utilisez un schéma de base de données autre que 'public', vous devez ajouter une clause SCHEMA à la commande ci-dessus. Par ex. :
 `echo "CREATE EXTENSION IF NOT EXISTS timescaledb SCHEMA yourschema CASCADE;" | sudo -u postgres psql zabbix`

Exécutez ensuite le script `postgresql/timescaledb/schema.sql`. Pour les nouvelles installations, le script doit être exécuté après la création de la base de données PostgreSQL standard avec le schéma/les données initiaux (voir création de la base de données).

```
cat /usr/share/zabbix/sql-scripts/postgresql/timescaledb/schema.sql | sudo -u zabbix psql zabbix
```

Attention:
Veuillez ignorer les messages d'avertissement indiquant que les bonnes pratiques ne sont pas respectées lors de l'exécution du script `schema.sql` sur TimescaleDB version 2.9.0 et ultérieure. Malgré cet avertissement, la configuration sera effectuée avec succès.

La migration des données existantes d'historique, de tendances et de journal d'audit peut prendre beaucoup de temps. Le serveur Zabbix et l'interface doivent être arrêtés pendant toute la durée de la migration.

Le script `schema.sql` définit les paramètres de nettoyage suivants :

- Remplacer la période d'historique des éléments
- Remplacer la période de tendances des éléments

Pour utiliser le nettoyage partitionné pour l'historique et les tendances, ces deux options doivent être activées. Il est également possible d'activer le remplacement individuellement, soit uniquement pour l'historique, soit uniquement pour les tendances.

Pour PostgreSQL et TimescaleDB, le script `postgresql/timescaledb/schema.sql` définit deux paramètres supplémentaires :

- Activer la compression
- Compresser les enregistrements de plus de 7 jours

Pour que le housekeeper supprime correctement les données compressées, les options *Override item history period* et *Override item trend period* doivent toutes deux être activées. Si le remplacement est désactivé et que les tables contiennent des chunks compressés, le housekeeper ne supprimera pas les données de ces tables, et des avertissements concernant une configuration incorrecte seront affichés dans les sections *Housekeeping* et *System information*.

Tous ces paramètres peuvent être modifiés dans *Administration* > *Housekeeping* après l'installation.

Note:
Vous pouvez exécuter l'outil timescaledb-tune fourni par TimescaleDB afin d'optimiser les paramètres de configuration PostgreSQL dans votre `postgresql.conf`.

Mise à niveau du schéma TimescaleDB

Lors de la mise à niveau de Zabbix vers une version contenant de nouvelles hypertables TimescaleDB, le serveur Zabbix ne configure pas automatiquement ces hypertables (par exemple, lors d'une mise à niveau de Zabbix 6.4 vers 8.0, car les versions 7.0.0 et 7.0.2 ont introduit de nouvelles hypertables).

Pour configurer les nouvelles hypertables TimescaleDB, suivez ces étapes :

1. Démarrez le serveur Zabbix ; cela met à niveau la base de données existante.
2. Vérifiez dans le fichier journal du serveur que la mise à niveau de la base de données est terminée ; une fois terminée, arrêtez le serveur Zabbix. Notez que le serveur consigne un avertissement s'il tente d'activer la compression pour une table qui n'est pas une hypertable.
3. Exécutez le script `postgresql/timescaledb/schema.sql` ; cela configure les nouvelles hypertables TimescaleDB. Notez que, depuis Zabbix 7.0.0, l'emplacement et le nom du script ont été modifiés de `postgresql/timescaledb.sql` à `postgresql/timescaledb/schema.sql`.

Attention:
Veuillez ignorer les messages d'avertissement indiquant que les bonnes pratiques ne sont pas respectées lors de l'exécution du script `schema.sql` sur TimescaleDB version 2.9.0 et ultérieure. Malgré cet avertissement, la configuration sera effectuée avec succès.

Compression TimescaleDB

La compression native de TimescaleDB est prise en charge pour toutes les tables Zabbix qui sont des hypertables TimescaleDB. Lors de la mise à niveau ou de la migration vers TimescaleDB, la compression initiale des grandes tables peut prendre beaucoup de temps.

Notez que la compression est prise en charge sous la licence communautaire Timescale "timescale" et qu'elle n'est pas prise en charge sous la licence Apache 2.0 "apache". Si Zabbix détecte que la compression n'est pas prise en charge, un message d'avertissement est écrit dans le journal du serveur Zabbix et les utilisateurs ne peuvent pas activer la compression dans l'interface.

Note:
Il est recommandé aux utilisateurs de se familiariser avec la compression dans la documentation TimescaleDB avant d'utiliser la compression.

Notez que certaines limitations sont imposées par la compression, en particulier:

- Les modifications des chunks compressés (insertions, suppressions, mises à jour) ne sont pas autorisées
- Les changements de schéma pour les tables compressées ne sont pas autorisés.

Les paramètres de compression peuvent être modifiés dans le bloc *Compression de l'historique, des tendances et du journal d'audit* de la section *Administration > Housekeeping* de l'interface Zabbix.

Parameter	Default	Comments
Enable compression	Enabled	Cocher ou décocher la case n'active/désactive pas la compression immédiatement. Comme la compression est gérée par le Housekeeper, les modifications prendront effet dans un délai pouvant aller jusqu'à 2 fois `HousekeepingFrequency` heures (défini dans `zabbix_server.conf`) Après la désactivation de la compression, les nouveaux chunks qui entrent dans la période de compression ne seront pas compressés. Cependant, toutes les données déjà compressées resteront compressées. Pour décompresser des chunks précédemment compressés, suivez les instructions de la documentation TimescaleDB. Lors d'une mise à niveau depuis d'anciennes versions de Zabbix avec prise en charge de TimescaleDB, la compression ne sera pas activée par défaut.
Compress records older than	7d	Ce paramètre ne peut pas être inférieur à 7 jours. En raison de l'invariabilité des chunks compressés, toutes les données arrivant en retard (par exemple, des données retardées par un proxy) et plus anciennes que cette valeur seront ignorées.

Pour de meilleures performances de mise à jour des tendances, vous pouvez réduire `chunk_time_interval` pour les tables `trends` et `trends_uint` de 30 jours à 7 jours ou moins, selon le nombre d'éléments qui utilisent les tendances. L'objectif de ce paramètre est de respecter les bonnes pratiques de TimescaleDB et de garantir que la taille des chunks reste dans les limites des ressources disponibles du système.

Configuration d'Elasticsearch

Zabbix peut stocker les données d'historique dans Elasticsearch comme alternative à une base de données relationnelle.

> **Attention:**
> La prise en charge d'Elasticsearch est actuellement expérimentale.

Ce guide couvre la configuration des versions prises en charge d'Elasticsearch. Si vous utilisez une version différente, certaines fonctionnalités peuvent ne pas fonctionner comme prévu.

Elasticsearch peut stocker les types de valeurs suivants :

Item value type	Database table	Elasticsearch type
Numeric (unsigned)	history_uint	uint
Numeric (float)	history	dbl
Character	history_str	str
Log	history_log	log
Text	history_text	text
Binary	history_bin	not supported by Zabbix
JSON	history_json	json

> **Note:**
> Elasticsearch n'accepte pas les tableaux JSON. Une valeur JSON doit être soit un objet unique, soit un ensemble d'objets.

Notes importantes

- Elasticsearch nécessite libcurl. Voir requirements pour plus de détails.
- Le housekeeper ne supprime pas les données d'Elasticsearch. Pour contrôler la durée de conservation des données, voir Index Lifecycle Management.
- Zabbix ne calcule ni ne stocke les tendances dans Elasticsearch. Envisagez d'étendre la période de stockage de l'historique afin de conserver les données plus anciennes.
- Lorsque Elasticsearch est utilisé, les requêtes de plage récupérant des valeurs depuis la base de données sont limitées par l'horodatage de la période de stockage des données.
- Elasticsearch n'est pas pris en charge pour les proxies Zabbix.

Si Elasticsearch n'est pas encore installé, consultez le guide d'installation officiel avant de continuer.

Configuration d'Elasticsearch

Pour stocker les données d'historique dans Elasticsearch, vous devez :

- Créer un index pour chaque type de valeur que vous souhaitez stocker — c'est là qu'Elasticsearch stocke les données, de manière similaire à une table dans une base de données relationnelle.
- Définir un mapping pour chaque index — cela définit la structure des données, de manière similaire au schéma d'une table.
- Configurer un pipeline d'ingestion pour traiter les valeurs avant leur stockage (requis pour les valeurs JSON et les index basés sur des dates).

Elasticsearch peut stocker les données dans un seul index par type de valeur, ou dans plusieurs index basés sur des dates. Les deux approches sont décrites ci-dessous.

Stockage de l'historique dans un seul index

Dans cette approche, toutes les données d'historique pour un type de valeur donné sont écrites dans un seul index (par exemple, `uint` ou `text`).

Pour créer un index pour le type de valeur *Numérique (non signé)*, envoyez la requête suivante (avec `/uint` dans l'URL) à votre instance Elasticsearch :

```
curl -X PUT \
 http://localhost:9200/uint \
 -H 'content-type:application/json' \
 -d '{
     "settings": {
      "index": {
         "number_of_replicas": 1,
         "number_of_shards": 5
      }
   },
   "mappings": {
      "properties": {
         "itemid": { "type": "long" },
         "clock": { "format": "epoch_second", "type": "date" },
         "value": { "type": "long" }
      }
   }
}'
```

Elasticsearch répondra avec une confirmation indiquant que l'index a été créé :

```
{"acknowledged": true, "shards_acknowledged": true, "index": "uint"}
```

Des requêtes similaires doivent être envoyées pour chaque type de valeur supplémentaire que vous souhaitez stocker dans Elasticsearch.

Note:
Les mappings pour tous les types de valeur sont disponibles dans le dépôt source de Zabbix.

Par exemple, pour créer un index pour le type de valeur *Texte* :

```
curl -X PUT \
 http://localhost:9200/text \
 -H 'content-type:application/json' \
 -d '{
   "settings": {
      "index": {
         "number_of_replicas": 1,
         "number_of_shards": 5
      }
   },
   "mappings": {
      "properties": {
         "itemid": { "type": "long" },
         "clock": { "format": "epoch_second", "type": "date" },
         "value": {
            "fields": {
               "analyzed": { "index": true, "type": "text", "analyzer": "standard" }
            },
            "index": false,
            "type": "text"
         }
      }
   }
}'
```

Type de valeur JSON

Contrairement aux autres types de valeur, les valeurs JSON nécessitent un traitement supplémentaire avant le stockage.

L'index ci-dessous utilise des champs distincts pour les valeurs analysées et brutes ; un pipeline d'ingestion est donc nécessaire pour analyser chaque valeur en tant que JSON et la stocker dans le champ approprié.

Pour créer un index pour le type de valeur *JSON*, envoyez la requête suivante (avec `/json` dans l'URL) à votre instance Elasticsearch.

```
curl -X PUT \
 http://localhost:9200/json \
 -H 'content-type:application/json' \
 -d '{
   "settings": {
      "number_of_shards": 5,
      "number_of_replicas": 1
   },
   "mappings": {
      "dynamic": false,
      "properties": {
         "itemid": { "type": "long" },
         "clock": { "type": "date", "format": "epoch_second" },
         "ns": { "type": "long" },
         "value_parsed": { "type": "flattened" },
         "value_raw": { "type": "keyword", "ignore_above": 1000000 }
    }
  }
}'
```

Créez ensuite le pipeline d'ingestion :

```
curl -X PUT \
 http://localhost:9200/_ingest/pipeline/json \
 -H 'content-type:application/json' \
 -d '{
   "processors": [
      {
         "json": {
            "field": "value",
            "target_field": "value_parsed",
            "ignore_failure": true
         }
      },
      {
         "set": {
            "if": "ctx.value_parsed == null",
            "field": "value_raw",
            "value": "{{{ value }}}"
         }
      }
   ],
   "on_failure": [
      {
         "set": {
            "field": "value_raw",
            "value": "{{{ value }}}"
         }
      }
   ]
}'
```

Elasticsearch répondra par une confirmation indiquant que le pipeline d'ingestion a été créé :

```
{"acknowledged": true}
```

Stockage de l'historique dans des index basés sur la date

Au lieu d'écrire toutes les données d'historique dans un seul index (par exemple, `uint`), Elasticsearch peut répartir ces données sur plusieurs index basés sur la date (par exemple, `uint-2026-01-01`, `uint-2026-01-02`). Cela facilite la gestion du volume de données et de la rétention au fil du temps.

Pour activer cette fonctionnalité, vous devez :

- Créer un modèle d'index pour chaque type de valeur que vous souhaitez stocker — cela indique à Elasticsearch quel mapping appliquer lorsqu'il crée automatiquement un nouvel index basé sur la date.

- Créer un pipeline d'ingestion pour chaque type de valeur — il traite chaque valeur entrante et l'achemine vers le bon index basé sur la date.

Modèles d'index

Pour créer un modèle pour l'index `text`, envoyez une requête avec les détails suivants :

- Utilisez `_template/text_template` dans l'URL de votre instance Elasticsearch.
- Utilisez `"text*"` dans le champ `"index_patterns"` pour faire correspondre le nom de l'index.
- Utilisez un mapping pour le type de valeur `text` (voir les mappings dans le dépôt source de Zabbix).

```
curl -X PUT \
 http://localhost:9200/_template/text_template \
 -H 'content-type:application/json' \
 -d '{
   "index_patterns": [ "text*" ],
   "settings": {
      "index": {
         "number_of_replicas": 1,
         "number_of_shards": 5
      }
   },
   "mappings": {
      "properties": {
         "itemid": { "type": "long" },
         "clock": { "format": "epoch_second", "type": "date" },
         "value": {
            "fields": {
               "analyzed": { "index": true, "type": "text", "analyzer": "standard" }
            },
            "index": false,
            "type": "text"
         }
      }
   }
}'
```

Modèle pour l'index `json` :

```
curl -X PUT \
 http://localhost:9200/_template/json_template \
 -H 'content-type:application/json' \
 -d '{
   "index_patterns": [ "json*" ],
   "settings": {
      "number_of_shards": 5,
      "number_of_replicas": 1
   },
   "mappings": {
      "dynamic": false,
      "properties": {
         "itemid": { "type": "long" },
         "clock": { "type": "date", "format": "epoch_second" },
         "ns": { "type": "long" },
         "value_parsed": { "type": "flattened" },
         "value_raw": { "type": "keyword", "ignore_above": 1000000 }
      }
   }
}'
```

Pipelines d'ingestion

Pour créer un pipeline d'ingestion pour l'index `text` :

- Utilisez `_ingest/pipeline/text-pipeline` dans l'URL de votre instance Elasticsearch.
- Incluez un processeur `date_index_name` pour acheminer chaque valeur vers l'index basé sur la date correct en fonction de son horodatage.

```
curl -X PUT \
 http://localhost:9200/_ingest/pipeline/text-pipeline \
 -H 'content-type:application/json' \
 -d '{
   "description": "daily text index naming",
   "processors": [
      {
         "date_index_name": {
            "field": "clock",
            "date_formats": ["UNIX"],
            "index_name_prefix": "text-",
            "date_rounding": "d"
         }
      }
   ]
}'
```

Pour l'index `json`, le pipeline doit également analyser la valeur JSON avant de l'acheminer vers l'index correct :

```
curl -X PUT \
 http://localhost:9200/_ingest/pipeline/json-pipeline \
 -H 'content-type:application/json' \
 -d '{
   "description": "daily json index naming"
   "processors": [
      {
         "json": {
            "field": "value",
            "target_field": "value_parsed",
            "ignore_failure": true
         }
      },
      {
         "script": {
            "source": "if (ctx.value_parsed == null || !(ctx.value_parsed instanceof Map)) { ctx.value_raw
         }
      },
      {
         "date_index_name": {
            "field": "clock",
            "date_formats": [ "UNIX" ],
            "index_name_prefix": "json-",
            "date_rounding": "d"
         }
      }
   ]
}'
```

Configuration du serveur Zabbix

Dans le fichier de configuration de votre serveur Zabbix (`zabbix_server.conf`), définissez le paramètre `HistoryProvider`.

Par exemple, pour stocker les valeurs de type *Character*, *Log*, *Text* et *JSON* dans Elasticsearch (tout en conservant les valeurs *Numeric* dans une base de données) :

```
HistoryProvider=elasticsearch;value_types="str,log,text,json",url=http://localhost:9200
```

Si vous utilisez des index basés sur la date, ajoutez `date_index=1` au paramètre :

```
HistoryProvider=elasticsearch;value_types="str,log,text,json",url=http://localhost:9200,date_index=1
```

Après avoir effectué les modifications, redémarrez le serveur Zabbix :

```
systemctl restart zabbix-server
```

Configuration de l'interface Zabbix

Dans votre fichier de configuration de l'interface Zabbix (`zabbix.conf.php`), définissez la variable `$HISTORY_PROVIDERS` afin qu'elle corresponde à la configuration du serveur :

```
$HISTORY_PROVIDERS[] = [
  'types' => ['str','log','text','json'],
  'provider' => 'elasticsearch',
  'url' => 'http://localhost:9200'
];
```

Dépannage

Les étapes suivantes peuvent vous aider à résoudre les problèmes liés à votre configuration Elasticsearch :

1. Vérifiez les journaux d'Elasticsearch ou du serveur Zabbix pour détecter d'éventuelles erreurs.
2. Pour identifier les requêtes lentes, utilisez l'option `log_slow_queries` dans le paramètre de configuration du serveur Zabbix `HistoryProvider`.
3. Vérifiez qu'Elasticsearch autorise l'accès depuis le serveur Zabbix et l'interface Zabbix.
4. Vérifiez que `auto_create_index` est activé :

```
curl -X GET \
 "http://localhost:9200/_cluster/settings?include_defaults=true&filter_path=**.auto_create_index"

#### {"defaults": {"action": {"auto_create_index": "false"} } }
```

Pour l'activer, envoyez la requête suivante :

```
curl -X PUT \
 http://localhost:9200/_cluster/settings \
 -H 'content-type:application/json' \
 -d '{
   "persistent": {
      "action.auto_create_index": "true"
   }
}'

#### {"acknowledged": true, "persistent": {"action": {"auto_create_index": "true"} }, "transient": {} }
```

5. Vérifiez que les mappings, les modèles et les pipelines d'ingestion sont corrects en envoyant des requêtes `GET` à leurs URL respectives :

```
curl -X GET http://localhost:9200/json
curl -X GET http://localhost:9200/_template/json*
curl -X GET http://localhost:9200/_ingest/pipeline/json*
```

Vous pouvez comparer les réponses reçues avec les réponses attendues dans la documentation de l'API Elasticsearch.

6. Vérifiez si des shards sont dans un état d'échec ; le redémarrage d'Elasticsearch peut résoudre le problème.
7. Interrogez Elasticsearch pour vérifier si les données collectées par Zabbix sont stockées, par exemple :

```
curl 'http://localhost:9200/json/_search' \
 -H 'Content-Type: application/json' \
 -d '{
   "query": {
     "term": {
       "itemid": 42269
     }
   }
 }'
```

8. Si vous devez réinitialiser votre configuration Elasticsearch et recommencer, vous pouvez supprimer tous les index, modèles et pipelines d'ingestion :

```
curl -X DELETE "http://localhost:9200/_all"
curl -X DELETE "http://localhost:9200/_template/*"
curl -X DELETE "http://localhost:9200/_ingest/pipeline/*"
```

Configuration de ClickHouse

Zabbix peut stocker les données d'historique dans ClickHouse comme alternative à une base de données relationnelle.

Note:
ClickHouse peut également être utilisé comme source de données de télémétrie pour la surveillance des performances des applications (APM) ; pour plus de détails, consultez *Administration* > *Source de données*.

Ce guide couvre la configuration des versions prises en charge de ClickHouse. Si vous utilisez une autre version, certaines fonctionnalités peuvent ne pas fonctionner comme prévu.

ClickHouse peut stocker les types de valeurs suivants :

Type de valeur de l'élément	Table de base de données	Type ClickHouse
Numérique (non signé)	history_uint	uint
Numérique (flottant)	history	dbl
Caractères	history_str	str
Journal	history_log	log
Texte	history_text	text
Binaire	history_bin	non pris en charge par Zabbix
JSON	history_json	json

Note:
ClickHouse n'accepte pas les tableaux JSON. Une valeur JSON doit être soit un objet unique, soit un ensemble d'objets. De plus, ClickHouse traite les clés JSON avec NULL de la même manière que les clés manquantes.

Remarques importantes

- Le housekeeper ne supprime pas les données de ClickHouse. Pour contrôler la durée de conservation des données, configurez la période de stockage des données ClickHouse dans ClickHouse.
- Zabbix ne **calcule ni ne stocke** les tendances dans ClickHouse. Envisagez de prolonger la période de stockage de l'historique afin de conserver les données plus anciennes.
- Si vous souhaitez migrer vos données d'historique depuis une base de données Zabbix existante (MySQL ou PostgreSQL) vers ClickHouse, consultez les scripts de migration du schéma et de l'historique ClickHouse.
- ClickHouse n'est pas pris en charge comme stockage de l'historique pour les proxy Zabbix, mais peut être utilisé comme source de données de télémétrie pour un proxy Zabbix avec APM activé.
- Si vous disposez d'un cluster ClickHouse, consultez Configuration des clusters ClickHouse.

Configuration de ClickHouse

Vous devez créer et configurer une base de données Zabbix et un utilisateur, puis importer le schéma de la base de données.

Ce guide fournit des instructions pour les installations de ClickHouse via Docker ou via package.

Docker

1. Créez et configurez la base de données et l'utilisateur Zabbix lors de l'exécution du conteneur ClickHouse :

```
sudo docker run -d \
  --name clickhouse \
  -e CLICKHOUSE_DB=zabbix \
  -e CLICKHOUSE_USER=zabbix \
  -e CLICKHOUSE_DEFAULT_ACCESS_MANAGEMENT=1 \
  -e CLICKHOUSE_PASSWORD=<password> \
  -p 8123:8123/tcp \
  -p 9000:9000/tcp \
  --cap-add=SYS_NICE \
  --cap-add=NET_ADMIN \
  --cap-add=IPC_LOCK \
  --ulimit nofile=262144:262144 \
  -v clickhouse_data:/var/lib/clickhouse \
  clickhouse/clickhouse-server:26.4
```

2. Vérifiez que ClickHouse fonctionne et que vous pouvez vous y connecter :

```
sudo docker exec -it clickhouse \
  clickhouse-client \
  --query "SELECT version()"

#### 26.4.4.38
```

3. Importez le schéma de la base de données à l'aide du script history_all.sh depuis votre répertoire Zabbix :

```
./usr/share/zabbix/sql-scripts/clickhouse/history_all.sh \
  --user zabbix \
  --password <password> \
  --db zabbix \
  --server http://localhost:8123
```

Note:
La période de conservation des données (Time-To-Live, ou TTL) pour ClickHouse est, par défaut, de 31 jours. Pour la modifier, utilisez l'option `--ttl` lors de l'import du schéma de la base de données (par exemple, `--ttl 604800` pour 7 jours), ou configurez-la ultérieurement.

4. Vérifiez que les tables ont été créées :

```
sudo docker exec -it clickhouse \
  clickhouse-client \
  --user zabbix \
  --password <password> \
  --query "SHOW TABLES FROM zabbix"

#### history
#### history_json
#### history_log
#### history_str
#### history_text
#### history_uint
```

Packages

1. Démarrez ClickHouse :

```
sudo service clickhouse-server start
```

2. Créez et configurez la base de données et l'utilisateur Zabbix :

```
clickhouse-client --user default

localhost :) CREATE DATABASE IF NOT EXISTS zabbix
localhost :) CREATE USER IF NOT EXISTS zabbix IDENTIFIED WITH sha256_password BY '<password>'
localhost :) GRANT CREATE, ALTER, DROP, INSERT, SELECT, UPDATE, OPTIMIZE ON zabbix.* TO zabbix
localhost :) quit
```

3. Importez le schéma de la base de données à l'aide du script history_all.sh depuis votre répertoire Zabbix :

```
./usr/share/zabbix/sql-scripts/clickhouse/history_all.sh \
  --user zabbix \
  --password <password> \
  --db zabbix \
  --server http://localhost:8123
```

Note:
La période de conservation des données (Time-To-Live, ou TTL) pour ClickHouse est, par défaut, de 31 jours. Pour la modifier, utilisez l'option `--ttl` lors de l'import du schéma de la base de données (par exemple, `--ttl 604800` pour 7 jours), ou configurez-la ultérieurement.

4. Vérifiez que les tables ont été créées :

```
clickhouse-client \
  --user zabbix \
```

```
  --password <password> \
  --query "SHOW TABLES FROM zabbix"

#### history
#### history_json
#### history_log
#### history_str
#### history_text
#### history_uint
```

Configuration du serveur Zabbix

Dans le fichier de configuration de votre serveur Zabbix (`zabbix_server.conf`), définissez le paramètre `HistoryProvider`.

Par exemple, pour stocker tous les types de valeurs pris en charge dans ClickHouse :

```
HistoryProvider=clickhouse;value_types="uint,dbl,str,log,text,json",url=http://localhost:8123,db=zabbix,us
```

Après avoir effectué les modifications, redémarrez le serveur Zabbix :

```
systemctl restart zabbix-server
```

Configuration de l'interface Zabbix

Dans votre fichier de configuration de l'interface Zabbix (`zabbix.conf.php`), définissez la variable `$HISTORY_PROVIDERS` afin qu'elle corresponde à la configuration du serveur :

```
$HISTORY_PROVIDERS[] = [
  'types' => ['uint','dbl','str','log','text','json'],
  'provider' => 'clickhouse',
  'url' => 'http://localhost:8123',
  'db' => 'zabbix',
  'username' => 'zabbix',
  'password' => '<password>'
];
```

Configuration supplémentaire

Les étapes ci-dessous sont facultatives. Vous n'en avez pas besoin pour une configuration de base.

Configuration de la période de conservation des données ClickHouse

La période de conservation des données (Time-To-Live, ou TTL) pour ClickHouse est, par défaut, de 31 jours. Pour la modifier, exécutez les commandes ci-dessous.

Les exemples ci-dessous utilisent Docker. Si vous avez installé ClickHouse à l'aide de paquets, exécutez les requêtes directement dans le client ClickHouse.

1. Modifiez la table (remplacez `history_json` et 3600 par les valeurs dont vous avez besoin) :

```
sudo docker exec -it clickhouse \
  clickhouse-client \
  --user zabbix \
  --password <password> \
  --query "ALTER TABLE zabbix.history_json MODIFY TTL clock_ns + toIntervalSecond(3600)"
```

2. Appliquez immédiatement les modifications :

```
sudo docker exec -it clickhouse \
  clickhouse-client \
  --user zabbix \
  --password <password> \
  --query "OPTIMIZE TABLE zabbix.history_json MATERIALIZE TTL"
```

3. Vérifiez que la période de conservation des données a changé :

```
sudo docker exec -it clickhouse \
  clickhouse-client \
  --user zabbix \
  --password <password> \
  --query "SHOW CREATE TABLE zabbix.history_json"
```

```
#### CREATE TABLE zabbix.history_json
#### (
####     `itemid` UInt64,
####     `clock_ns` DateTime64(9),
####     `value` JSON,
####     `value_str` String
#### )
#### ENGINE = MergeTree
#### PARTITION BY toDate(clock_ns)
#### PRIMARY KEY (itemid, clock_ns)
#### ORDER BY (itemid, clock_ns)
#### TTL clock_ns + toIntervalSecond(3600)
#### SETTINGS index_granularity = 8192
```

4. Redémarrez le serveur Zabbix pour actualiser la période de conservation des données dans *Administration* > *Housekeeping* :

```
systemctl restart zabbix-server
```

Configuration des clusters ClickHouse

Cette section fournit les étapes de configuration des clusters ClickHouse avec réplication et sharding.

Les étapes ci-dessous utilisent Docker et le ClickHouse `cluster_2S_2R` de la documentation ClickHouse (exécutant ClickHouse 26.4.4.38 avec ClickHouse Keeper).

Avant de commencer, il est recommandé de placer un équilibreur de charge devant votre cluster ClickHouse pour Zabbix. La configuration du serveur Zabbix et de l'interface ne prend en charge qu'une seule URL ClickHouse, elle ne peut donc pointer que vers un seul noeud. Si ce noeud tombe en panne, un équilibreur de charge redirigera Zabbix vers un noeud sain à la place, via cette même URL.

1. Créez la base de données Zabbix :

```
sudo docker exec -it clickhouse-01 \
  clickhouse-client \
  --user default \
  --query "CREATE DATABASE IF NOT EXISTS zabbix ON CLUSTER cluster_2S_2R ENGINE = Replicated('/clickhouse/
```

> **Attention:**
> La clause `ON CLUSTER` n'atteint que les noeuds qui font partie du cluster au moment où vous exécutez la commande. Si vous ajoutez des noeuds à votre cluster ultérieurement, exécutez cette commande à nouveau.

2. Créez et configurez l'utilisateur Zabbix :

```
sudo docker exec -it clickhouse-01 \
  clickhouse-client \
  --user default \
  --query "CREATE USER IF NOT EXISTS zabbix ON CLUSTER cluster_2S_2R IDENTIFIED WITH sha256_password BY '<

sudo docker exec -it clickhouse-01 \
  clickhouse-client \
  --user default \
  --query "GRANT ON CLUSTER cluster_2S_2R CREATE, ALTER, DROP, INSERT, SELECT, UPDATE, OPTIMIZE ON zabbix.
```

3. Importez le schéma de base de données à l'aide du script `history_all.sh` depuis votre répertoire Zabbix :

```
./usr/share/zabbix/sql-scripts/clickhouse/history_all.sh \
  --user zabbix \
  --password <password> \
  --db zabbix \
  --server http://localhost:8123 \
  --engine "ReplicatedMergeTree()"
```

Note:
La période de conservation des données (Time-To-Live, ou TTL) pour ClickHouse est, par défaut, de 31 jours. Pour la modifier, utilisez l'option `--ttl` lors de l'import du schéma de base de données (par exemple, `--ttl 604800` pour 7 jours), ou configurez la plus tard.

Note:
Si votre cluster ClickHouse nécessite des chemins de réplique explicites (par exemple, si la configuration de votre serveur ClickHouse ne définit pas de chemins par défaut pour les tables répliquées), exécutez les scripts de schéma (`history_schema.sh`, `history_uint_schema.sh`, etc.) individuellement et définissez un chemin différent pour chaque table :

```
./usr/share/zabbix/sql-scripts/clickhouse/history_schema.sh \
  --user zabbix \
  --password <password> \
  --db zabbix \
  --server http://localhost:8123 \
  --engine "ReplicatedMergeTree('/clickhouse/tables/{shard}/history', '{replica}')"
```

Pour plus de détails sur la réplication ClickHouse, consultez Replicating data et Replicated* table engines dans la documentation ClickHouse.

4. Configurez le sharding pour chaque type de table. Renommez les tables et créez une table `Distributed` avec le nom d'origine, pointant vers la table renommée. Cette table `Distributed` est celle que Zabbix lit et écrit réellement ; la table répartit automatiquement les données entre les shards. Par exemple, pour la table `history` :

```
sudo docker exec -it clickhouse-01 \
  clickhouse-client \
  --user default \
  --query "RENAME TABLE zabbix.history TO zabbix.history_local"

sudo docker exec -it clickhouse-01 \
  clickhouse-client \
  --user default \
  --query "CREATE TABLE zabbix.history AS zabbix.history_local ENGINE = Distributed('cluster_2S_2R', 'zabb
```

Note:
Pour plus de détails sur le sharding ClickHouse, consultez Distributed table engine dans la documentation ClickHouse.

5. Configurez le serveur Zabbix.

6. Configurez l'interface Zabbix.

7. Vérifiez que les données Zabbix sont réparties entre les shards. Le nombre de lignes de `zabbix.history_local` peut changer entre deux vérifications, car le serveur Zabbix insère continuellement de nouvelles données. Exécutez cette commande sur chaque noeud :

```
sudo docker exec -it clickhouse-01 \
  clickhouse-client \
  --user default \
  --query "SELECT count(*) FROM zabbix.history_local"

#### 2446 (clickhouse-01)
#### 2471 (clickhouse-02)
#### 2494 (clickhouse-03)
```

8. Vérifiez que la réplication est synchronisée (`active_replicas` doit correspondre à `total_replicas`) :

```
sudo docker exec -it clickhouse-01 \
  clickhouse-client \
  --user default \
  --query "SELECT database, table, is_leader, total_replicas, active_replicas FROM system.replicas WHERE d

#### zabbix  history_json_local  1 2 2
#### zabbix  history_local       1 2 2
#### zabbix  history_log_local   1 2 2
```

```
#### zabbix  history_str_local   1 2 2
#### zabbix  history_text_local  1 2 2
#### zabbix  history_uint_local  1 2 2
```

Dépannage

Les étapes suivantes peuvent vous aider à résoudre les problèmes liés à votre configuration ClickHouse :

1. Vérifiez les journaux de ClickHouse ou du serveur Zabbix pour détecter des erreurs.
2. Pour identifier les requêtes lentes, utilisez l'option `log_slow_queries` dans le paramètre de configuration du serveur Zabbix `HistoryProvider`.
3. Vérifiez que ClickHouse autorise l'accès depuis le serveur Zabbix et l'interface Zabbix.
4. Interrogez ClickHouse pour vérifier si les données collectées par Zabbix sont stockées, par exemple :

```
sudo docker exec -it clickhouse \
  clickhouse-client \
  --user zabbix \
  --password <password> \
  --query "SELECT * FROM zabbix.history_uint WHERE itemid = 42269"
```

Notes spécifiques aux distributions pour configurer NGINX pour Zabbix

RHEL

Sur RHEL 8, 9 et 10, NGINX est inclus dans AppStream : aucun dépôt supplémentaire n'est nécessaire. Activez-le et installez-le simplement avec :

```
dnf module list nginx
dnf -y install nginx
```

Sur RHEL 7, NGINX peut être installé depuis le dépôt CentOS 7 Extras (si vous utilisez CentOS) avec :

```
yum -y install nginx
```

Si vous préférez utiliser les toutes dernières versions en amont, configurez le dépôt officiel NGINX dans `/etc/yum.repos.d/nginx.repo` :

```
cat > /etc/yum.repos.d/nginx.repo << 'EOF'
[nginx-stable]
name=nginx stable repo
baseurl=https://nginx.org/packages/rhel/$releasever/$basearch/
gpgcheck=1
enabled=1
gpgkey=https://nginx.org/keys/nginx_signing.key
EOF
dnf -y install nginx
```

SLES 15

Dans SUSE Linux Enterprise Server 15, vous devez configurer `php-fpm` (le chemin vers le fichier de configuration peut varier légèrement selon le service pack) :

```
cp /etc/php8/fpm/php-fpm.conf{.default,}
cp /etc/php8/fpm/php-fpm.d/www.conf{.default,}
sed -i 's/user = nobody/user = wwwrun/; s/group = nobody/group = www/' /etc/php8/fpm/php-fpm.d/www.conf
```

Configuration SAML avec Microsoft Entra ID

Vue d'ensemble

Cette section fournit des directives pour configurer l'authentification unique et l'approvisionnement des utilisateurs dans Zabbix à partir de Microsoft Entra ID (anciennement Microsoft Azure Active Directory) à l'aide de l'authentification SAML 2.0.

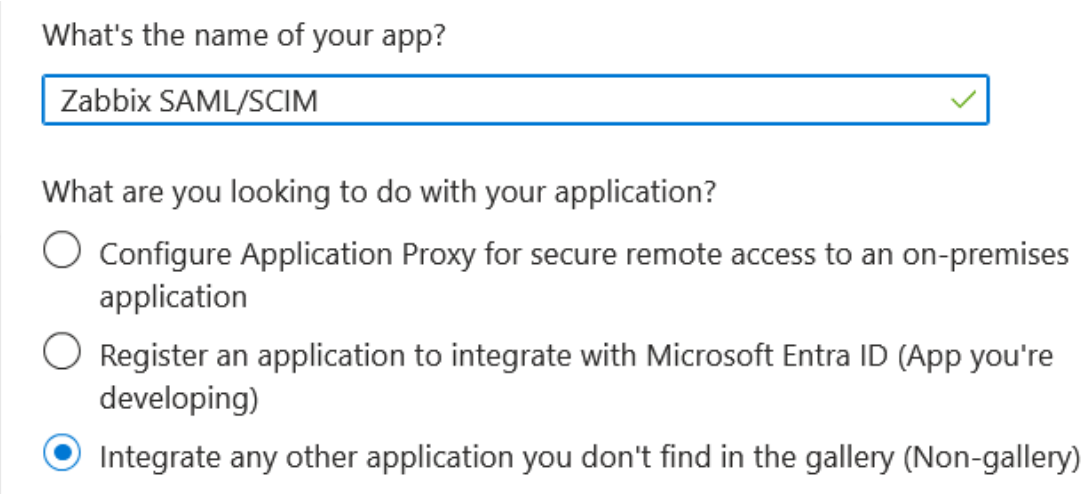
Configuration de Microsoft Entra ID

Création d'une application

1. Connectez-vous au centre d'administration Microsoft Entra à l'adresse Microsoft Entra ID. À des fins de test, vous pouvez créer un compte d'essai gratuit dans Microsoft Entra ID.

2. Dans le centre d'administration Microsoft Entra, sélectionnez *Applications* > *Applications d'entreprise* > *Nouvelle application* > *Créer votre propre application*.

3. Ajoutez le nom de votre application et sélectionnez l'option *Intégrer toute autre application...*. Ensuite, cliquez sur *Créer*.

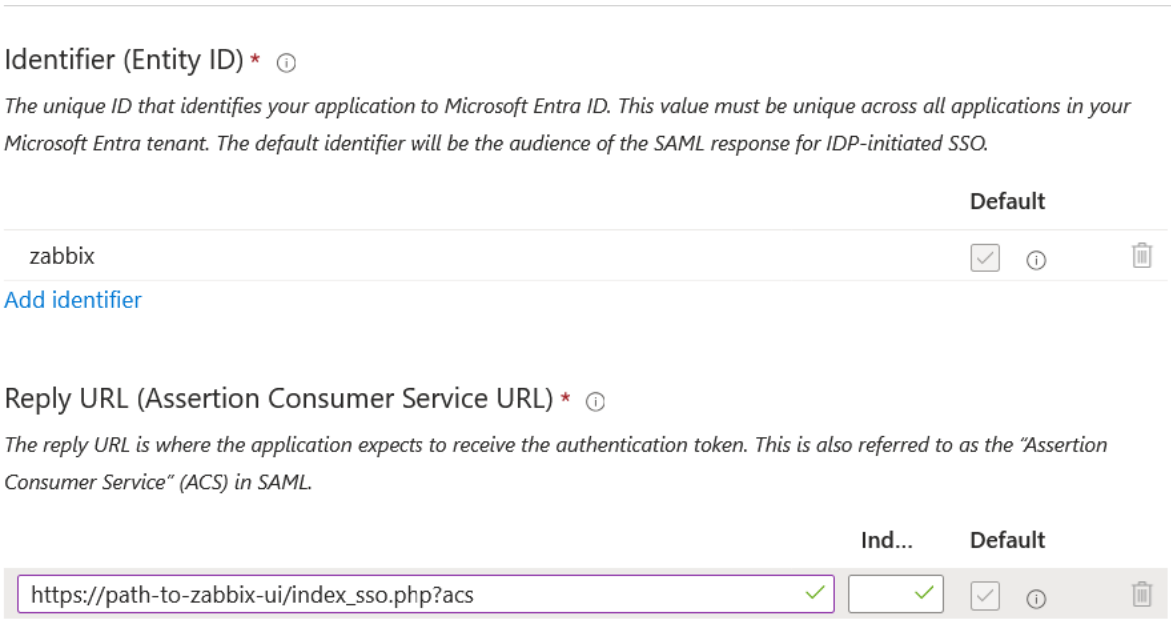


Configuration de l'authentification unique

1. Dans la page de votre application, accédez à *Set up single sign on* et cliquez sur *Get started*. Sélectionnez ensuite *SAML*.

2. Modifiez *Basic SAML Configuration* :

- Dans *Identifier (Entity ID)*, définissez un nom unique permettant d'identifier votre application auprès de Microsoft Entra ID, par exemple `zabbix` ;
- Dans *Reply URL (Assertion Consumer Service URL)*, définissez le point de terminaison d'authentification unique de Zabbix : `https://<path-to-zabbix-ui>/index_sso.php?acs` :



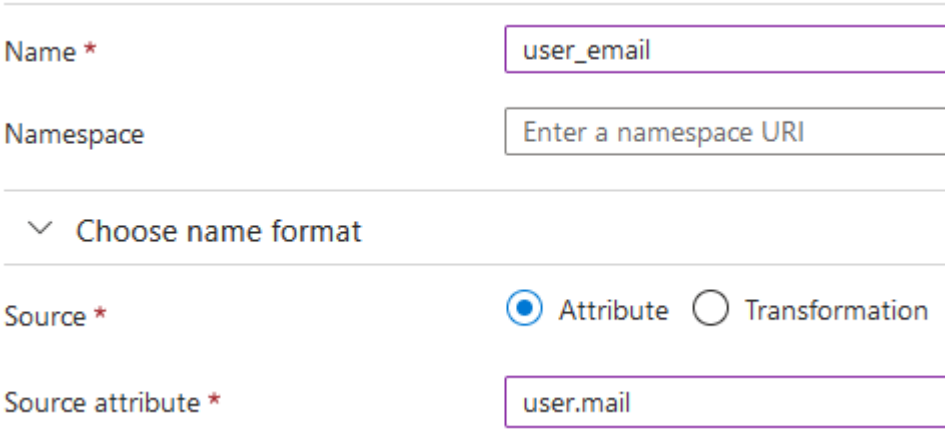
Notez que `https` est obligatoire. Pour utiliser cette fonctionnalité avec Zabbix, ajoutez la ligne suivante au fichier de configuration de l'interface :

```
$SSO['SETTINGS'] = ['use_proxy_headers' => true];
```

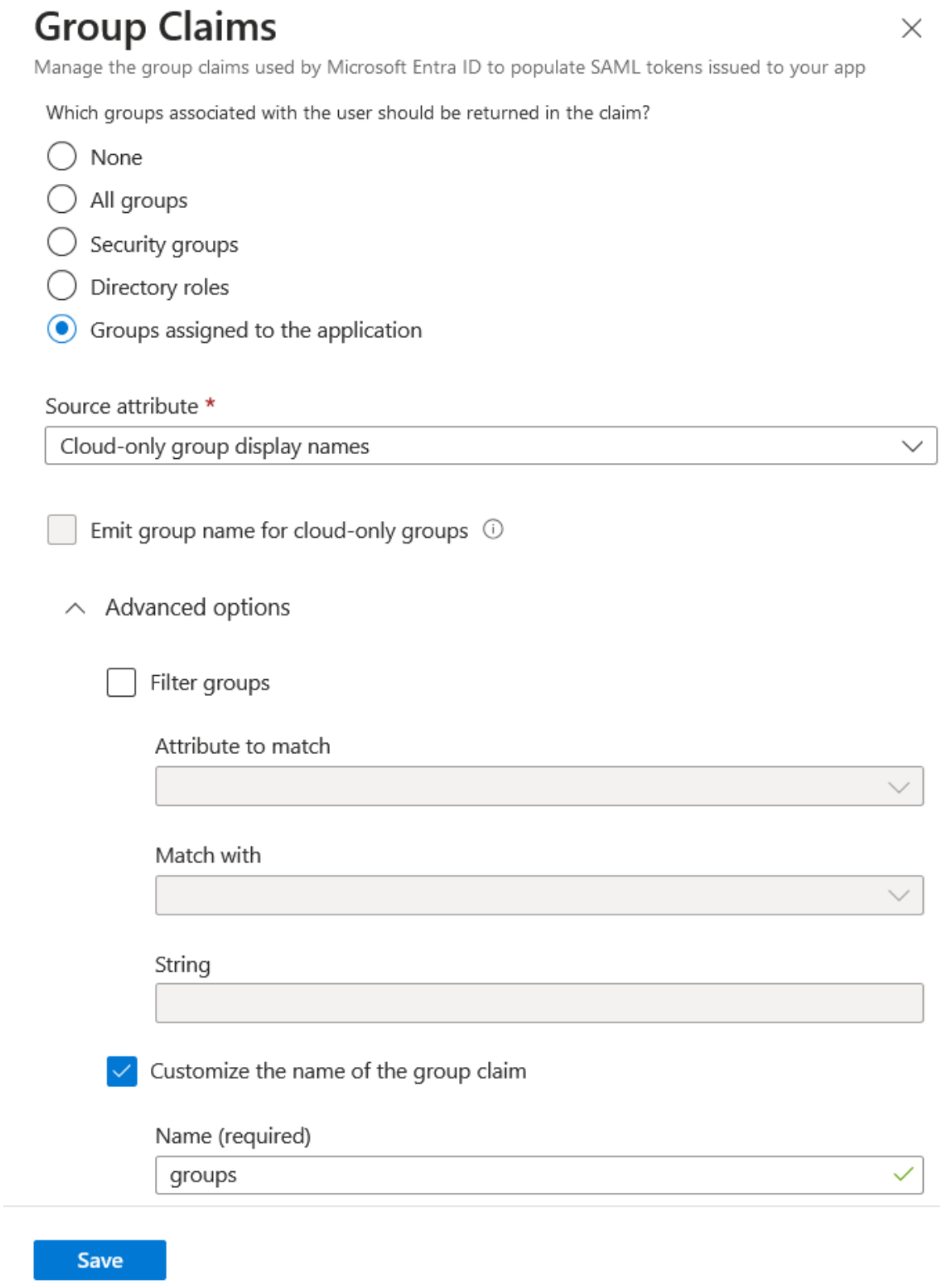
3. Modifiez *Attributes & Claims*. Vous devez ajouter tous les attributs que vous souhaitez transmettre à Zabbix (user_name, user_lastname, user_email, user_mobile, groups).

Les noms des attributs sont arbitraires. D'autres noms d'attributs peuvent être utilisés, mais ils doivent correspondre à la valeur du champ respectif dans les paramètres SAML de Zabbix.

- Cliquez sur *Add new claim* pour ajouter un attribut :



- Cliquez sur *Add a group claim* pour ajouter un attribut permettant de transmettre les groupes à Zabbix :



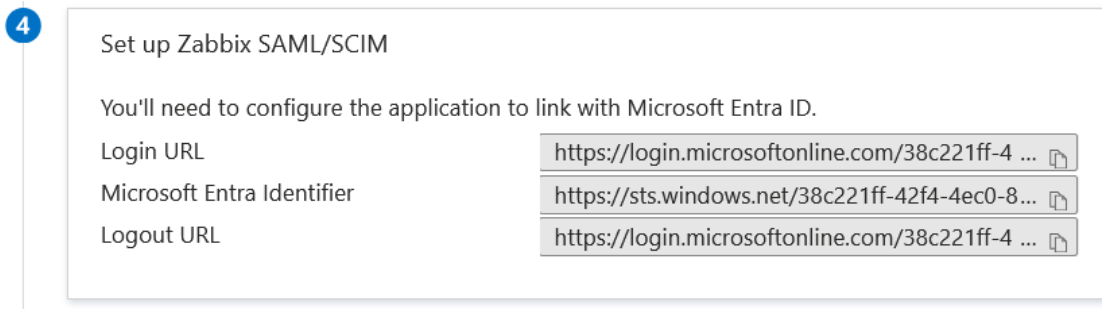
Dans cette revendication, il est important que les noms des groupes (et non leurs identifiants) soient transmis à Zabbix par l'*attribut source* sélectionné. Sinon, le provisionnement JIT des utilisateurs ne fonctionnera pas correctement.

4. Dans *SAML Certificates*, téléchargez le certificat Base64 fourni par Entra ID et placez-le dans `conf/certs` de l'installation de l'interface Zabbix (l'emplacement peut varier selon le système d'exploitation et la source de votre installation Zabbix).

Définissez les permissions à 644 en exécutant :

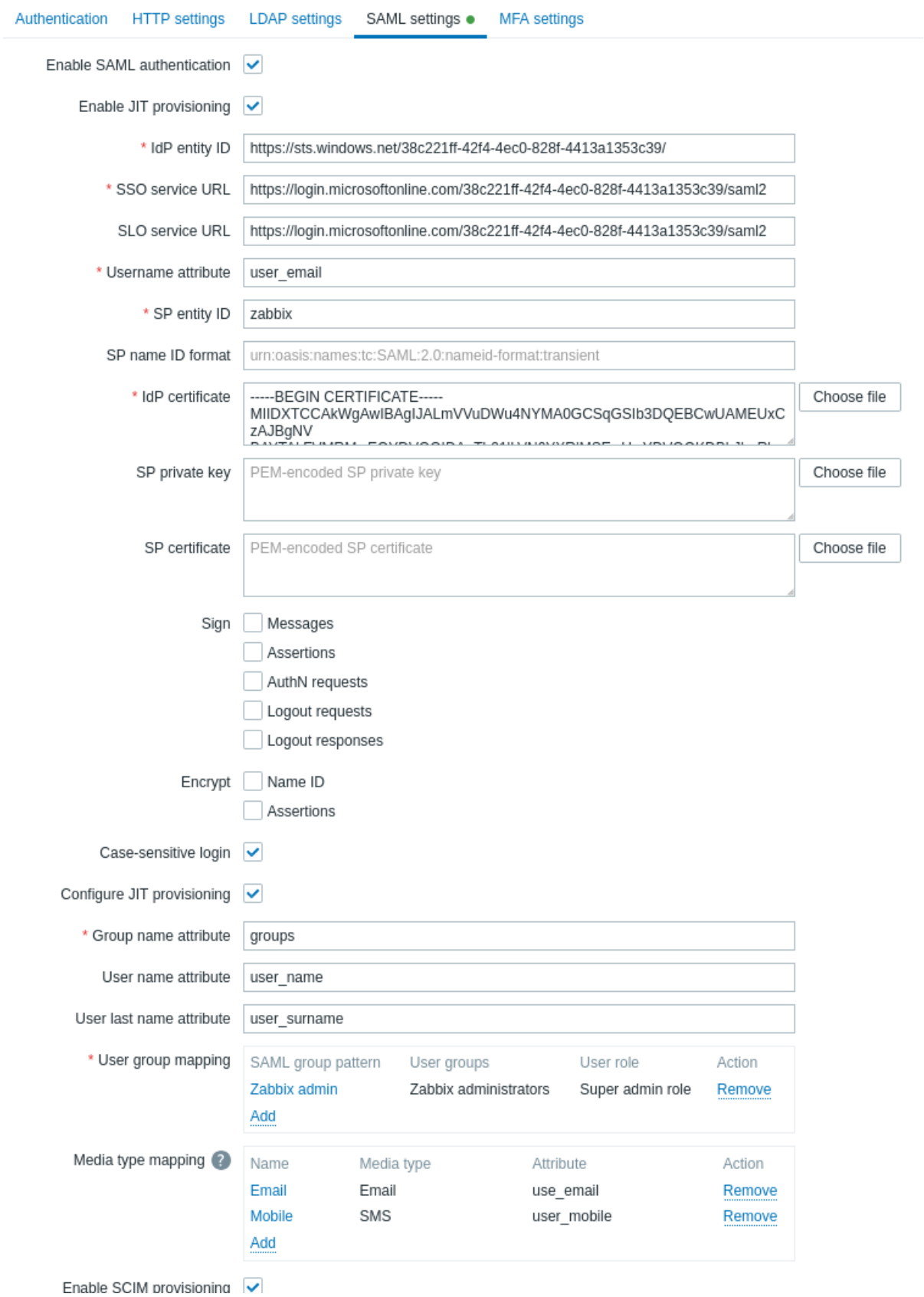
```
chmod 644 entra.cer
```

5. Utilisez les valeurs de *Set up <your app name>* dans Entra ID pour configurer l'authentification SAML de Zabbix (voir la section suivante) :



Configuration de Zabbix

1. Dans Zabbix, accédez aux paramètres SAML et renseignez les options de configuration en fonction de la configuration d'Entra ID :



Champ Zabbix	Champ de configuration dans Entra ID	Exemple de valeur
ID d'entité IdP	Identificateur Microsoft Entra	
URL du service SSO	URL de connexion	
URL du service SLO	URL de déconnexion	
ID d'entité SP	Identificateur (ID d'entité)	
Attribut du nom d'utilisateur	Attribut personnalisé (revendication)	`user_email`
Attribut du nom du groupe	Attribut personnalisé (revendication)	`groups`
Attribut du prénom de l'utilisateur	Attribut personnalisé (revendication)	`user_name`

Champ Zabbix	Champ de configuration dans Entra ID	Exemple de valeur
Attribut du nom de famille de l'utilisateur	Attribut personnalisé (revendication)	`user_lastname`

2.Ajoutez le certificat Base64 fourni par Entra ID.

Si `$SSO['CERT_STORAGE'] = 'database'` est défini dans le fichier de configuration de l'interface (`zabbix.conf.php`), vous pouvez coller le texte du certificat ou téléverser le fichier de certificat dans l'interface pendant la configuration SAML : aucun fichier n'est requis sur le système de fichiers.

Si `$SSO['CERT_STORAGE'] = 'file'` est défini dans le fichier de configuration, le certificat doit être disponible sur le système de fichiers (par défaut dans `ui/conf/certs` ou dans le chemin configuré dans le fichier de configuration) et l'interface ne stockera pas les certificats dans la base de données. Notez que si `$SSO['CERT_STORAGE']` n'est pas défini ou est commenté, le stockage dans un fichier est utilisé et les certificats sont lus depuis `ui/conf/certs`.

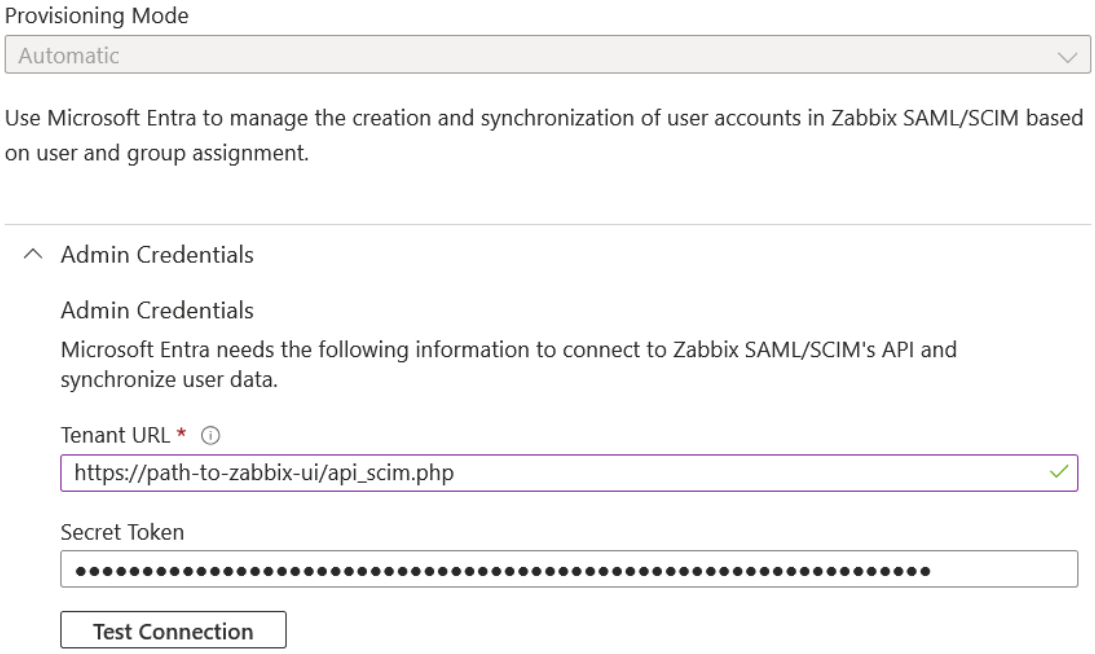
Il est également nécessaire de configurer le mappage des groupes d'utilisateurs. Le mappage des médias est facultatif.

3. Cliquez sur le bouton *Mettre à jour* pour enregistrer ces paramètres.

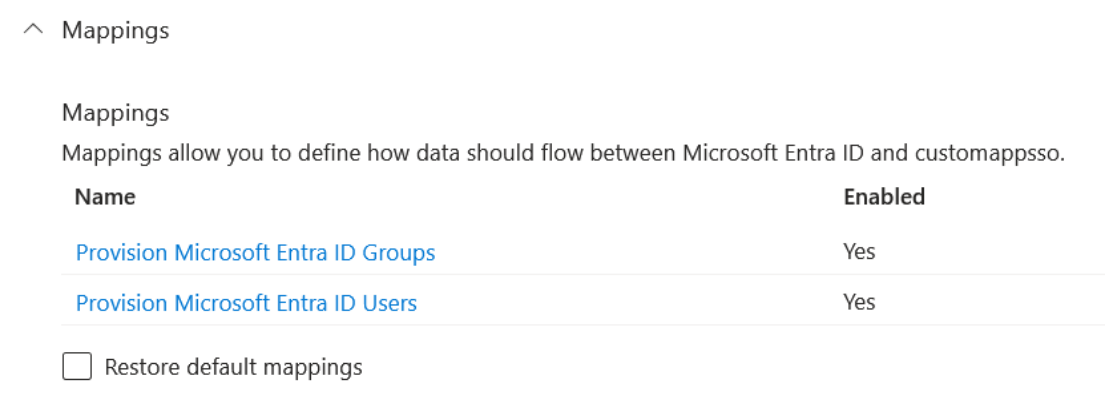
Provisionnement SCIM des utilisateurs

1. Dans la page de votre application Entra ID, ouvrez la page Provisioning depuis le menu principal.
Cliquez sur *Get started*, puis sélectionnez le mode de provisionnement automatique :

- Dans *Tenant URL*, définissez la valeur suivante : `https://<path-to-zabbix-ui>/api_scim.php`
- Dans *Secret token*, saisissez un jeton API Zabbix avec les permissions Super admin.
- Cliquez sur *Test connection* pour vérifier si la connexion est établie.

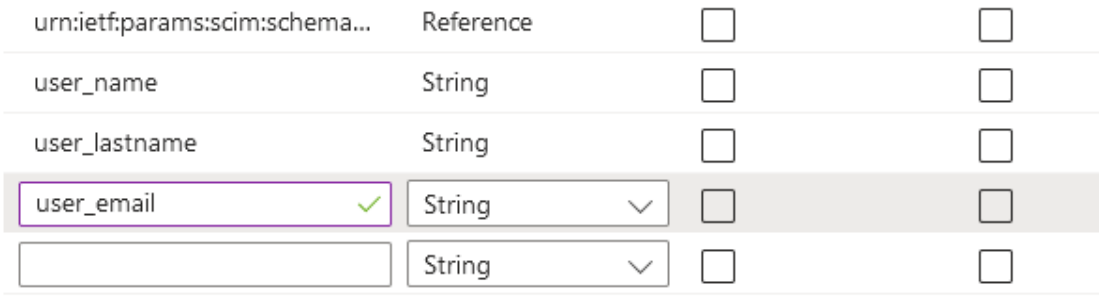


2. Vous pouvez maintenant ajouter tous les attributs qui seront transmis via SCIM à Zabbix.
Pour ce faire, cliquez sur *Mappings*, puis sur *Provision Microsoft Entra ID Users*.



En bas de la liste Attribute Mapping, □□□□□□ez *Show advanced options*, puis cliquez sur *Edit attribute list for customappsso*.

En bas de la liste des attributs, ajoutez vos propres attributs avec le type 'String' :



Enregistrez la liste.

3. Vous pouvez maintenant ajouter des mappages pour les attributs ajoutés.
En bas de la liste Attribute Mapping, cliquez sur *Add New Mapping* et créez les mappages comme indiqué ci-dessous :



Une fois tous les mappages ajoutés, enregistrez la liste des mappages.



4. Comme prérequis au provisionnement des utilisateurs dans Zabbix, vous devez avoir configuré des utilisateurs et des groupes dans Entra ID.

Pour ce faire, accédez au *Microsoft Entra admin center*, puis ajoutez des utilisateurs/groupes dans les pages Users et Groups correspondantes.

5. Une fois les utilisateurs et les groupes créés dans Entra ID, vous pouvez aller dans le menu *Users and groups* de votre application et les ajouter à l'application.

6. Allez dans le menu *Provisioning* de votre application, puis cliquez sur *Start provisioning* pour provisionner les utilisateurs vers Zabbix.

Notez que la requête PATCH Users dans Entra ID ne prend pas en charge les changements de média.

Signature des demandes d'authentification

Il est possible de configurer Entra ID pour valider la signature des demandes d'authentification signées.

Pour que cela fonctionne, créez les clés publique et privée :

```
openssl req -x509 -newkey rsa:4096 -keyout /usr/share/zabbix/conf/certs/request-sign.key -out /usr/share/z
```

Attribuez les permissions :

```
chown apache /usr/share/zabbix/conf/certs/request-sign.key
chmod 400 /usr/share/zabbix/conf/certs/request-sign.key
```

Mettez à jour la configuration de l'interface Zabbix en ajoutant :

```
$SSO['SP_KEY'] = 'conf/certs/request-sign.key';
$SSO['SP_CERT'] = 'conf/certs/request-sign.crt';
```

Dépannage

Des problèmes d'authentification peuvent survenir avec les navigateurs Microsoft Edge lorsqu'un utilisateur qui tente de se connecter à Zabbix via SAML est déjà connecté avec le profil Microsoft Edge. L'utilisateur peut notamment parvenir à se connecter à Zabbix avec Microsoft Edge en mode privé.

Pour éviter les problèmes d'authentification dans ce cas, il peut être nécessaire de définir `requestedAuthnContext` sur `false` dans le fichier de configuration de l'interface de Zabbix.

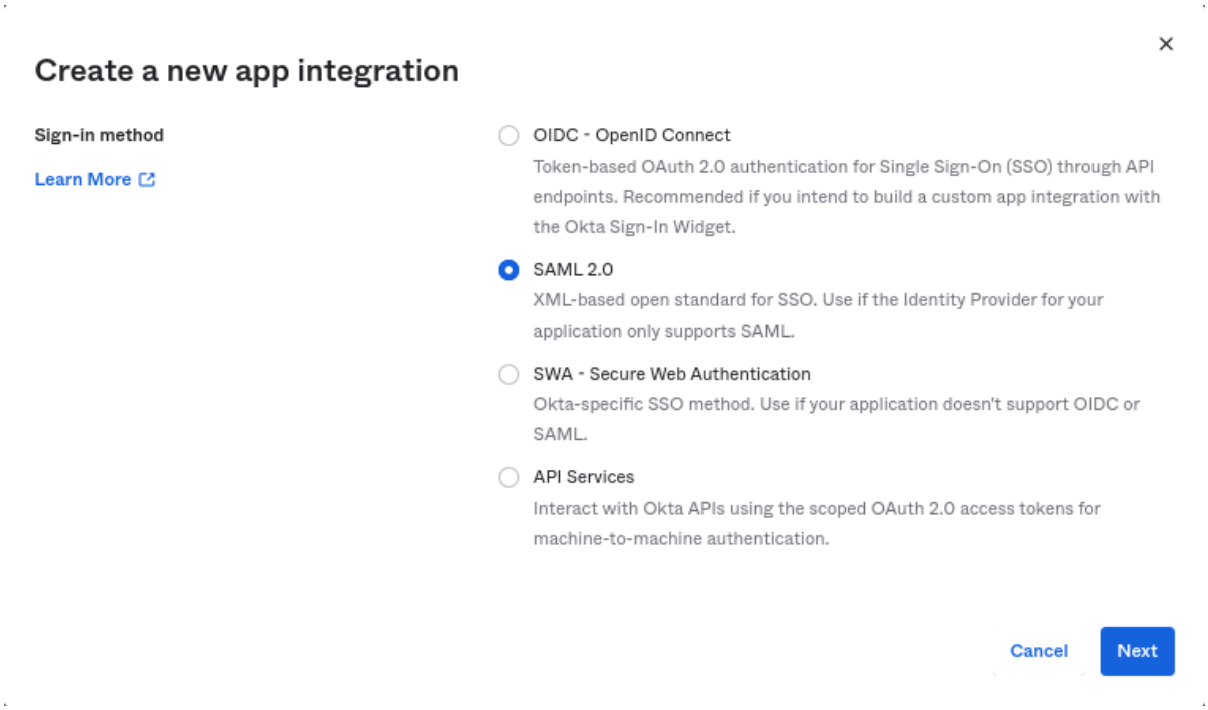
```
$SSO['SETTINGS'] = [
    'security' => [
        'requestedAuthnContext' => false
    ]
];
```

Configuration SAML avec Okta

Cette section fournit des recommandations pour configurer Okta afin d'activer l'authentification SAML 2.0 et le provisionnement des utilisateurs pour Zabbix.

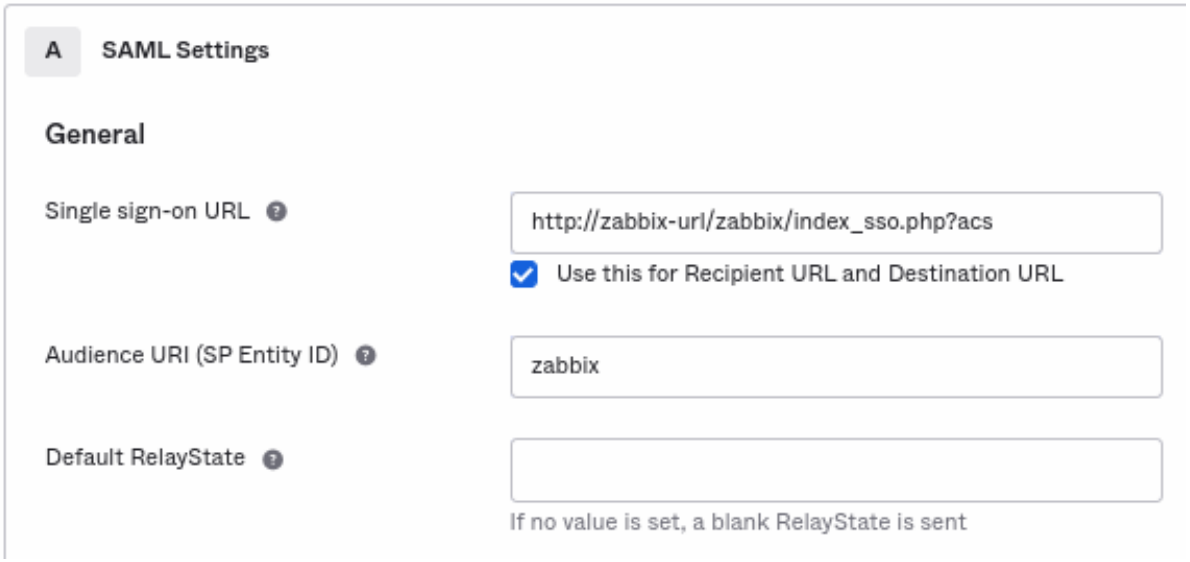
Configuration Okta

1. Accédez à https://developer.okta.com/signup/ et inscrivez-vous/connectez-vous à votre compte.
2. Dans l'interface web Okta, accédez à *Applications* > *Applications*.
3. Cliquez sur *Create App Integration*.

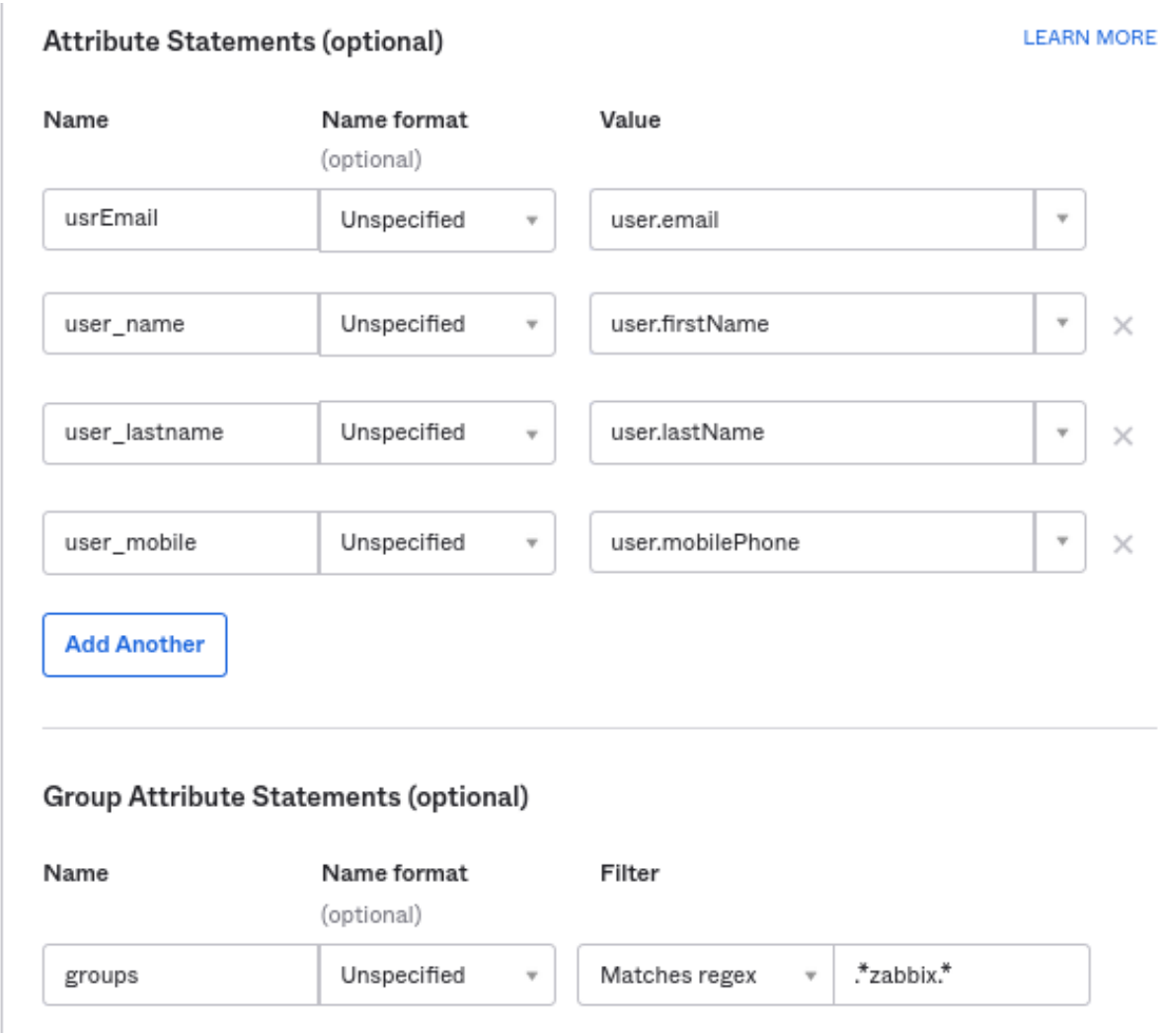


Sélectionnez "SAML 2.0" comme méthode de connexion, puis cliquez sur *Next*.

4. Dans les paramètres généraux, renseignez le nom de l'application, puis cliquez sur *Next*.
5. Dans la configuration SAML, saisissez les valeurs indiquées ci-dessous, puis cliquez sur *Next*.



- Dans **General**, ajoutez :
  - *Single sign-on URL* : `http://<your-zabbix-url>/zabbix/index_sso.php?acs<br>` Notez l'utilisation de "http" et non de "https", afin que le paramètre `acs` ne soit pas supprimé dans la requête. La case *Use this for Recipient URL and Destination URL* doit également être cochée.
  - *Audience URI (SP Entity ID)* : `zabbix<br>` Notez que cette valeur sera utilisée dans l'assertion SAML comme identifiant unique du fournisseur de services (si elle ne correspond pas, l'opération sera rejetée). Il est possible de spécifier une URL ou toute autre chaîne de caractères dans ce champ.
  - *Default RelayState* :
 Laissez ce champ vide ; si une redirection personnalisée est nécessaire, elle peut être ajoutée dans Zabbix dans les paramètres *Users* > *Users*.
  - Renseignez les autres champs selon vos préférences.
- Dans **Attribute Statements/Group Attribute Statements**, ajoutez :



Ces déclarations d'attributs sont insérées dans les assertions SAML partagées avec Zabbix.

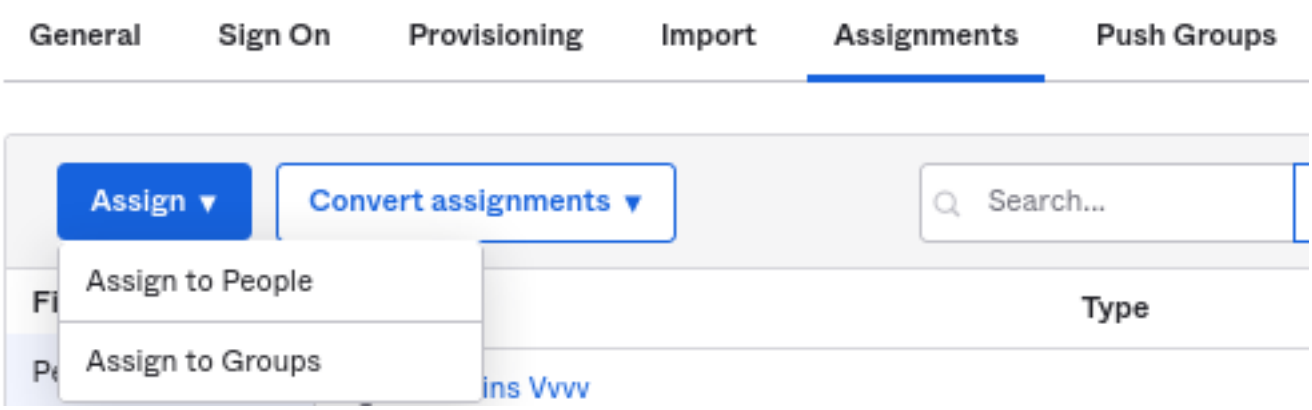
Les noms d'attributs utilisés ici ne sont que des exemples arbitraires. Vous pouvez utiliser d'autres noms d'attributs ; toutefois, ils doivent correspondre à la valeur du champ respectif dans les paramètres SAML de Zabbix.

Si vous souhaitez configurer la connexion SAML à Zabbix *sans* provisionnement JIT des utilisateurs, seul l'attribut e-mail est requis.

Note:
Si vous prévoyez d'utiliser une connexion chiffrée, générez les certificats de chiffrement privé et public, puis téléversez le certificat public vers Okta. Le formulaire de téléversement du certificat apparaît lorsque *Assertion Encryption* est défini sur "Encrypted" (cliquez sur *Show Advanced Settings* pour trouver ce paramètre).

6. Dans l'onglet suivant, sélectionnez "I'm a software vendor. I'd like to integrate my app with Okta" puis cliquez sur *Finish*.

7. Accédez à l'onglet "Assignments" de l'application nouvellement créée et cliquez sur le bouton *Assign*, puis sélectionnez *Assign to People* dans la liste déroulante.



8. Dans la fenêtre contextuelle qui apparaît, attribuez l'application aux personnes qui utiliseront SAML 2.0 pour s'authentifier auprès de Zabbix, puis cliquez sur *Save and go back*.

9. Téléchargez le certificat IdP. Définissez les permissions 644 sur celui-ci en exécutant :

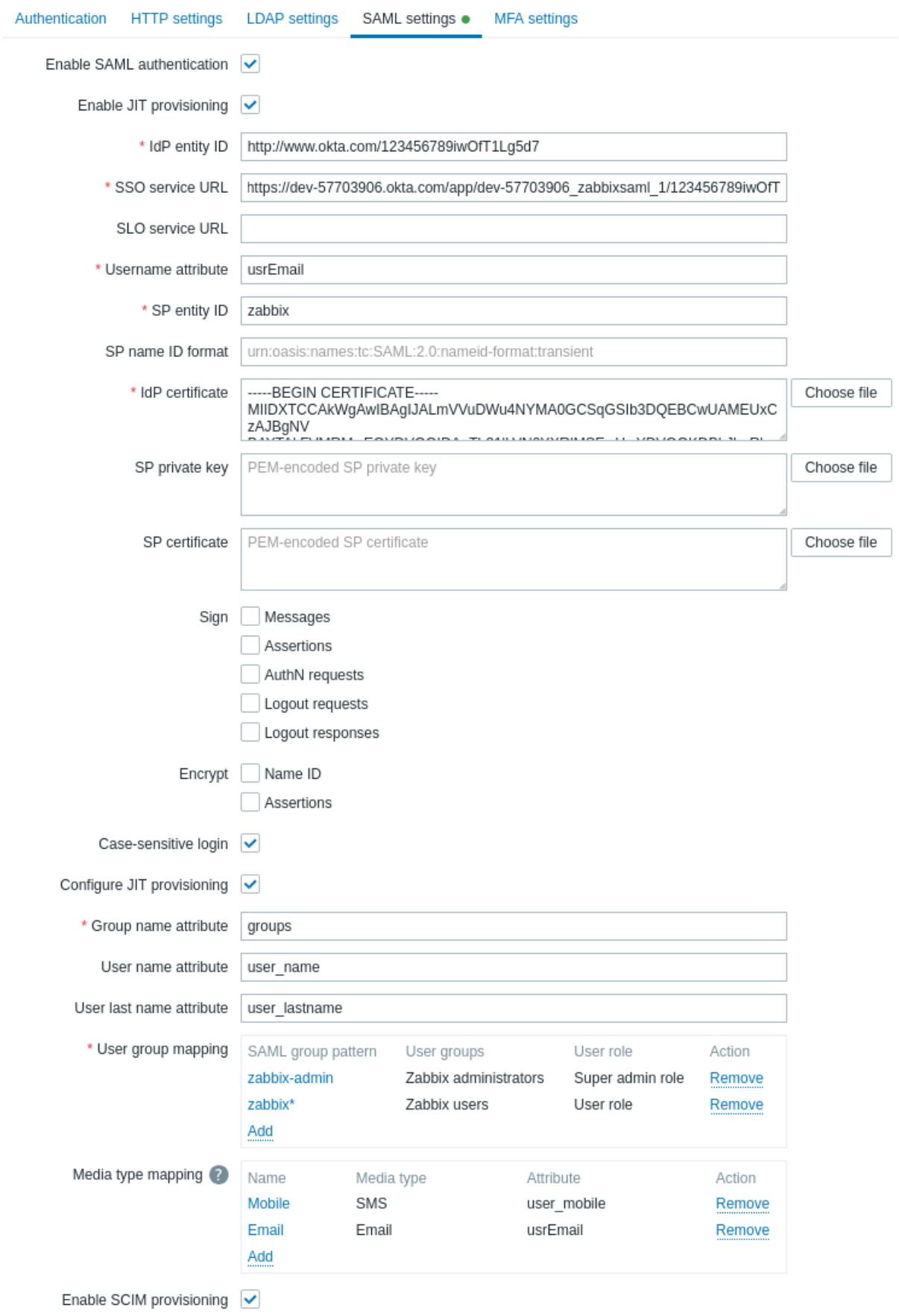
```
chmod 644 idp.crt
```

10. Accédez à l'onglet "Sign On" et cliquez sur le bouton *View Setup Instructions*.

Les **instructions** de configuration s'ouvriront dans un nouvel onglet ; laissez cet onglet ouvert pendant la configuration de Zabbix.

Configuration de Zabbix

1. Dans Zabbix, accédez aux paramètres SAML et renseignez les options de configuration en fonction des instructions de configuration d'Okta :



Champ Zabbix	Champ de configuration dans Okta	Exemple de valeur
ID d'entité IdP	Identity Provider Issuer	
URL du service SSO	Identity Provider Single Sign-On URL	
Attribut du nom d'utilisateur	Attribute name	`usrEmail`
ID d'entité SP	Audience URI	`zabbix`
Attribut du nom du groupe	Attribute name	`groups`

Champ Zabbix	Champ de configuration dans Okta	Exemple de valeur
Attribut du prénom de l'utilisateur	Attribute name	`user_name`
Attribut du nom de famille de l'utilisateur	Attribute name	`user_lastname`

Il est également nécessaire de configurer le mappage des groupes d'utilisateurs et des médias.

2. Ajoutez le certificat Base64 fourni dans les instructions de configuration SAML d'Okta.

Si `$SSO['CERT_STORAGE'] = 'database'` est défini dans le fichier de configuration de l'interface (`zabbix.conf.php`), vous pouvez coller le texte du certificat ou téléverser le fichier de certificat dans l'interface lors de la configuration SAML : aucun fichier n'est requis sur le système de fichiers.

Si `$SSO['CERT_STORAGE'] = 'file'` est défini dans le fichier de configuration, le certificat doit être disponible sur le système de fichiers (par défaut dans `ui/conf/certs` ou dans le chemin configuré dans le fichier de configuration) et l'interface ne stockera pas les certificats dans la base de données. Notez que si `$SSO['CERT_STORAGE']` n'est pas défini ou est commenté, le stockage dans un fichier est utilisé et les certificats sont lus depuis `ui/conf/certs`.

3. Si *Assertion Encryption* est défini sur « Encrypted » dans Okta, la case « Assertions » du paramètre *Encrypt* doit également être cochée dans Zabbix.

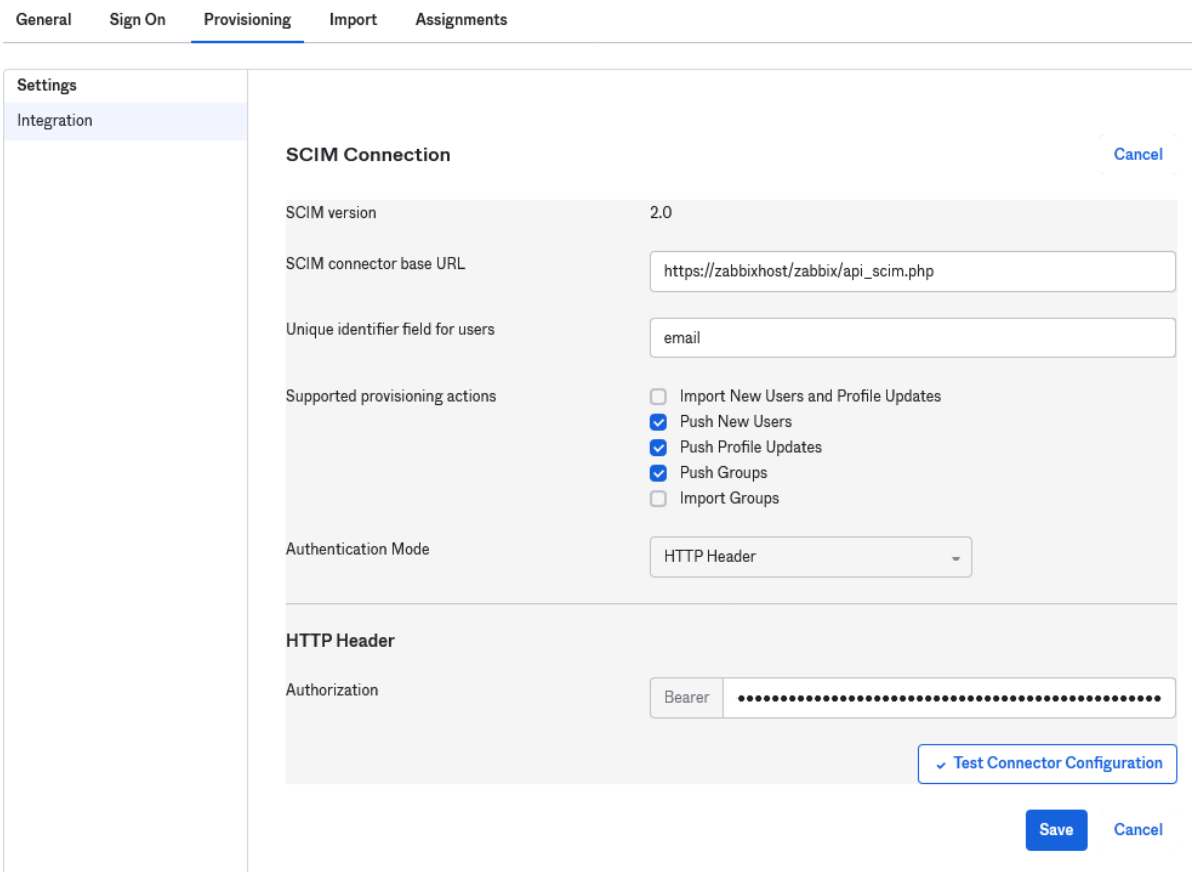
4. Cliquez sur le bouton *Update* pour enregistrer ces paramètres.

Provisionnement SCIM

1. Pour activer le provisionnement SCIM, accédez à *General* > *App Settings* de l'application dans Okta.

Cochez la case *Enable SCIM provisioning*. Un nouvel onglet *Provisioning* apparaît alors.

2. Accédez à l'onglet « Provisioning » pour configurer une connexion SCIM : - Dans *SCIM connector base URL*, indiquez le chemin vers l'interface Zabbix et ajoutez `api_scim.php`, c'est-à-dire :
 `https://<your-zabbix-url>/zabbix/api_scim.php` - *Unique identifier field for users* : `email` - *Authentication mode* : `HTTP header` - Dans *Authorization*, saisissez un token API valide avec des droits de Super administrateur.



Attention:
Si vous rencontrez des problèmes d'authentification, consultez Authorization header forwarding.

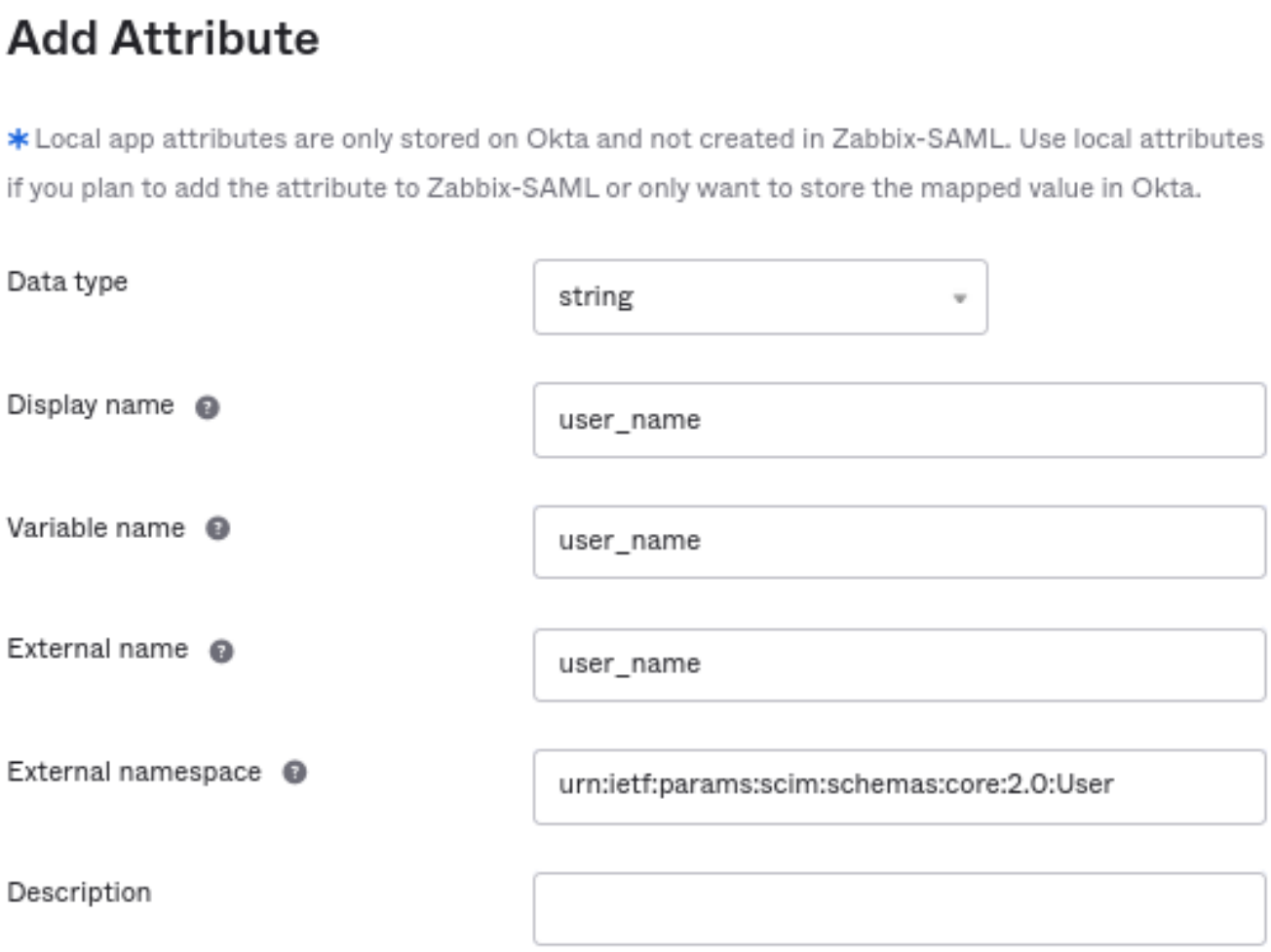
3. Cliquez sur *Test Connector Configuration* pour tester la connexion. Si tout est correct, un message de réussite s'affiche.

4. Dans *Provisioning* > *To App*, veillez à cocher les cases suivantes : - Create Users - Update User Attributes - Deactivate Users

Cela garantit que ces types de requêtes seront envoyés à Zabbix.

5. Vérifiez que tous les attributs définis dans SAML sont également définis dans SCIM. Vous pouvez accéder à l'éditeur de profil de votre application dans *Provisioning* > *To App*, en cliquant sur *Go to Profile Editor*.

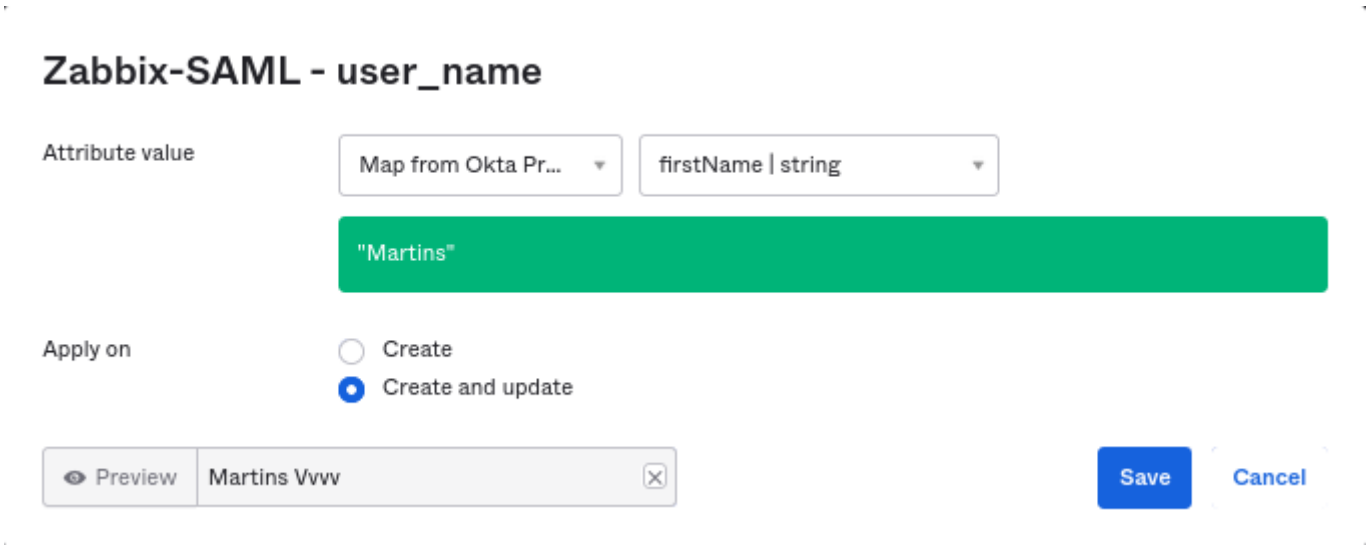
Cliquez sur *Add Attribute*. Renseignez les valeurs *Display name*, *Variable name* et *External name* avec le nom de l'attribut SAML, par exemple `user_name`.



External namespace doit être identique au schéma utilisateur : `urn:ietf:params:scim:schemas:core:2.0:User`

6. Accédez à *Provisioning* > *To App* > *Attribute Mappings* de votre application. Cliquez sur *Show Unmapped Attributes* en bas de la page. Les attributs nouvellement ajoutés apparaissent.

7. Mappez chaque attribut ajouté.



8. Ajoutez des utilisateurs dans l'onglet « Assignments ». Les utilisateurs doivent auparavant avoir été ajoutés dans *Directory* > *People*. Toutes ces affectations seront envoyées sous forme de requêtes à Zabbix.

9. Ajoutez des groupes dans l'onglet « Push Groups ». Le modèle de mappage des groupes d'utilisateurs dans les paramètres SAML de Zabbix doit correspondre à un groupe spécifié ici. En l'absence de correspondance, l'utilisateur ne peut pas être créé dans Zabbix.

Les informations sur les membres des groupes sont envoyées chaque fois qu'une modification est effectuée.

Configuration SAML avec OneLogin

Vue d'ensemble

Cette section fournit des directives pour configurer l'authentification unique et le provisionnement des utilisateurs dans Zabbix depuis OneLogin à l'aide de l'authentification SAML 2.0.

Configuration OneLogin

Création de l'application

1. Connectez-vous à votre compte sur OneLogin. À des fins de test, vous pouvez créer un compte développeur gratuit dans OneLogin.

2. Dans l'interface web OneLogin, accédez à *Applications* > *Applications*.

3. Cliquez sur "Add App" et recherchez l'application appropriée. Les instructions de cette page sont basées sur l'exemple d'application *SCIM Provisioner with SAML (SCIM v2 Enterprise, full SAML)*.

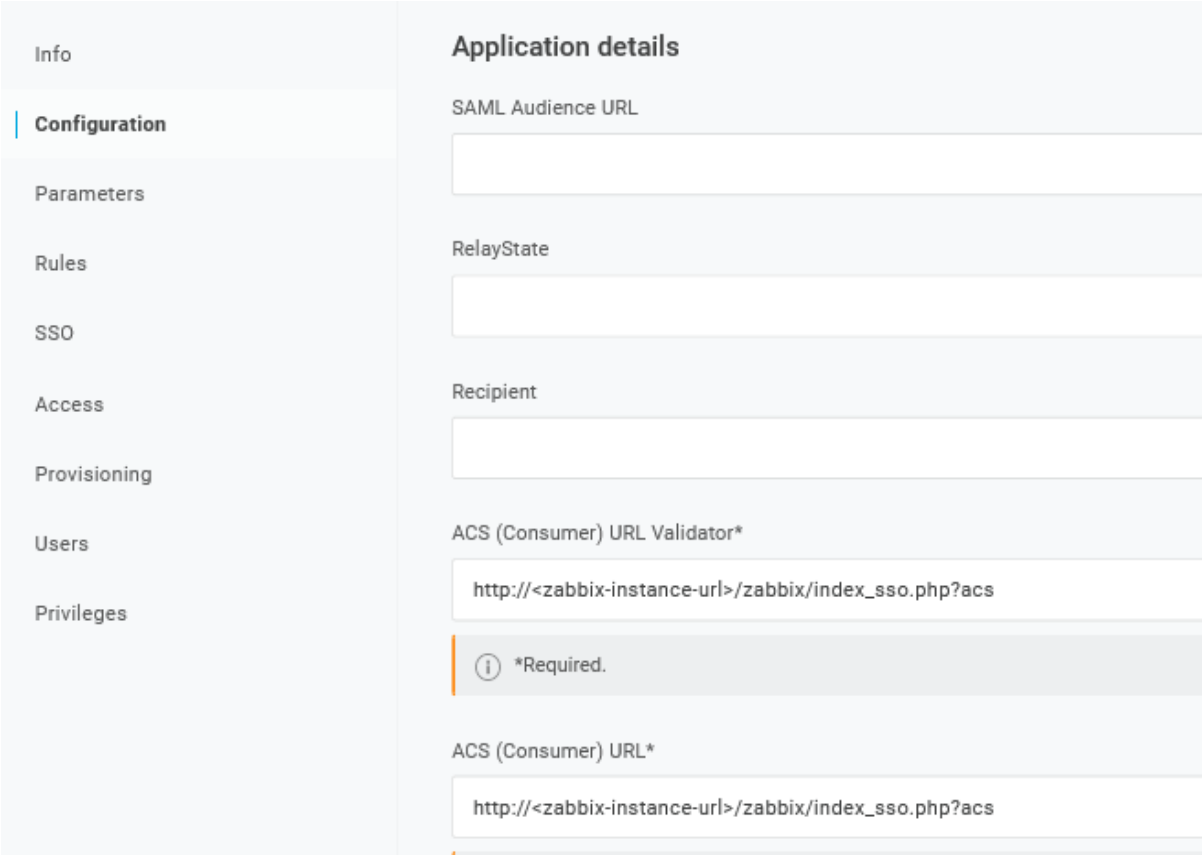
4. Pour commencer, vous pouvez souhaiter personnaliser le nom d'affichage de votre application. Vous pouvez également ajouter l'icône et les détails de l'application. Ensuite, cliquez sur *Save*.

Configuration du provisionnement SSO/SCIM

1. Dans *Configuration* > *Application details*, définissez le point de terminaison d'authentification unique Zabbix `http://<zabbix-instance-` comme valeur des champs suivants :

- *ACS (Consumer) URL Validator*
- *ACS (Consumer) URL*

Notez l'utilisation de « http », et non de « https », afin que le paramètre `acs` ne soit pas supprimé de la requête.



Il est également possible d'utiliser « https ». Pour l'utiliser avec Zabbix, ajoutez la ligne suivante au fichier de configuration de l'interface (`zabbix.conf.php`) :

```
$SSO['SETTINGS'] = ['use_proxy_headers' => true];
```

Laissez les autres options avec leurs valeurs par défaut.

2. Dans *Configuration* > *API connection*, définissez les valeurs suivantes :

- *SCIM Base URL* : `https://<zabbix-instance-url>/zabbix/api_scim.php`
- *SCIM JSON Template* : doit contenir tous les attributs personnalisés que vous souhaitez transmettre à Zabbix via SCIM, tels que `user_name`, `user_lastname`, `user_email` et `user_mobile` :

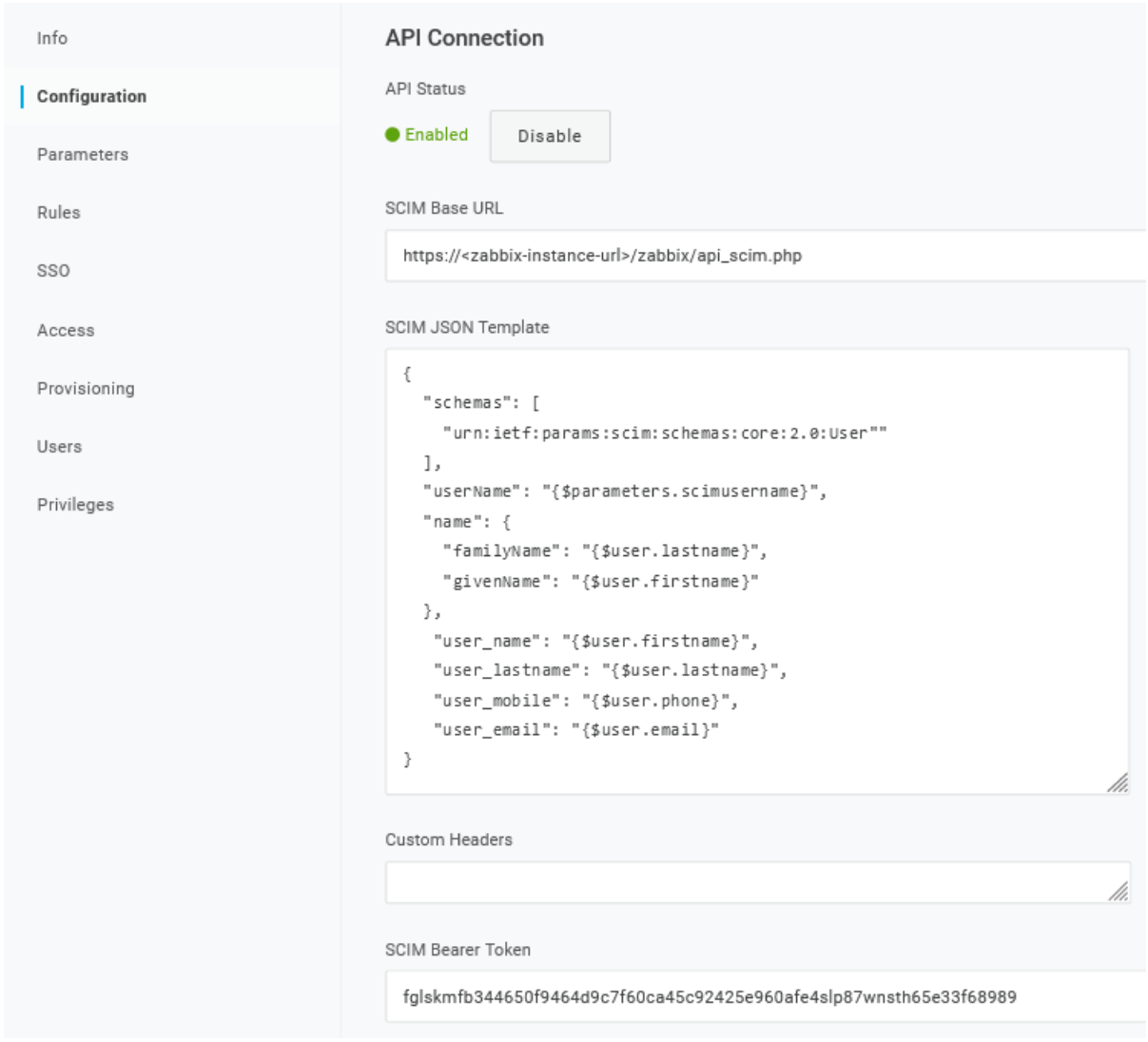
```
{
  "schemas": [
    "urn:ietf:params:scim:schemas:core:2.0:User"
  ],
  "userName": "{$parameters.scimusername}",
  "name": {
    "familyName": "{$user.lastname}",
    "givenName": "{$user.firstname}"
  },
   "user_name": "{$user.firstname}",
   "user_lastname": "{$user.lastname}",
   "user_mobile": "{$user.phone}",
   "user_email": "{$user.email}"
}
```

Les noms des attributs sont arbitraires. D'autres noms d'attributs peuvent être utilisés, mais ils doivent correspondre à la valeur du champ respectif dans les paramètres SAML de Zabbix.

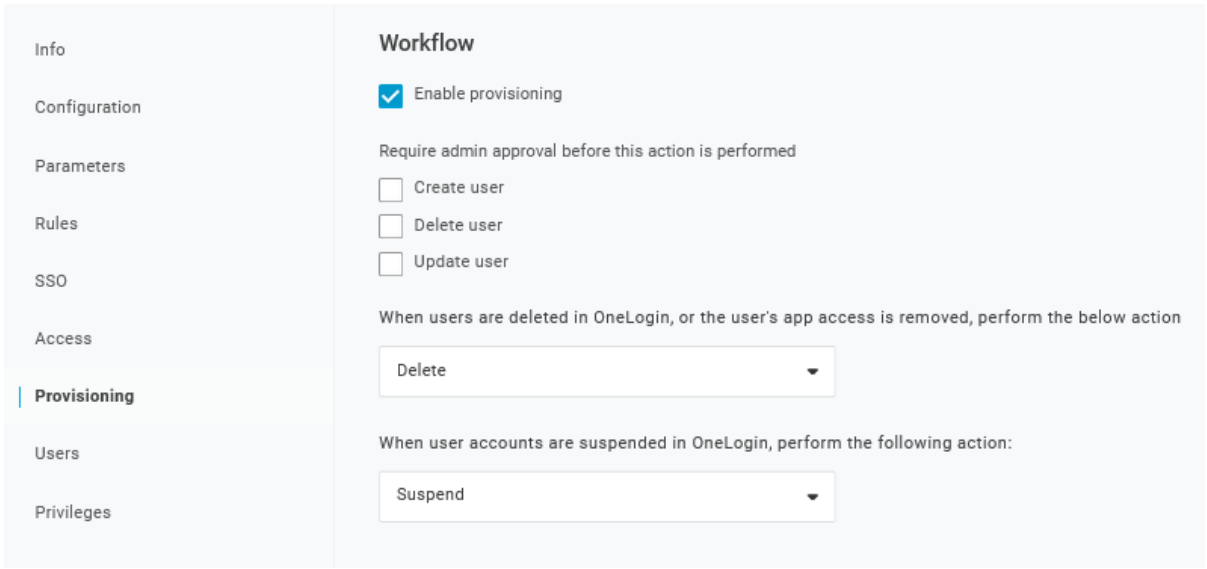
Notez que, pour que le provisionnement des utilisateurs fonctionne, OneLogin doit recevoir en réponse un attribut « name » avec « givenName » et « familyName », même si celui-ci n'était pas requis par le fournisseur de services. Il est donc nécessaire de le spécifier dans le schéma, dans la partie de configuration de l'application.

- *SCIM Bearer Token* : saisissez un jeton d'API Zabbix disposant des permissions de super administrateur.

Cliquez sur *Enable* pour activer la connexion.

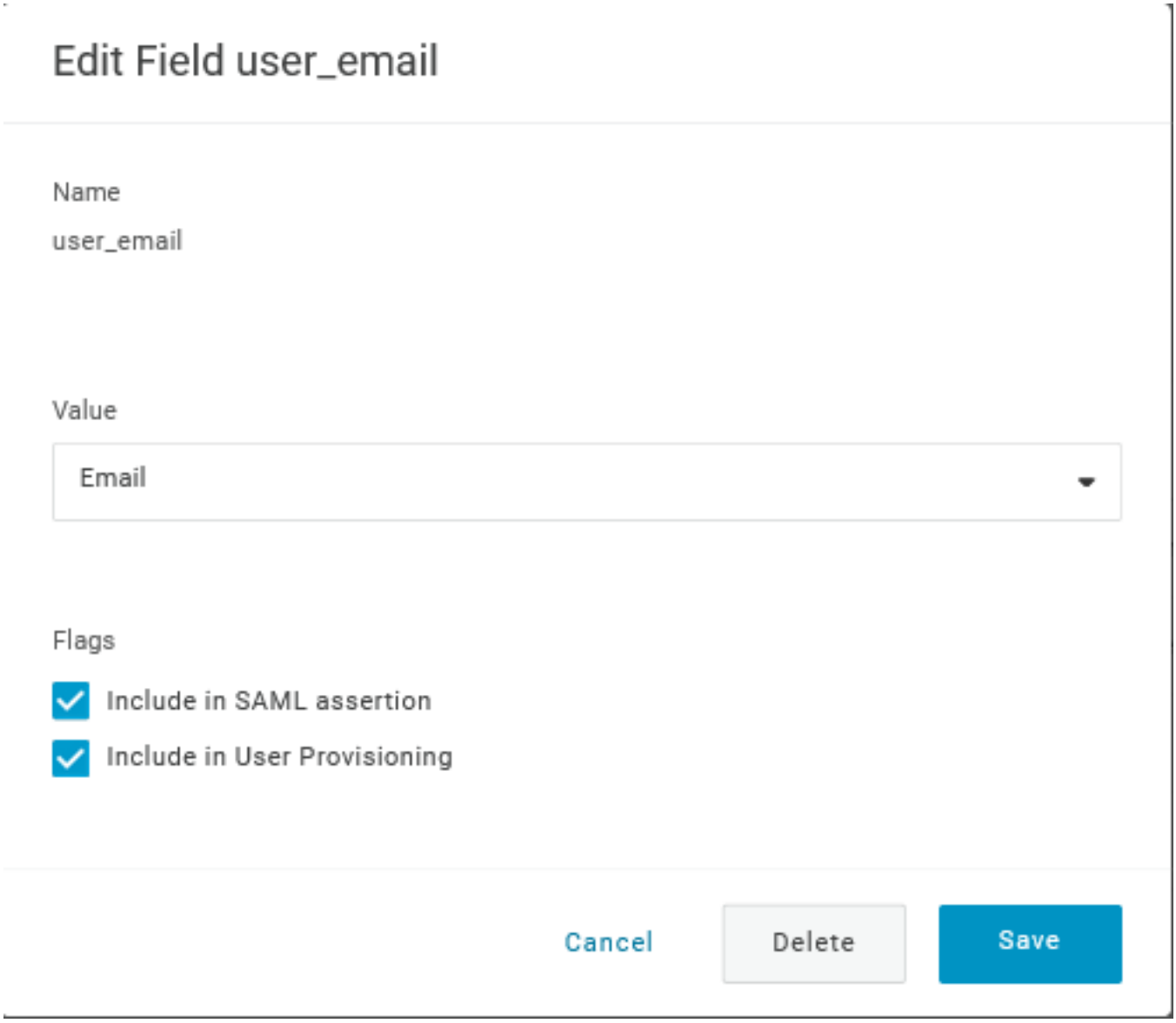


3. Dans la page *Provisioning*, activez l'option de provisionnement :



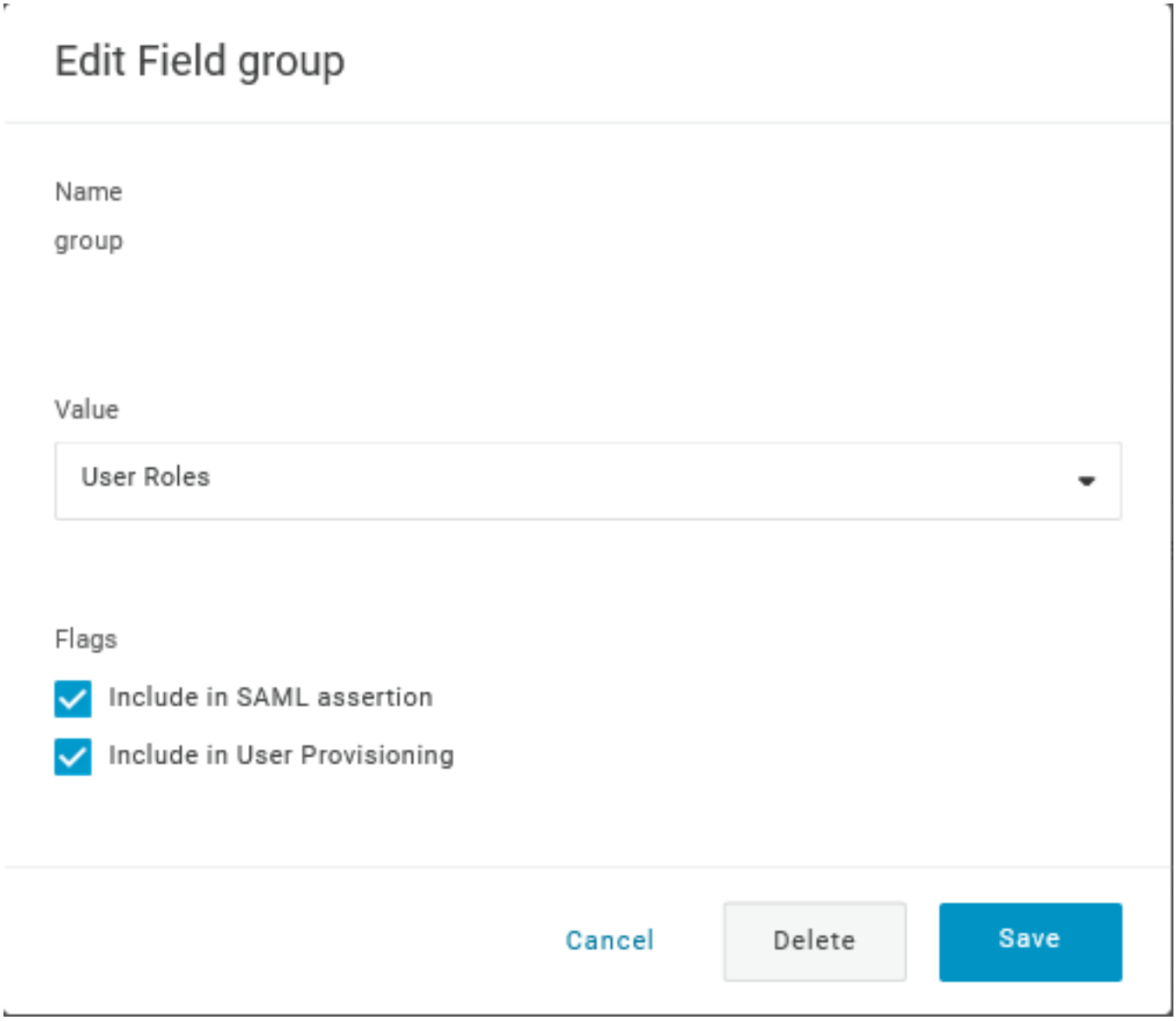
4. La page *Parameters* contient une liste de paramètres par défaut :

   - Vérifiez que « scimusername » correspond à la valeur de connexion de l'utilisateur dans OneLogin (par exemple, l'adresse e-mail) ;
   - Activez l'option *Include in User Provisioning* pour le paramètre « Groups » ;
   - Cliquez sur « + » pour créer les paramètres personnalisés requis pour les assertions SAML et le provisionnement des utilisateurs, tels que `user_name`, `user_lastname`, `user_email` et `user_mobile` :

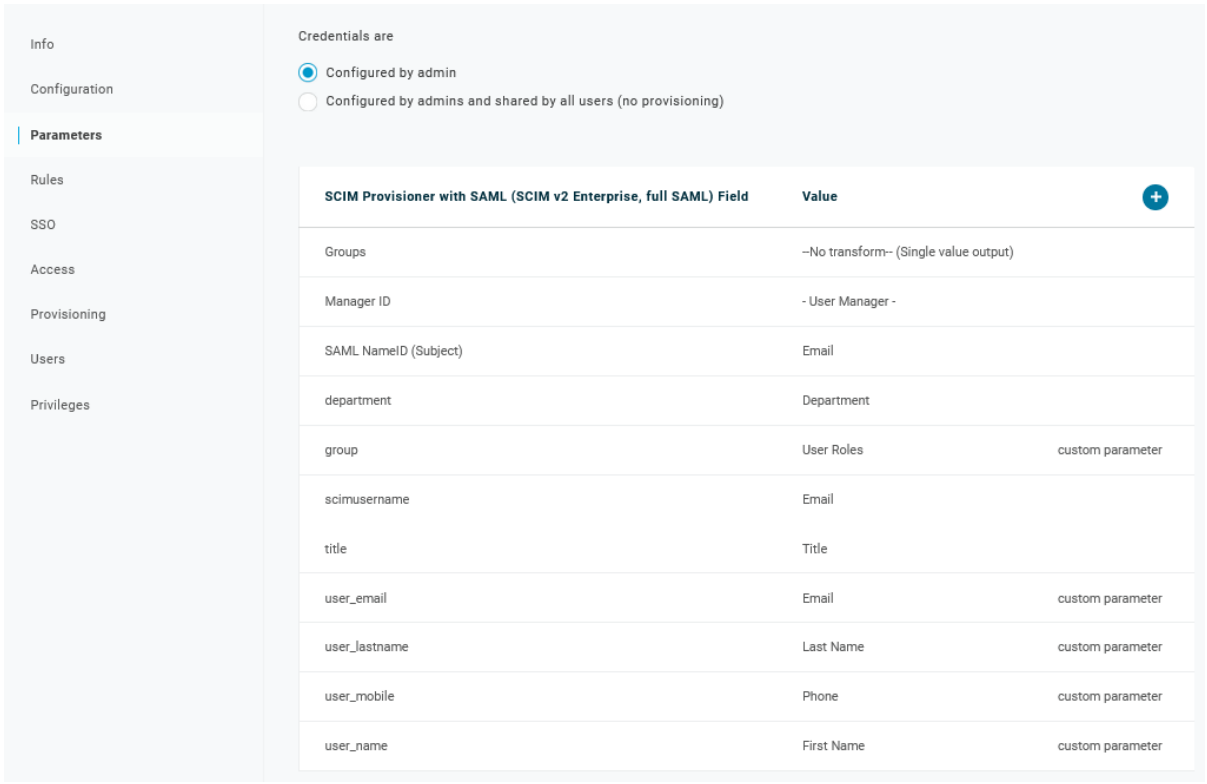


Lors de l'ajout d'un paramètre, veillez à activer les options *Include in SAML assertion* et *Include in User Provisioning*.

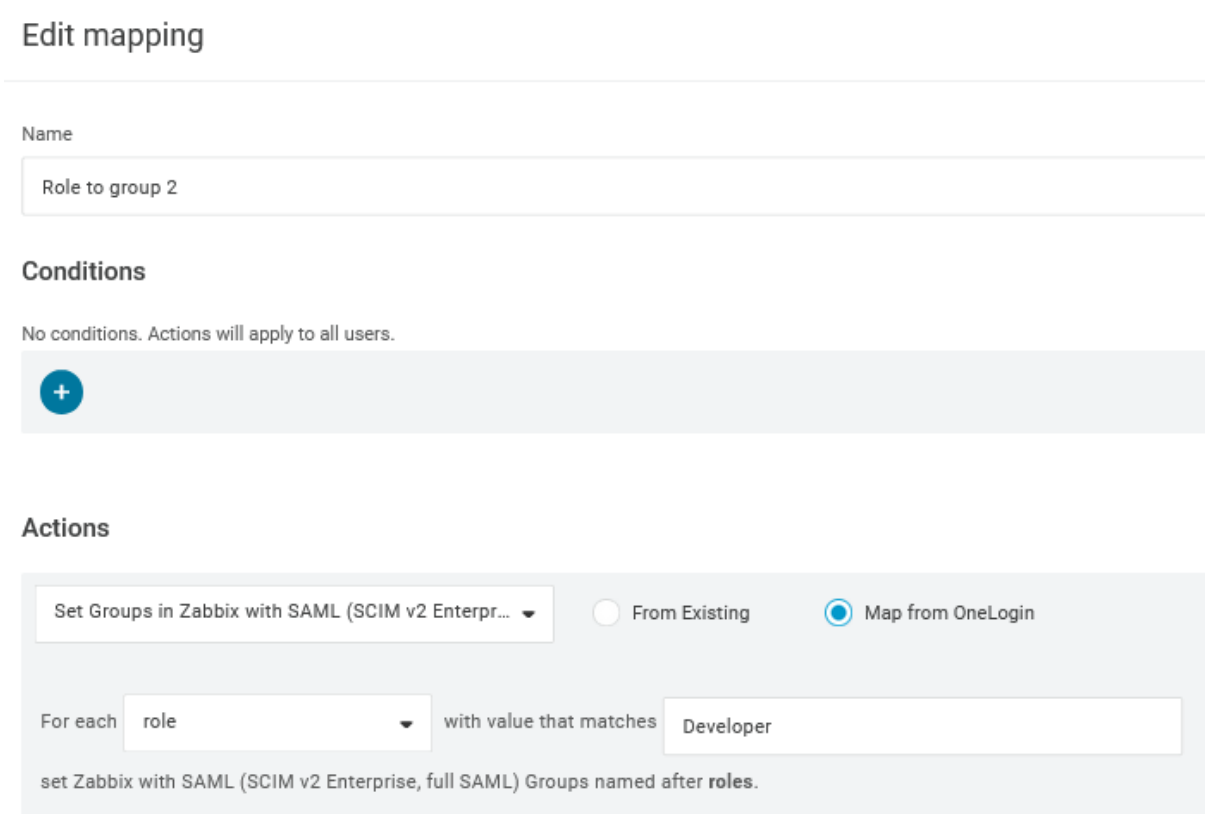
- Ajoutez un paramètre « group » correspondant aux rôles des utilisateurs dans OneLogin. Les rôles des utilisateurs seront transmis sous forme de chaîne, séparés par un point-virgule `;`. Les rôles des utilisateurs OneLogin seront utilisés pour créer des groupes d'utilisateurs dans Zabbix :



Vérifiez la liste des paramètres :



5. Dans la page *Rules*, créez des mappages de rôles d'utilisateurs vers le paramètre Groups par défaut.



Vous pouvez utiliser une expression régulière pour transmettre des rôles spécifiques en tant que groupes. Les noms de rôles ne doivent pas contenir `;`, car OneLogin l'utilise comme séparateur lors de l'envoi d'un attribut comportant plusieurs rôles.

6. Téléchargez le certificat IdP. Définissez ses permissions sur 644 en exécutant :

```
chmod 644 idp.crt
```

Configuration de Zabbix

1. Dans Zabbix, accédez aux paramètres SAML et renseignez les options de configuration en fonction de la configuration OneLogin :



Champ Zabbix	Champ de configuration dans OneLogin	Exemple de valeur
ID d'entité IdP	URL de l'émetteur (voir l'onglet SSO de votre application dans OneLogin)	

Champ Zabbix	Champ de configuration dans OneLogin	Exemple de valeur
URL du service SSO	Point de terminaison SAML 2.0 (HTTP) (voir l'onglet SSO de votre application dans OneLogin)	
URL du service SLO	Point de terminaison SLO (HTTP) (voir l'onglet SSO de votre application dans OneLogin)	
Attribut du nom d'utilisateur	Paramètre personnalisé	`user_email`
Attribut du nom du groupe	Paramètre personnalisé	`group`
Attribut du prénom de l'utilisateur	Paramètre personnalisé	`user_name`
Attribut du nom de famille de l'utilisateur	Paramètre personnalisé	`user_lastname`

Il est également nécessaire de configurer le mappage des groupes d'utilisateurs. Le mappage des médias est facultatif. Cliquez sur *Mettre à jour* pour enregistrer ces paramètres.

2. Ajoutez le certificat Base64 fourni par OneLogin.

Si `$SSO['CERT_STORAGE'] = 'database'` est défini dans le fichier de configuration de l'interface (`zabbix.conf.php`), vous pouvez coller le texte du certificat ou charger le fichier de certificat dans l'interface pendant la configuration SAML : aucun fichier n'est requis sur le système de fichiers.

Si `$SSO['CERT_STORAGE'] = 'file'` est défini dans le fichier de configuration, le certificat doit être disponible sur le système de fichiers (par défaut dans `ui/conf/certs` ou dans le chemin configuré dans le fichier de configuration) et l'interface n'enregistrera pas les certificats dans la base de données. Notez que si `$SSO['CERT_STORAGE']` n'est pas défini ou est commenté, le stockage dans un fichier est utilisé par défaut et les certificats sont lus depuis `ui/conf/certs`.

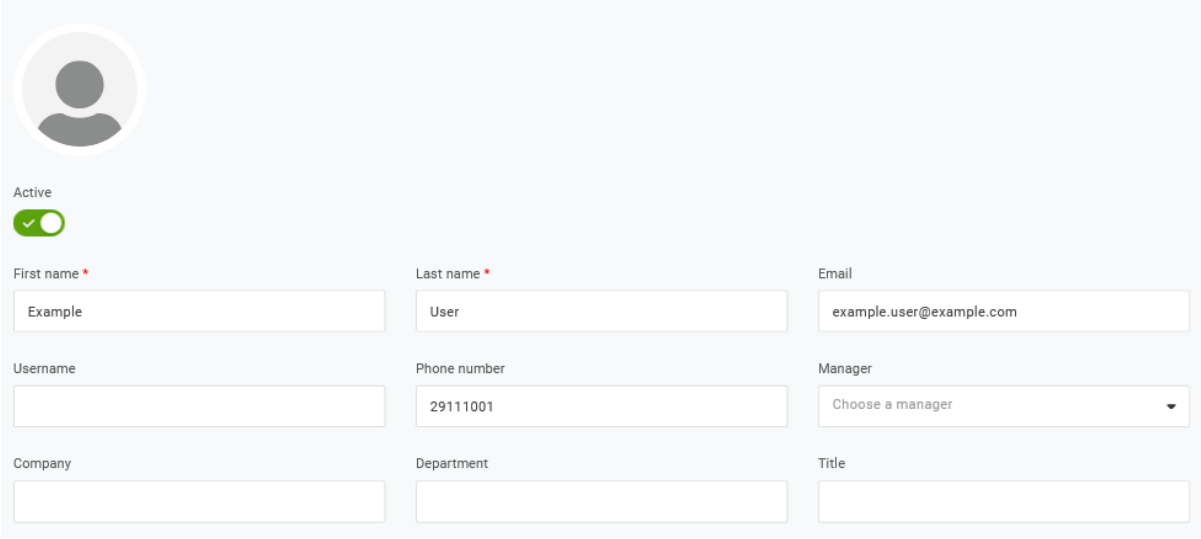
Vous pouvez accéder au téléchargement du certificat dans OneLogin en sélectionnant *Applications* > *SSO*, puis en cliquant sur *View details* sous le certificat actuel.

3. Cliquez sur le bouton *Mettre à jour* pour enregistrer ces paramètres.

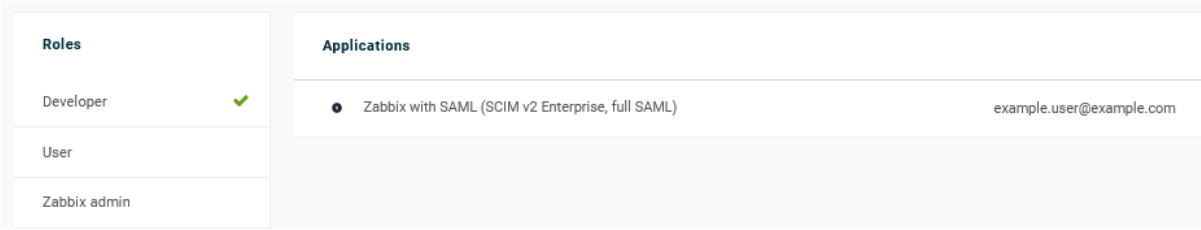
Provisionnement des utilisateurs SCIM

Avec le provisionnement des utilisateurs activé, il est désormais possible d'ajouter ou de mettre à jour des utilisateurs et leurs rôles dans OneLogin, puis de les provisionner immédiatement dans Zabbix.

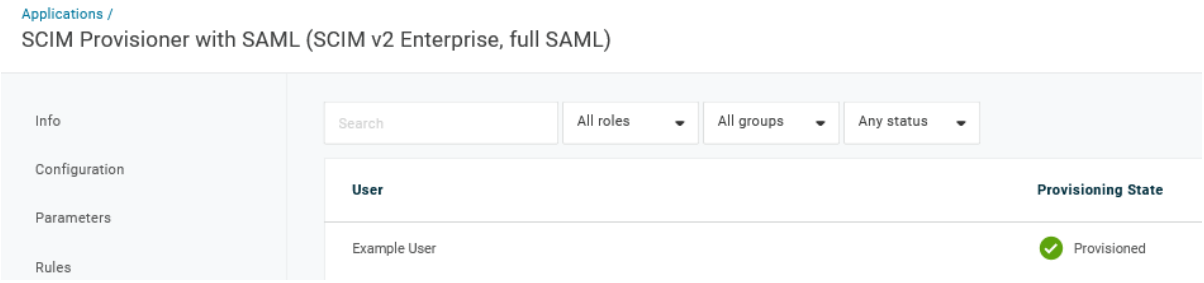
Par exemple, vous pouvez créer un nouvel utilisateur :



Ajoutez-le à un rôle utilisateur et à l'application qui provisionnera l'utilisateur :



Lors de l'enregistrement de l'utilisateur, celui-ci sera provisionné dans Zabbix. Dans *Application* > *Users*, vous pouvez vérifier l'état du provisionnement des utilisateurs actuels de l'application :



S'il est correctement provisionné, l'utilisateur apparaît dans la liste des utilisateurs Zabbix.



Username ▲	Name	Last name	User role	Groups	Is online?	Login	Frontend access
Admin	Zabbix	Administrator	Super admin role	Zabbix administrators	Yes (2023-04-18 21:11:43)	Ok	System default
example.user@example.com	Example	User	Admin role	Zabbix administrators	No	Ok	SAML

Configuration des rapports planifiés

Aperçu

Cette section fournit des instructions sur l'installation du service web Zabbix et la configuration de Zabbix afin d'activer la génération de rapports planifiés.

Installation

Un nouveau processus service web Zabbix et le navigateur Google Chrome doivent être installés pour permettre la génération de rapports planifiés. Le service web peut être installé sur la même machine que celle où le serveur Zabbix est installé ou sur une machine différente. Le navigateur Google Chrome doit être installé sur la même machine que celle où le service web est installé.

Le paquet officiel zabbix-web-service est disponible dans le dépôt Zabbix. Le navigateur Google Chrome n'est pas inclus dans ces paquets et doit être installé séparément.

Pour compiler le service web Zabbix à partir des sources, consultez Installation du service web Zabbix.

Après l'installation, exécutez `zabbix_web_service` sur la machine où le service web est installé :

```
zabbix_web_service
```

Configuration

Pour garantir une communication correcte entre tous les éléments impliqués, assurez-vous que les paramètres de configuration du serveur et de configuration de l'interface sont correctement configurés.

Serveur Zabbix

Les paramètres suivants du fichier de configuration du serveur Zabbix doivent être mis à jour : `WebServiceURL` et `StartReportWriters`.

WebServiceURL

Ce paramètre est requis pour activer la communication avec le service web. L'URL doit respecter le format `http[s]://host:port/report`.

- Par défaut, le service web écoute sur le port 10053. Un autre port peut être spécifié dans le fichier de configuration du service web.
- La spécification du chemin `/report` est obligatoire (ce chemin est codé en dur et ne peut pas être modifié).

Exemple :

```
WebServiceURL=http://localhost:10053/report
```

> **Attention:**
> Il est fortement recommandé de configurer le chiffrement entre le serveur Zabbix et le service web Zabbix à l'aide de certificats. Par défaut, les données transmises entre le serveur Zabbix et le service web Zabbix ne sont pas chiffrées, ce qui peut entraîner un accès non autorisé.

StartReportWriters

Ce paramètre détermine le nombre de processus de génération de rapports à démarrer. S'il n'est pas défini ou s'il est égal à 0, la génération de rapports est désactivée. Selon le nombre et la fréquence des rapports requis, il est possible d'activer de 1 à 100 processus de génération de rapports.

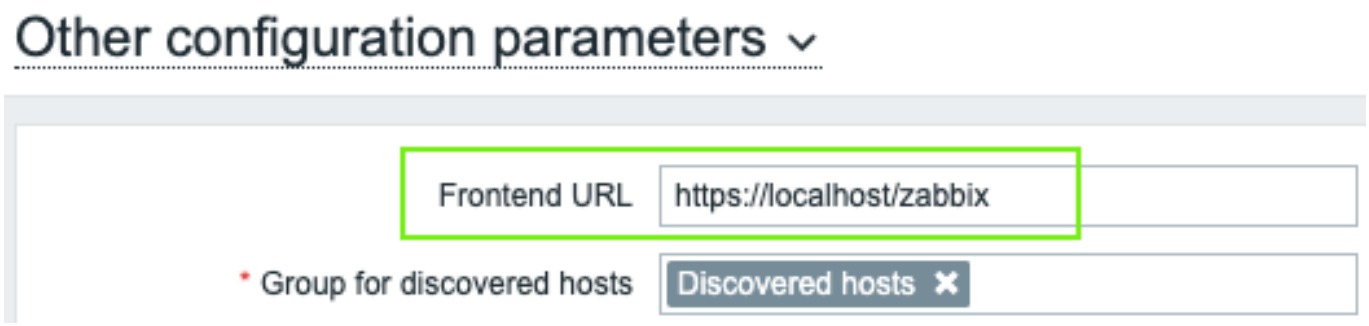
Exemple :

```
StartReportWriters=3
```

Interface web Zabbix

Un paramètre *Frontend URL* doit être défini pour permettre la communication entre l'interface web Zabbix et le service web Zabbix :

- Accédez à la section du menu de l'interface web *Administration > General > Other*
- Indiquez l'URL complète de l'interface web Zabbix dans le paramètre *Frontend URL*



Note:
Une fois la procédure de configuration terminée, vous pouvez configurer et envoyer un rapport de test afin de vous assurer que tout fonctionne correctement.

Éléments

Cette section contient des informations de référence et des notes techniques sur les éléments, y compris des détails sur les paramètres pour des clés d'élément spécifiques, des détails propres à certaines plateformes, ainsi que des notes sur des configurations spéciales d'éléments.

Veuillez utiliser la barre latérale pour trouver les informations pertinentes pour votre cas d'utilisation.

Paramètres de vm.memory.size

Aperçu

Cette section fournit des détails sur certains paramètres de l'élément d'agent vm.memory.size[<mode>].

Paramètres

Les paramètres suivants sont disponibles pour cet élément :

- **active** - mémoire actuellement utilisée ou utilisée très récemment, et donc présente en RAM
- **anon** - mémoire non associée à un fichier (ne peut pas être relue depuis celui-ci)
- **available** - mémoire disponible, calculée différemment selon la plateforme (voir le tableau ci-dessous)
- **buffers** - cache pour des éléments tels que les métadonnées du système de fichiers
- **cached** - cache pour divers éléments
- **exec** - code exécutable, généralement issu d'un fichier (programme)
- **file** - cache pour le contenu des fichiers récemment consultés
- **free** - mémoire immédiatement disponible pour toute entité demandant de la mémoire
- **inactive** - mémoire marquée comme non utilisée
- **pavailable** - mémoire « available » en pourcentage de « total » (calculée comme `available/total*100`)
- **pinned** - identique à « wired »
- **pused** - mémoire « used » en pourcentage de « total » (calculée comme `used/total*100`)
- **shared** - mémoire pouvant être accédée simultanément par plusieurs processus
- **slab** - quantité totale de mémoire utilisée par le noyau pour mettre en cache des structures de données pour son propre usage

- **total** - mémoire physique totale disponible
- **used** - mémoire utilisée, calculée différemment selon la plateforme (voir le tableau ci-dessous)
- **wired** - mémoire marquée pour rester en permanence en RAM. Elle n'est jamais déplacée sur le disque.

Warning:
Certains de ces paramètres sont spécifiques à la plateforme et peuvent ne pas être disponibles sur votre plateforme. Voir Zabbix agent items pour plus de détails.

Calcul spécifique à la plateforme de **available** et **used** :

Platform	**"available"**	**"used"**
AIX	free + cached	mémoire réelle utilisée
FreeBSD	inactive + cached + free	active + wired + cached
HP UX	free	total - free
Linux<3.14	free + buffers + cached	total - free
Linux 3.14+ (également rétroporté vers 3.10 sur RHEL 7)	/proc/meminfo, voir « MemAvailable » dans la documentation du noyau Linux pour plus de détails. Notez que free + buffers + cached n'est plus égal à « available », car tout le cache de pages ne peut pas être libéré et qu'un seuil bas est utilisé dans le calcul.	total - free
NetBSD	inactive + execpages + file + free	total - free
OpenBSD	inactive + free + cached	active + wired
OSX	inactive + free	active + wired
Solaris	free	total - free
Win32	free	total - free

Attention:
La somme de *vm.memory.size[used]* et *vm.memory.size[available]* n'est pas nécessairement égale au total. Par exemple, sur FreeBSD :
 * Les mémoires active, inactive, wired et cached sont considérées comme utilisées, car elles stockent des informations utiles.
 * En même temps, les mémoires inactive, cached et free sont considérées comme disponibles, car ces types de mémoire peuvent être instantanément attribués aux processus qui demandent davantage de mémoire. Ainsi, la mémoire inactive est à la fois utilisée et disponible simultanément. Pour cette raison, l'élément *vm.memory.size[used]* est conçu uniquement à des fins d'information, tandis que *vm.memory.size[available]* est conçu pour être utilisé dans des déclencheurs.

Voir également

1. Détails supplémentaires sur le calcul de la mémoire dans différents systèmes d'exploitation

Niveau d'autorisation minimum pour les éléments de l'agent Windows

Aperçu

Lors de la surveillance de systèmes à l'aide d'un agent, une bonne pratique consiste à obtenir des métriques de l'hôte sur lequel l'agent est installé. Pour utiliser le principe du moindre privilège, il est nécessaire de déterminer quelles métriques sont obtenues de l'agent.

Le tableau de ce document vous permet de sélectionner les droits minimaux pour garantir le bon fonctionnement de l'agent Zabbix.

Si un autre utilisateur est sélectionné pour que l'agent fonctionne, plutôt que "LocalSystem", alors pour le fonctionnement de l'agent en tant que service Windows, le nouvel utilisateur doit avoir les droits "Se connecter en tant que service" de "Local Policy→User Rights Assignment" et le droit de créer, d'écrire et de supprimer le fichier journal de l'agent Zabbix. Un utilisateur Active Directory doit être ajouté au groupe *Performance Monitor Users*.

Note:
Lorsque vous travaillez avec les droits d'un agent basés sur le groupe "minimum techniquement acceptable", la fourniture préalable de droits sur les objets à surveiller est requise.

Éléments agent courants pris en charge sous Windows

Clé d'élément	Groupe d'utilisateurs	
	Recommandé	Minimum techniquement acceptable (les fonctionnalités sont limitées)
agent.hostname	Invités	Invités
agent.ping	Invités	Invités
agent.variant	Invités	Invités
agent.version	Invités	Invités
log	Administrateurs	Invités
log.count	Administrateurs	Invités
logrt	Administrateurs	Invités
logrt.count	Administrateurs	Invités
net.dns	Invités	Invités
net.dns.perf	Invités	Invités
net.dns.record	Invités	Invités
net.if.discovery	Invités	Invités
net.if.in	Invités	Invités
net.if.out	Invités	Invités
net.if.total	Invités	Invités
net.tcp.listen	Invités	Invités
net.tcp.port	Invités	Invités
net.tcp.service	Invités	Invités
net.tcp.service.perf	Invités	Invités
net.udp.service	Invités	Invités
net.udp.service.perf	Invités	Invités
proc.num	Administrateurs	Invités
system.cpu.discovery	Utilisateurs du Moniteur de performances	Utilisateurs du Moniteur de performances
system.cpu.load	Utilisateurs du Moniteur de performances	Utilisateurs du Moniteur de performances
system.cpu.num	Invités	Invités
system.cpu.util	Utilisateurs du Moniteur de performances	Utilisateurs du Moniteur de performances
system.hostname	Invités	Invités
system.localtime	Invités	Invités
system.run	Administrateurs	Invités
system.sw.arch	Invités	Invités
system.swap.size	Invités	Invités
system.uname	Invités	Invités
system.uptime	Utilisateurs du Moniteur de performances	Utilisateurs du Moniteur de performances
vfs.dir.count	Administrateurs	Invités
vfs.dir.get	Administrateurs	Invités
vfs.dir.size	Administrateurs	Invités
vfs.file.cksum	Administrateurs	Invités
vfs.file.contents	Administrateurs	Invités
vfs.file.exists	Administrateurs	Invités
vfs.file.md5sum	Administrateurs	Invités
vfs.file.regexp	Administrateurs	Invités
vfs.file.regmatch	Administrateurs	Invités
vfs.file.size	Administrateurs	Invités
vfs.file.time	Administrateurs	Invités
vfs.fs.discovery	Administrateurs	Invités
vfs.fs.size	Administrateurs	Invités
vfs.fs.get	Administrateurs	Invités
vm.memory.size	Invités	Invités
web.page.get	Invités	Invités
web.page.perf	Invités	Invités
web.page.regexp	Invités	Invités
zabbix.stats	Invités	Invités

Clés d'élément spécifiques à Windows

Clé d'élément	Groupe d'utilisateurs	
	Recommandé	Minimum techniquement acceptable (la fonctionnalité est limitée)
eventlog	Lecteurs du journal des événements	Invités
net.if.list	Invités	Invités
perf_counter	Utilisateurs du Moniteur de performances	Utilisateurs du Moniteur de performances
proc_info	Administrateurs	Invités
service.discovery	Invités	Invités
service.info	Invités	Invités
services	Invités	Invités
wmi.get	Administrateurs	Invités
vm.vmemory.size	Invités	Invités

Encodage des valeurs retournées

Le serveur Zabbix attend que chaque valeur textuelle retournée soit encodée en UTF8. Cela s'applique à tout type de vérifications : agent Zabbix, SSH, Telnet, etc.

Différents systèmes/périphériques surveillés et différentes vérifications peuvent renvoyer des caractères non ASCII dans la valeur. Dans de tels cas, presque toutes les clés **zabbix** possibles contiennent un paramètre de clé d'élément supplémentaire - **<encoding>**. Ce paramètre de clé est facultatif, mais il doit être spécifié si la valeur retournée n'est pas encodée en UTF8 et contient des caractères non ASCII. Sinon, le résultat peut être inattendu et imprévisible.

Une description du comportement avec différents backends de base de données dans de tels cas suit.

MySQL

Si une valeur contient un caractère non ASCII dans un encodage autre que UTF8, ce caractère et les suivants seront ignorés lorsque la base de données stockera cette valeur. Aucun message d'avertissement ne sera écrit dans le fichier `zabbix_server.log`.
 Concerne au moins la version 5.1.61 de MySQL.

PostgreSQL

Si une valeur contient un caractère non ASCII dans un encodage non UTF8, cela entraînera l'échec de la requête SQL (PGRES_FATAL_ERROR:ERROR invalid byte sequence for encoding) et les données ne seront pas stockées. Un message d'avertissement approprié sera écrit dans le fichier `zabbix_server.log`.
 Applicable au moins à la version 9.1.3 de PostgreSQL.

Prise en charge des fichiers volumineux

La prise en charge des fichiers volumineux, souvent abrégée en LFS, désigne la capacité à travailler avec des fichiers de plus de 2 GB sur des systèmes d'exploitation 32 bits. La prise en charge des fichiers volumineux affecte au moins la surveillance des fichiers journaux et tous les éléments vfs.file.*. La prise en charge des fichiers volumineux dépend des capacités du système au moment de la compilation de Zabbix, mais elle est complètement désactivée sur Solaris 32 bits en raison de son incompatibilité avec procfs et swapctl.

Capteur

Vue d'ensemble

Chaque puce de capteur possède son propre répertoire dans l'arborescence sysfs `/sys/devices`. Pour trouver toutes les puces de capteur, il est plus facile de suivre les liens symboliques des périphériques depuis `/sys/class/hwmon/hwmon*`, où `*` est un nombre réel (0,1,2,...).

Les relevés des capteurs se trouvent soit dans le répertoire `/sys/class/hwmon/hwmon*/` pour les périphériques virtuels, soit dans le répertoire `/sys/class/hwmon/hwmon*/device` pour les périphériques non virtuels. Un fichier nommé `name`, situé dans les répertoires `hwmon*` ou `hwmon*/device`, contient le nom de la puce, qui correspond au nom du pilote du noyau utilisé par la puce de capteur.

Il n'y a qu'une seule valeur de relevé de capteur par fichier. Le schéma commun de nommage des fichiers contenant les relevés des capteurs dans l'un des répertoires mentionnés ci-dessus est : `<type><number>_<item>`, où

- **type** - pour les puces de capteur, il s'agit de "in" (tension), "temp" (température), "fan" (ventilateur), etc.
- **item** - "input" (valeur mesurée), "max" (seuil haut), "min" (seuil bas), etc.

- **number** - toujours utilisé pour les éléments pouvant être présents plus d'une fois (commence généralement à 1, sauf pour les tensions qui commencent à 0). Si les fichiers ne se rapportent pas à un élément spécifique, ils portent un nom simple sans numéro.

Les informations concernant les capteurs disponibles sur l'hôte peuvent être obtenues à l'aide des outils **sensors-detect** et **sensors** (paquet lm-sensors). **Sensors-detect** aide à déterminer quels modules sont nécessaires pour les capteurs disponibles. Lorsque les modules sont chargés, le programme **sensors** peut être utilisé pour afficher les relevés de toutes les puces de capteur. L'étiquetage des relevés de capteurs utilisé par ce programme peut être différent du schéma de nommage commun (`<type><number>_<item>`) :

- s'il existe un fichier nommé `<type><number>_label`, alors l'étiquette contenue dans ce fichier sera utilisée à la place du nom `<type><number><item>` ;
- s'il n'existe pas de fichier `<type><number>_label`, alors le programme recherche dans `/etc/sensors.conf` (cela peut aussi être `/etc/sensors3.conf`, ou un autre fichier) une substitution de nom.

Cet étiquetage permet à l'utilisateur de déterminer quel type de matériel est utilisé. S'il n'existe ni fichier `<type><number>_label` ni étiquette dans le fichier de configuration, le type de matériel peut être déterminé par l'attribut name (`hwmon*/device/name`). Les noms réels des capteurs acceptés par `zabbix_agent` peuvent être obtenus en exécutant le programme **sensors** avec le paramètre `-u` (`sensors -u`).

Dans le programme **sensors**, les capteurs disponibles sont séparés selon le type de bus (adaptateur ISA, adaptateur PCI, adaptateur SPI, périphérique virtuel, interface ACPI, adaptateur HID).

Obtention des identifiants de capteur

Les libellés des capteurs, tels qu'ils sont affichés par la commande `sensors`, ne peuvent pas toujours être utilisés directement, car l'étiquetage varie selon les fabricants de puces. Par exemple, la sortie de `sensors` peut contenir les lignes suivantes :

```
sensors
in0:          +2.24 V  (min =  +0.00 V, max =  +3.32 V)
Vcore:        +1.15 V  (min =  +0.00 V, max =  +2.99 V)
+3.3V:        +3.30 V  (min =  +2.97 V, max =  +3.63 V)
+12V:        +13.00 V  (min =  +0.00 V, max = +15.94 V)
M/B Temp:     +30.0°C  (low  = -127.0°C, high = +127.0°C)
```

Parmi ceux-ci, un seul nom affiché correspond directement à un identifiant interne :

```
zabbix_get -s 127.0.0.1 -k sensor[lm85-i2c-0-2e,in0]
2.240000
```

Toute tentative d'utiliser d'autres libellés affichés (comme `Vcore` ou `+12V`) ne fonctionnera pas :

```
zabbix_get -s 127.0.0.1 -k sensor[lm85-i2c-0-2e,Vcore]
ZBX_NOTSUPPORTED
```

Pour trouver l'identifiant interne du capteur accepté par Zabbix, exécutez `sensors -u`. Dans la sortie, on peut observer ce qui suit :

```
sensors -u
...
Vcore:
  in1_input: 1.15
  in1_min: 0.00
  in1_max: 2.99
  in1_alarm: 0.00
...
+12V:
  in4_input: 13.00
  in4_min: 0.00
  in4_max: 15.94
  in4_alarm: 0.00
...
```

Ainsi, le libellé `Vcore` correspond à l'identifiant `in1`, et `+12V` correspond à `in4`. Selon la spécification hwmon sysfs du noyau, il s'agit de tensions sur les broches de la puce et, de manière générale, elles peuvent nécessiter une mise à l'échelle.

```
zabbix_get -s 127.0.0.1 -k sensor[lm85-i2c-0-2e,in1]
1.301000
```

Zabbix peut récupérer non seulement les mesures de tension (`in`), mais aussi de courant (`curr`), de température (`temp`) et de vitesse de ventilateur (`fan`).

Compatibilité

Anciennes distributions

Les relevés des capteurs sont obtenus à partir du répertoire `/proc/sys/dev/sensors` :

- **device** - nom du périphérique (si <mode> est utilisé, il s'agit d'une expression régulière)
- **sensor** - nom du capteur (si <mode> est utilisé, il s'agit d'une expression régulière)
- **mode** - valeurs possibles : `avg`, `max`, `min` (si ce paramètre est omis, device et sensor sont traités littéralement)

Exemple de clé : `sensor[w83781d-i2c-0-2d,temp1]`.

Distributions actuelles

Les relevés des capteurs sont obtenus à partir du répertoire `/sys/class/hwmon` :

- **device** - nom du périphérique (expression non régulière). Le nom du périphérique peut être le nom réel du périphérique (par ex. `0000:00:18.3`) ou le nom obtenu à l'aide du programme sensors (par ex. `k8temp-pci-00c3`). C'est à l'utilisateur de choisir quel nom utiliser.
- **sensor** - nom du capteur (expression non régulière).
- **mode** - valeurs possibles : `avg`, `max`, `min` (si ce paramètre est omis, device et sensor sont traités littéralement).

Exemples de clés :

`sensor[k8temp-pci-00c3,temp,max]` ou `sensor[0000:00:18.3,temp1]`.

`sensor[smsc47b397-isa-0880,in,avg]` ou `sensor[smsc47b397.2176,in1]`.

Notes sur le paramètre memtype dans les éléments proc.mem

Vue d'ensemble

Le paramètre **memtype** est pris en charge sur les plateformes Linux, AIX, FreeBSD et Solaris.

Trois valeurs courantes de `memtype` sont prises en charge sur toutes ces plateformes : `pmem`, `rss` et `vsize`. En outre, des valeurs de `memtype` spécifiques à certaines plateformes sont prises en charge sur certaines d'entre elles.

AIX

Consultez dans le tableau les valeurs prises en charge pour le paramètre `memtype` sur AIX.

Valeur prise en charge	Description	Source dans la structure procentry64	Essaie d'être compatible avec
`vsize` [1]	Taille de la mémoire virtuelle	`pi_size`	
`pmem`	Pourcentage de mémoire réelle	`pi_prm`	`ps -o pmem`
`rss`	Taille de l'ensemble résident	`pi_trss + pi_drss`	`ps -o rssize`
`size`	Taille du processus (code + données)	`pi_dvm`	Colonne SIZE de `ps gvw`
`dsize`	Taille des données	`pi_dsize`	
`tsize`	Taille du texte (code)	`pi_tsize`	Colonne TSIZ de `ps gvw`
`sdsize`	Taille des données de la bibliothèque partagée	`pi_sdsize`	
`drss`	Taille des données résidentes	`pi_drss`	
`trss`	Taille du texte résident	`pi_trss`	

Notes pour AIX :

1. Lors du choix des paramètres de la clé d'élément `proc.mem[]` sur AIX, essayez de spécifier des critères de sélection de processus précis. Sinon, vous risquez d'obtenir des processus indésirables comptabilisés dans le résultat de `proc.mem[]`.

Exemple :

```
$ zabbix_agentd -t proc.mem[,,,NonExistingProcess,rss]
proc.mem[,,,NonExistingProcess,rss]           [u|2879488]
```

Cet exemple montre que le fait de spécifier uniquement le paramètre de ligne de commande (expression régulière à rechercher) entraîne la prise en compte de l'agent Zabbix lui-même, ce qui n'est probablement pas le résultat souhaité.

2. N'utilisez pas `ps -ef` pour parcourir les processus : cette commande affiche uniquement les processus qui ne sont pas liés au noyau. Utilisez `ps -Af` pour voir tous les processus qui seront visibles par l'agent Zabbix.

3. Examinons, à l'aide de l'exemple de `topasrec`, comment l'agent Zabbix sélectionne les processus avec `proc.mem[]`.

```
$ ps -Af | grep topasrec
root 10747984        1   0   Mar 16      -  0:00 /usr/bin/topasrec  -L -s 300 -R 1 -r 6 -o /var/perf daily
```

`proc.mem[]` possède les arguments suivants :

`proc.mem[<name>,<user>,<mode>,<cmdline>,<memtype>]`

où : - Le premier critère est un nom de processus (argument `<name>`). Dans notre exemple, l'agent Zabbix le verra comme `topasrec`. Pour obtenir une correspondance, vous devez spécifier `topasrec` ou le laisser vide. - Le deuxième critère est un nom d'utilisateur (argument `<user>`). Pour obtenir une correspondance, vous devez spécifier `root` ou le laisser vide. - Le troisième critère utilisé pour sélectionner le processus est un argument `<cmdline>`. L'agent Zabbix verra sa valeur comme `/usr/bin/topasrec -L -s 300 -R 1 -r 6 -o /var/perf/daily/ -ypersistent=1 -O type=bin -ystart_time=04:08:54,Mar16,2023`. Pour obtenir une correspondance, vous devez spécifier une expression régulière correspondant à cette chaîne ou la laisser vide.

Les arguments `<mode>` et `<memtype>` sont appliqués après l'utilisation des trois critères mentionnés ci-dessus.

FreeBSD

Consultez dans le tableau les valeurs prises en charge pour le paramètre `memtype` sur FreeBSD.

Valeur prise en charge	Description	Source dans la structure kinfo_proc	Tentative de compatibilité avec
`vsize`	Taille de la mémoire virtuelle	`kp_eproc.e_vm.vm_map.size` ou `ki_size`	`ps -o vsz`
`pmem`	Pourcentage de la mémoire réelle	calculé à partir de `rss`	`ps -o pmem`
`rss`	Taille de l'ensemble résident	`kp_eproc.e_vm.vm_rssize` ou `ki_rssize`	`ps -o rss`
`size` [1]	Taille du processus (code + données + pile)	`tsize + dsize + ssize`	
`tsize`	Taille du texte (code)	`kp_eproc.e_vm.vm_tsize` ou `ki_tsize`	`ps -o tsiz`
`dsize`	Taille des données	`kp_eproc.e_vm.vm_dsize` ou `ki_dsize`	`ps -o dsiz`
`ssize`	Taille de la pile	`kp_eproc.e_vm.vm_ssize` ou `ki_ssize`	`ps -o ssiz`

Linux

Consultez dans le tableau les valeurs prises en charge pour le paramètre `memtype` sous Linux.

Valeur prise en charge	Description	Source dans le fichier /proc/<pid>/status
`vsize` [1]	Taille de la mémoire virtuelle	`VmSize`
`pmem`	Pourcentage de la mémoire réelle	`(VmRSS/total_memory) * 100`
`rss`	Taille de l'ensemble résident	`VmRSS`
`data`	Taille du segment de données	`VmData`
`exe`	Taille du segment de code	`VmExe`
`hwm`	Taille maximale de l'ensemble résident	`VmHWM`
`lck`	Taille de la mémoire verrouillée	`VmLck`
`lib`	Taille des bibliothèques partagées	`VmLib`
`peak`	Taille maximale de la mémoire virtuelle	`VmPeak`
`pin`	Taille des pages épinglées	`VmPin`
`pte`	Taille des entrées de table des pages	`VmPTE`
`size`	Taille des segments de code + données + pile du processus	`VmExe + VmData + VmStk`
`stk`	Taille du segment de pile	`VmStk`

Valeur prise en charge	Description	Source dans le fichier /proc/<pid>/status
swap	Taille de l'espace d'échange utilisé	VmSwap

Remarques pour Linux :

1. Toutes les valeurs memtype ne sont pas prises en charge par les anciens noyaux Linux. Par exemple, les noyaux Linux 2.4 ne prennent pas en charge les valeurs hwm, pin, peak, pte et swap.
2. Nous avons constaté que l'autosurveillance du processus de vérification active de l'agent Zabbix avec proc.mem[...,...,...,...,c affiche une valeur supérieure de 4 kB à celle indiquée par la ligne VmData dans le fichier /proc/<pid>/status de l'agent. Au moment de l'automesure, le segment de données de l'agent augmente de 4 kB, puis revient à sa taille précédente.

Solaris

Consultez dans le tableau les valeurs prises en charge pour le paramètre memtype sur Solaris.

Valeur prise en charge	Description	Source dans la structure psinfo	Tente d'être compatible avec
vsize [1]	Taille de l'image du processus	pr_size	ps -o vsz
pmem	Pourcentage de mémoire réelle	pr_pctmem	ps -o pmem
rss	Taille de l'ensemble résident Elle peut être sous-estimée (voir la description de rss dans man ps).	pr_rssize	ps -o rss

Notes de bas de page

[1] Valeur par défaut.

Notes sur la sélection des processus dans les éléments proc.mem et proc.num

Processus modifiant leur ligne de commande

Certains programmes utilisent la modification de leur ligne de commande comme méthode d'affichage de leur activité en cours. Un utilisateur peut voir cette activité en exécutant les commandes ps et top. Parmi ces programmes figurent *PostgreSQL*, *Sendmail*, *Zabbix*.

Voyons un exemple sous Linux. Supposons que nous voulions surveiller un certain nombre de processus d'agent Zabbix.

La commande ps affiche les processus concernés comme suit :

```
$ ps -fu zabbix
UID        PID  PPID  C STIME TTY          TIME CMD
...
zabbix    6318     1  0 12:01 ?        00:00:00 sbin/zabbix_agentd -c /home/zabbix/zabbix_agentd.conf
zabbix    6319  6318  0 12:01 ?        00:00:01 sbin/zabbix_agentd: collector [idle 1 sec]
zabbix    6320  6318  0 12:01 ?        00:00:00 sbin/zabbix_agentd: listener #1 [waiting for connection]
zabbix    6321  6318  0 12:01 ?        00:00:00 sbin/zabbix_agentd: listener #2 [waiting for connection]
zabbix    6322  6318  0 12:01 ?        00:00:00 sbin/zabbix_agentd: listener #3 [waiting for connection]
zabbix    6323  6318  0 12:01 ?        00:00:00 sbin/zabbix_agentd: active checks #1 [idle 1 sec]
...
```

La sélection des processus par nom et par utilisateur fonctionne :

```
$ zabbix_get -s localhost -k 'proc.num[zabbix_agentd,zabbix]'
6
```

Renommons maintenant l'exécutable zabbix_agentd en zabbix_agentd_30 et redémarrons-le.

ps affiche maintenant :

```
$ ps -fu zabbix
UID        PID  PPID  C STIME TTY          TIME CMD
...
zabbix    6715     1  0 12:53 ?        00:00:00 sbin/zabbix_agentd_30 -c /home/zabbix/zabbix_agentd.conf
zabbix    6716  6715  0 12:53 ?        00:00:00 sbin/zabbix_agentd_30: collector [idle 1 sec]
zabbix    6717  6715  0 12:53 ?        00:00:00 sbin/zabbix_agentd_30: listener #1 [waiting for connection
```

```
zabbix    6718  6715  0 12:53 ?        00:00:00 sbin/zabbix_agentd_30: listener #2 [waiting for connection
zabbix    6719  6715  0 12:53 ?        00:00:00 sbin/zabbix_agentd_30: listener #3 [waiting for connection
zabbix    6720  6715  0 12:53 ?        00:00:00 sbin/zabbix_agentd_30: active checks #1 [idle 1 sec]
...
```

À présent, la sélection des processus par nom et par utilisateur produit un résultat incorrect :

```
$ zabbix_get -s localhost -k 'proc.num[zabbix_agentd_30,zabbix]'
1
```

Pourquoi un simple renommage de l'exécutable avec un nom plus long conduit-il à un résultat si différent ?

L'agent Zabbix commence par vérifier le nom du processus. Le fichier `/proc/<pid>/status` est ouvert et la ligne `Name` est vérifiée. Dans notre cas, les lignes `Name` sont :

```
$ grep Name /proc/{6715,6716,6717,6718,6719,6720}/status
/proc/6715/status:Name:   zabbix_agentd_3
/proc/6716/status:Name:   zabbix_agentd_3
/proc/6717/status:Name:   zabbix_agentd_3
/proc/6718/status:Name:   zabbix_agentd_3
/proc/6719/status:Name:   zabbix_agentd_3
/proc/6720/status:Name:   zabbix_agentd_3
```

Le nom du processus dans le fichier `status` est tronqué à 15 caractères.

Un résultat similaire peut être observé avec la commande `ps` :

```
$ ps -u zabbix
  PID TTY          TIME CMD
...
 6715 ?        00:00:00 zabbix_agentd_3
 6716 ?        00:00:01 zabbix_agentd_3
 6717 ?        00:00:00 zabbix_agentd_3
 6718 ?        00:00:00 zabbix_agentd_3
 6719 ?        00:00:00 zabbix_agentd_3
 6720 ?        00:00:00 zabbix_agentd_3
 ...
```

Évidemment, cela n'est pas égal à la valeur du paramètre `name` de `proc.num[]`, à savoir `zabbix_agentd_30`. N'ayant pas réussi à faire correspondre le nom du processus à partir du fichier `status`, l'agent Zabbix se rabat sur le fichier `/proc/<pid>/cmdline`.

La façon dont l'agent voit le fichier "cmdline" peut être illustrée en exécutant la commande suivante :

```
$ for i in 6715 6716 6717 6718 6719 6720; do cat /proc/$i/cmdline | awk '{gsub(/\x0/,"<NUL>"); print};'; d
sbin/zabbix_agentd_30<NUL>-c<NUL>/home/zabbix/zabbix_agentd.conf<NUL>
sbin/zabbix_agentd_30: collector [idle 1 sec]<NUL><NUL><NUL><NUL><NUL><NUL><NUL><NUL><NUL><NUL><NUL><NUL><
sbin/zabbix_agentd_30: listener #1 [waiting for connection]<NUL><NUL><NUL><NUL><NUL><NUL><NUL><NUL><NUL><N
sbin/zabbix_agentd_30: listener #2 [waiting for connection]<NUL><NUL><NUL><NUL><NUL><NUL><NUL><NUL><NUL><N
sbin/zabbix_agentd_30: listener #3 [waiting for connection]<NUL><NUL><NUL><NUL><NUL><NUL><NUL><NUL><NUL><N
sbin/zabbix_agentd_30: active checks #1 [idle 1 sec]<NUL><NUL><NUL><NUL><NUL><NUL><NUL><NUL><NUL><NUL><NUL
```

Dans notre cas, les fichiers `/proc/<pid>/cmdline` contiennent des octets nuls invisibles et non imprimables, utilisés pour terminer les chaînes en langage *C*. Dans cet exemple, les octets nuls sont affichés sous la forme "<NUL>".

L'agent Zabbix vérifie "cmdline" pour le processus principal et récupère `zabbix_agentd_30`, ce qui correspond à la valeur de notre paramètre `name`, `zabbix_agentd_30`. Ainsi, le processus principal est compté par l'élément `proc.num[zabbix_agentd_30,zabbix]`.

Lors de la vérification du processus suivant, l'agent récupère `zabbix_agentd_30: collector [idle 1 sec]` à partir du fichier `cmdline`, et cela ne correspond pas à notre paramètre `name` `zabbix_agentd_30`. Ainsi, seul le processus principal, qui ne modifie pas sa ligne de commande, est compté. Les autres processus de l'agent modifient leur ligne de commande et sont ignorés.

Cet exemple montre que le paramètre `name` ne peut pas être utilisé dans `proc.mem[]` et `proc.num[]` pour sélectionner des processus dans ce cas.

> **Note:**
> Pour l'élément `proc.get[]`, lorsque l'agent Zabbix vérifie "cmdline" pour le nom du processus, il n'utilise que la partie du nom commençant après le dernier slash et jusqu'au premier espace ou au signe deux-points. Le nom du processus reçu à partir du fichier cmdline ne sera utilisé que si son début correspond entièrement au nom de processus raccourci dans le fichier `status`. L'algorithme est le même à la fois pour le nom du processus dans le filtre et dans la sortie JSON.

L'utilisation du paramètre cmdline avec une expression régulière appropriée produit un résultat correct :

```
$ zabbix_get -s localhost -k 'proc.num[,zabbix,,zabbix_agentd_30[ :]]'
6
```

Soyez prudent lorsque vous utilisez les éléments proc.get[], proc.mem[] et proc.num[] pour surveiller des programmes qui modifient leurs lignes de commande.

Avant d'utiliser les paramètres name et cmdline dans les éléments proc.get[], proc.mem[] et proc.num[], vous pouvez tester les paramètres à l'aide de l'élément proc.num[] et de la commande ps.

Threads du noyau Linux

Les threads ne peuvent pas être sélectionnés avec le paramètre cmdline dans les éléments proc.get[], proc.mem[] et proc.num[]

Prenons comme exemple l'un des threads du noyau :

```
$ ps -ef| grep kthreadd
root         2     0  0 09:33 ?        00:00:00 [kthreadd]
```

Il peut être sélectionné avec le paramètre name du processus :

```
$ zabbix_get -s localhost -k 'proc.num[kthreadd,root]'
1
```

Mais la sélection par le paramètre cmdline du processus ne fonctionne pas :

```
$ zabbix_get -s localhost -k 'proc.num[,root,,kthreadd]'
0
```

La raison est que l'agent Zabbix prend l'expression régulière spécifiée dans le paramètre cmdline et l'applique au contenu de /proc/<pid>/cmdline du processus. Pour les threads du noyau, leurs fichiers /proc/<pid>/cmdline sont vides. Ainsi, le paramètre cmdline ne correspond jamais.

Comptage des threads dans les éléments proc.mem[] et proc.num[]

Les threads du noyau Linux sont comptés par l'élément proc.num[] mais ne signalent pas la mémoire dans l'élément proc.mem[]. Par exemple :

```
$ ps -ef | grep kthreadd
root         2     0  0 09:51 ?        00:00:00 [kthreadd]

$ zabbix_get -s localhost -k 'proc.num[kthreadd]'
1

$ zabbix_get -s localhost -k 'proc.mem[kthreadd]'
ZBX_NOTSUPPORTED: Cannot get amount of "VmSize" memory.
```

Mais que se passe-t-il si un processus utilisateur porte le même nom qu'un thread du noyau ? Alors cela pourrait ressembler à ceci :

```
$ ps -ef | grep kthreadd
root         2     0  0 09:51 ?        00:00:00 [kthreadd]
zabbix    9611  6133  0 17:58 pts/1    00:00:00 ./kthreadd

$ zabbix_get -s localhost -k 'proc.num[kthreadd]'
2

$ zabbix_get -s localhost -k 'proc.mem[kthreadd]'
4157440
```

proc.num[] compte à la fois le thread du noyau et le processus utilisateur. proc.mem[] rapporte la mémoire pour le processus utilisateur uniquement et compte la mémoire du thread du noyau comme si elle était à 0. Cela est différent du cas ci-dessus lorsque ZBX_NOTSUPPORTED a été signalé.

Soyez prudent lorsque vous utilisez les éléments proc.mem[] et proc.num[] si le nom du programme correspond à l'un des threads.

Avant de mettre des paramètres dans les éléments proc.mem[] et proc.num[], vous pouvez les tester à l'aide de l'élément proc.num[] et de la commande ps.

Détails d'implémentation des vérifications net.tcp.service et net.udp.service

L'implémentation des vérifications net.tcp.service et net.udp.service est détaillée sur cette page pour divers services spécifiés dans le paramètre service.

Paramètres de l'élément net.tcp.service

ftp

Crée une connexion TCP et attend que les 4 premiers caractères de la réponse soient "220 ", puis envoie "QUIT\r\n". Le port 21 par défaut est utilisé s'il n'est pas spécifié.

http

Crée une connexion TCP sans rien attendre ni rien envoyer. Le port par défaut 80 est utilisé s'il n'est pas spécifié.

https

Utilise (et fonctionne uniquement avec) libcurl, ne vérifie pas l'authenticité du certificat, ne vérifie pas le nom d'hôte dans le certificat SSL, récupère uniquement l'en-tête de la réponse (requête HEAD). Le port par défaut 443 est utilisé s'il n'est pas spécifié.

imap

Crée une connexion TCP et s'attend à ce que les 4 premiers caractères de la réponse soient "* OK", puis envoie "a1 LOGOUT\r\n". Le port par défaut 143 est utilisé s'il n'est pas spécifié.

ldap

Ouvre une connexion à un serveur LDAP et effectue une opération de recherche LDAP avec le filtre défini sur (objectClass=*). Attend une récupération réussie du premier attribut de la première entrée. Le port par défaut 389 est utilisé s'il n'est pas spécifié.

nntp

Crée une connexion TCP et s'attend à ce que les 3 premiers caractères de la réponse soient "200" ou "201", puis envoie "QUIT\r\n". Le port par défaut 119 est utilisé s'il n'est pas spécifié.

pop

Crée une connexion TCP et s'attend à ce que les 3 premiers caractères de la réponse soient "+OK", puis envoie "QUIT\r\n". Le port par défaut 110 est utilisé s'il n'est pas spécifié.

smtp

Creates a TCP connection and expects the first 3 characters of the response to be "220", followed by a space, the line ending or a dash. The lines containing a dash belong to a multiline response and the response will be re-read until a line without the dash is received. Then sends "QUIT\r\n". Default port 25 is used if not specified.

ssh

Crée une connexion TCP. Si la connexion a été établie, les deux côtés échangent une chaîne d'identification (SSH-major.minor-XXXX), où major et minor sont des versions de protocole et XXXX est une chaîne. Zabbix vérifie si la chaîne correspondant à la spécification est trouvée, puis renvoie la chaîne "SSH-major.minor-zabbix_agent\r\n" ou "0\n" en cas de non concordance. Le port par défaut 22 est utilisé s'il n'est pas spécifié.

tcp

Crée une connexion TCP sans rien attendre ni rien envoyer. Contrairement aux autres vérifications, le paramètre de port doit être spécifié.

telnet

Crée une connexion TCP et attend une invite de connexion (':' à la fin). Le port par défaut 23 est utilisé s'il n'est pas spécifié.

Paramètres de l'élément net.udp.service

ntp

Envoie un paquet SNTP sur UDP et valide la réponse conformément à la RFC 4330, section 5. Le port par défaut 123 est utilisé s'il n'est pas spécifié.

paramètres de proc.get

Vue d'ensemble

L'élément **proc.get**[<name>,<user>,<cmdline>,<mode>] est pris en charge sous Linux, Windows, FreeBSD, OpenBSD et NetBSD.

La liste des paramètres de processus renvoyés par l'élément varie en fonction du système d'exploitation et de la valeur de l'argument « mode ».

Linux

Les paramètres de processus suivants sont renvoyés sous Linux pour chaque mode :

mode=process	mode=thread	mode=summary
pid : PID	pid : PID	name : nom du processus
ppid : PID du processus parent	ppid : PID du processus parent	processes : nombre de processus
name : nom du processus	name : nom du processus	vsize : taille de la mémoire virtuelle
cmdline : commande avec arguments	user : utilisateur (réel) sous lequel le processus s'exécute	pmem : pourcentage de mémoire réelle
user : utilisateur (réel) sous lequel le processus s'exécute	group : groupe (réel) sous lequel le processus s'exécute	rss : taille de l'ensemble résident
group : groupe (réel) sous lequel le processus s'exécute	uid : ID utilisateur	data : taille du segment de données
uid : ID utilisateur	gid : ID du groupe sous lequel le processus s'exécute	exe : taille du segment de code
gid : ID du groupe sous lequel le processus s'exécute	tid : ID du thread	lib : taille des bibliothèques partagées
vsize : taille de la mémoire virtuelle	tname : nom du thread	lck : taille de la mémoire verrouillée
pmem : pourcentage de mémoire réelle	cputime_user : temps CPU total passé en mode utilisateur (valeur exprimée en tops d'horloge — diviser par `getconf CLK_TCK` ou `sysconf(_SC_CLK_TCK)` pour obtenir les secondes)	pin : taille des pages épinglées
rss : taille de l'ensemble résident	cputime_system : temps CPU total passé en mode système (valeur exprimée en tops d'horloge — diviser par `getconf CLK_TCK` ou `sysconf(_SC_CLK_TCK)` pour obtenir les secondes)	pte : taille des entrées de table des pages
data : taille du segment de données	state : état du thread	size : taille des segments de code + données + pile du processus
exe : taille du segment de code	ctx_switches : nombre de changements de contexte	stk : taille du segment de pile
hwm : taille maximale de l'ensemble résident	page_faults : nombre de défauts de page	swap : taille de l'espace d'échange utilisé
lck : taille de la mémoire verrouillée		cputime_user : temps CPU total passé en mode utilisateur (valeur exprimée en tops d'horloge — diviser par `getconf CLK_TCK` ou `sysconf(_SC_CLK_TCK)` pour obtenir les secondes)
lib : taille des bibliothèques partagées		cputime_system : temps CPU total passé en mode système (valeur exprimée en tops d'horloge — diviser par `getconf CLK_TCK` ou `sysconf(_SC_CLK_TCK)` pour obtenir les secondes)
peak : taille maximale de la mémoire virtuelle		ctx_switches : nombre de changements de contexte
pin : taille des pages épinglées		threads : nombre de threads
pte : taille des entrées de table des pages		page_faults : nombre de défauts de page
size : taille des segments de code + données + pile du processus		pss : taille proportionnelle de l'ensemble mémoire
stk : taille du segment de pile		

mode=process	mode=thread	mode=summary
swap : taille de l'espace d'échange utilisé cputime_user : temps CPU total passé en mode utilisateur (valeur exprimée en tops d'horloge — diviser par `getconf CLK_TCK` ou `sysconf(_SC_CLK_TCK)` pour obtenir les secondes) cputime_system : temps CPU total passé en mode système (valeur exprimée en tops d'horloge — diviser par `getconf CLK_TCK` ou `sysconf(_SC_CLK_TCK)` pour obtenir les secondes) state : état du processus (récupéré de manière transparente depuis procfs, forme longue) ctx_switches : nombre de changements de contexte threads : nombre de threads page_faults : nombre de défauts de page pss : taille proportionnelle de l'ensemble mémoire		

Systèmes d'exploitation de type BSD

Les paramètres de processus suivants sont renvoyés sur FreeBSD, OpenBSD et NetBSD pour chaque mode :

mode=process	mode=thread	mode=summary
pid: PID	pid: PID	name: nom du processus
ppid: PID parent	ppid: PID parent	processes: nombre de processus
jid: ID de la jail (FreeBSD uniquement)	jid: ID de la jail (FreeBSD uniquement)	vsize: taille de la mémoire virtuelle
jname: nom de la jail (FreeBSD uniquement)	jname: nom de la jail (FreeBSD uniquement)	pmem: pourcentage de mémoire réelle (FreeBSD uniquement)
name: nom du processus	name: nom du processus	rss: taille de l'ensemble résident
cmdline: commande avec arguments	user: utilisateur (réel) sous lequel le processus s'exécute	size: taille du processus (code + données + pile)
user: utilisateur (réel) sous lequel le processus s'exécute	group: groupe (réel) sous lequel le processus s'exécute	tsize: taille du texte (code)
group: groupe (réel) sous lequel le processus s'exécute	uid: ID utilisateur	dsize: taille des données
uid: ID utilisateur	gid: ID du groupe sous lequel le processus s'exécute	ssize: taille de la pile
gid: ID du groupe sous lequel le processus s'exécute	tid: ID du thread	cputime_user: temps CPU total passé en mode utilisateur (indiqué en secondes)
vsize: taille de la mémoire virtuelle	tname: nom du thread	cputime_system: temps CPU total passé en mode système (indiqué en secondes)
pmem: pourcentage de mémoire réelle (FreeBSD uniquement)	cputime_user: temps CPU total passé en mode utilisateur (indiqué en secondes)	ctx_switches: nombre de changements de contexte
rss: taille de l'ensemble résident	cputime_system: temps CPU total passé en mode système (indiqué en secondes)	threads: nombre de threads (non pris en charge pour NetBSD)
size: taille du processus (code + données + pile)	state: état du thread	stk: taille du segment de pile
tsize: taille du texte (code)	ctx_switches: nombre de changements de contexte	page_faults: nombre de défauts de page

mode=process	mode=thread	mode=summary
dsize: taille des données	io_read_op: nombre de fois où le système a dû effectuer une entrée	fds: nombre de descripteurs de fichiers (OpenBSD uniquement)
ssize: taille de la pile	io_write_op: nombre de fois où le système a dû effectuer une sortie	swap: taille de l’espace d’échange utilisé
cputime_user: temps CPU total passé en mode utilisateur (indiqué en secondes)		io_read_op: nombre de fois où le système a dû effectuer une entrée
cputime_system: temps CPU total passé en mode système (indiqué en secondes)		io_write_op: nombre de fois où le système a dû effectuer une sortie
state: état du processus (attente disque/en cours d’exécution/en veille/arrêt de traçage/zombie/autre)		
ctx_switches: nombre de changements de contexte		
threads: nombre de threads (non pris en charge pour NetBSD)		
page_faults: nombre de défauts de page		
fds: nombre de descripteurs de fichiers (OpenBSD uniquement)		
swap: taille de l’espace d’échange utilisé		
io_read_op: nombre de fois où le système a dû effectuer une entrée		
io_write_op: nombre de fois où le système a dû effectuer une sortie		

Windows

Les paramètres de processus suivants sont renvoyés sous Windows pour chaque mode :

mode=process	mode=thread	mode=summary
pid: PID	pid: PID	name: nom du processus
ppid: PID parent	ppid: PID parent	processes: nombre de processus
name: nom du processus	name: nom du processus	vmsize: taille de la mémoire virtuelle
user: utilisateur sous lequel le processus s’exécute	user: utilisateur sous lequel le processus s’exécute	wkset: taille de l’ensemble de travail du processus
sid: SID de l’utilisateur	sid: SID de l’utilisateur	cputime_user: temps CPU total passé en mode utilisateur (indiqué en millisecondes)
vmsize: taille de la mémoire virtuelle	tid: ID du thread	cputime_system: temps CPU total passé en mode système (indiqué en millisecondes)
wkset: taille de l’ensemble de travail du processus		threads: nombre de threads
cputime_user: temps CPU total passé en mode utilisateur (indiqué en millisecondes)		page_faults: nombre de défauts de page
cputime_system: temps CPU total passé en mode système (indiqué en millisecondes)		handles: nombre de handles
threads: nombre de threads		io_read_b: octets d’E/S lus
page_faults: nombre de défauts de page		io_write_b: octets d’E/S écrits
handles: nombre de handles		io_read_op: opérations de lecture d’E/S
io_read_b: octets d’E/S lus		io_write_op: opérations d’écriture d’E/S

mode=process	mode=thread	mode=summary
io_write_b: octets d'E/S écrits		io_other_b: octets d'E/S transférés, autres que les opérations de lecture et d'écriture
io_read_op: opérations de lecture d'E/S		io_other_op: opérations d'E/S autres que les opérations de lecture et d'écriture
io_write_op: opérations d'écriture d'E/S		
io_other_b: octets d'E/S transférés, autres que les opérations de lecture et d'écriture		
io_other_op: opérations d'E/S autres que les opérations de lecture et d'écriture		

Paramètres de l'interface de l'hôte inaccessible/non disponible

Aperçu

Plusieurs paramètres de configuration définissent le comportement du serveur Zabbix lorsqu'une vérification d'agent (Zabbix, SNMP, IPMI, JMX) échoue et qu'une interface d'hôte devient inaccessible.

Interface inaccessible

Une interface d'hôte est considérée comme inaccessible après un échec de vérification (erreur réseau, délai d'attente) par les agents Zabbix, SNMP, IPMI ou JMX. Depuis Zabbix 6.2.0, les vérifications actives de l'agent Zabbix influencent également la disponibilité de l'interface. Lorsque les vérifications actives deviennent indisponibles, elles contribuent à l'état global de disponibilité de l'interface de l'agent.

À partir du moment où une interface devient inaccessible, **UnreachableDelay** définit la fréquence à laquelle elle est revérifiée à l'aide de l'un des éléments (y compris les règles LLD). Ces revérifications sont effectuées par les pollers d'inaccessibilité (ou par les pollers IPMI pour les vérifications IPMI). Par défaut, l'intervalle entre deux vérifications de disponibilité successives est de 15 secondes.

Attention:
Les vérifications effectuées par les pollers asynchrones ne sont pas déplacées vers les pollers d'inaccessibilité.

Dans le journal du serveur Zabbix, l'inaccessibilité est indiquée par des messages tels que :

```
Zabbix agent item "system.cpu.load[percpu,avg1]" on host "New host" failed: first network error, wait for
Zabbix agent item "system.cpu.load[percpu,avg15]" on host "New host" failed: another network error, wait f
```

Les messages du journal précisent l'élément exact en échec ainsi que son type (agent Zabbix).

Note:
Le paramètre *Timeout* influence également la rapidité avec laquelle une interface est revérifiée pendant une période d'inaccessibilité. Si Timeout est défini sur 20 secondes et UnreachableDelay sur 30 secondes, la prochaine vérification aura lieu 50 secondes après la première tentative.

Le paramètre **UnreachablePeriod** définit la durée totale de la période d'inaccessibilité. Par défaut, UnreachablePeriod est de 45 secondes. Cette valeur doit être plusieurs fois supérieure à UnreachableDelay afin de garantir qu'une interface soit revérifiée plusieurs fois avant d'être marquée comme indisponible.

Un élément interne, zabbix[host,active_agent,available], permet de surveiller la disponibilité des vérifications actives dans les scénarios d'inaccessibilité.

Retour de l'interface en disponible

Lorsque la période d'inaccessibilité est terminée, l'interface est à nouveau interrogée, priorité décroissante pour l'élément qui a rendu l'interface en état inaccessible. Si l'interface inaccessible réapparaît, la surveillance revient à la normale automatiquement :

```
resuming Zabbix agent checks on host "New host": connection restored
```

Note:
Une fois l'interface disponible, l'hôte n'interroge pas tous ses éléments immédiatement pour deux raisons :

- Cela pourrait surcharger l'hôte.
- Le temps de restauration de l'interface ne correspond pas toujours à la planification d'interrogation de l'élément prévue.

Ainsi, une fois l'interface disponible, les éléments ne sont pas interrogés immédiatement, mais ils sont reportés à leur prochaine planification.

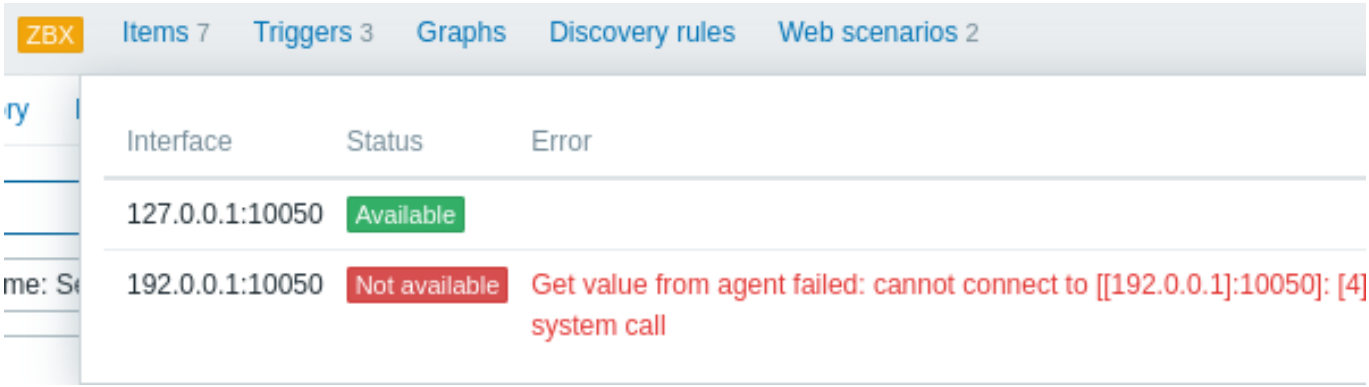
Interface indisponible

Après la fin de la période **UnreachablePeriod** et si l'interface n'est pas réapparue, l'interface est considérée comme indisponible.

Dans le journal du serveur, cela est indiqué par des messages comme ceux-ci :

```
temporarily disabling Zabbix agent checks on host "New host": interface unavailable
```

et dans le frontend l'icône de disponibilité de l'hôte passe du vert/gris au jaune/rouge (les détails de l'interface indisponible peuvent être consultés dans l'infobulle affichée lorsque le pointeur de la souris est placé sur l'icône de disponibilité de l'hôte) :



Le paramètre **UnavailableDelay** définit la fréquence à laquelle une interface est vérifiée pendant son indisponibilité.

Par défaut, elle est de 60 secondes (dans ce cas, « temporarily disabling », dans le message du journal ci-dessus, signifie que les vérifications seront désactivées pendant une minute).

Lorsque la connexion à l'interface est rétablie, la supervision revient également automatiquement à la normale :

```
enabling Zabbix agent checks on host "New host": interface became available
```

Surveillance à distance des statistiques Zabbix

Vue d'ensemble

Il est possible de rendre certaines métriques internes du serveur Zabbix et du proxy accessibles à distance par une autre instance Zabbix ou un outil tiers. Cela peut être utile afin que les prestataires de support/de services puissent surveiller à distance les serveurs/proxys Zabbix de leurs clients, ou, dans les organisations où Zabbix n'est pas l'outil de supervision principal, afin que les métriques internes de Zabbix puissent être surveillées par un système tiers dans une configuration de supervision globale.

Les statistiques internes de Zabbix sont exposées à un ensemble configurable d'adresses répertoriées dans le nouveau paramètre 'StatsAllowedIP' serveur/proxy. Les requêtes ne seront acceptées qu'à partir de ces adresses.

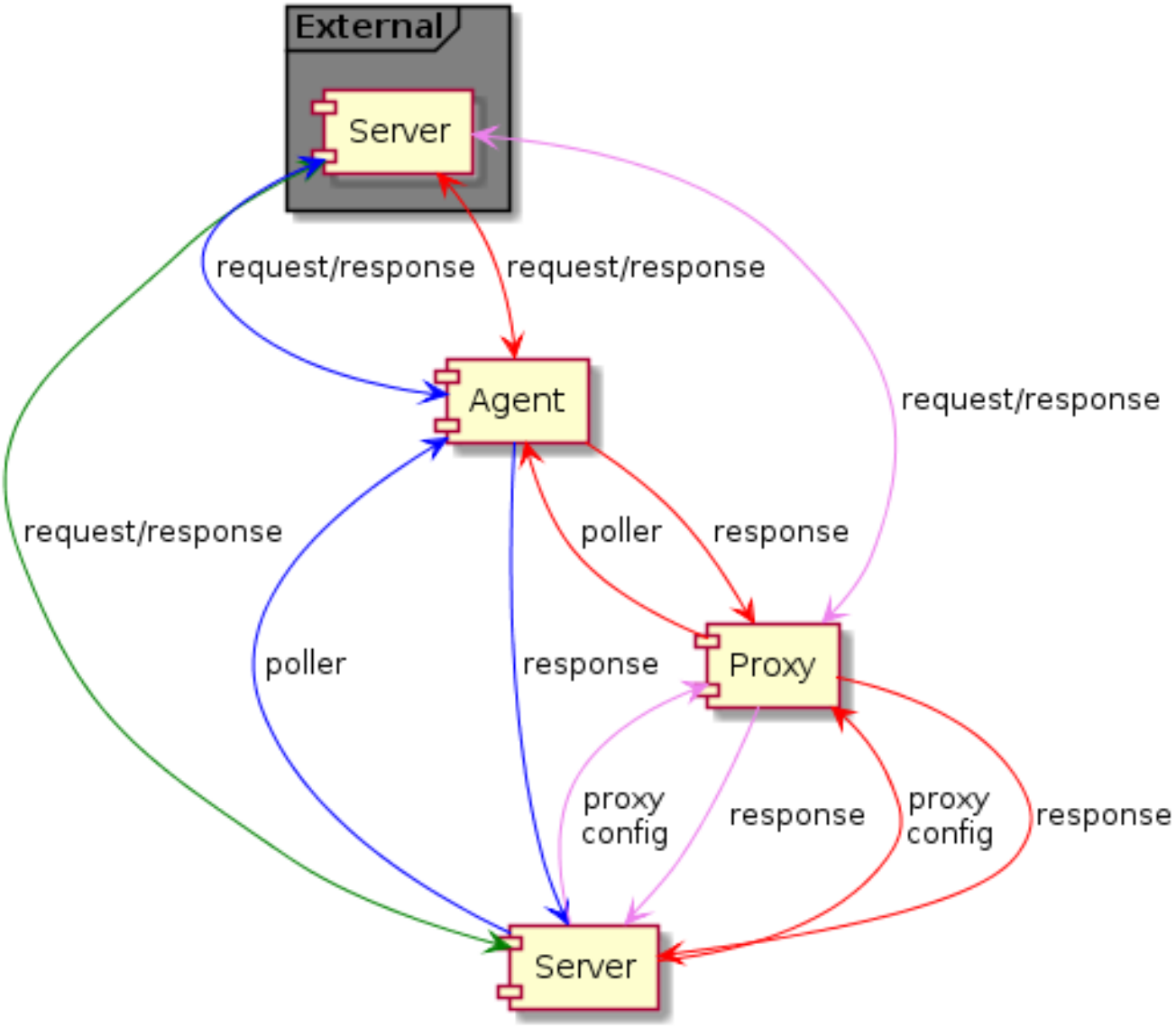
Eléments

Pour configurer l'interrogation des statistiques internes sur une autre instance Zabbix, vous pouvez utiliser deux éléments :

- L'élément interne `zabbix[stats,<ip>,<port>]` - pour des requêtes directes à distance du serveur/proxy Zabbix. <ip> et <port> sont utilisés pour identifier l'instance cible.
- L'élément agent `zabbix.stats[<ip>,<port>]` - pour des requêtes basées sur l'agent du serveur/proxy Zabbix. <ip> et <port> sont utilisés pour identifier l'instance cible.

Voir aussi : Éléments internes, Éléments agent Zabbix

Le schéma suivant illustre l'utilisation de l'un ou l'autre élément en fonction du contexte.



- - Serveur → instance Zabbix externe (`zabbix[stats,<ip>,<port>]`)
- - Serveur → proxy → instance Zabbix externe (`zabbix[stats,<ip>,<port>]`)
- - Serveur → agent → instance Zabbix externe (`zabbix.stats[<ip>,<port>]`)
- - Serveur → proxy → agent → instance Zabbix externe (`zabbix.stats[<ip>,<port>]`)

Pour vous assurer que l'instance cible permet de l'interroger par l'instance externe, indiquez l'adresse de l'instance externe dans le paramètre 'StatsAllowedIP' sur l'instance cible.

Métriques exposées

Les éléments de statistiques collectent les statistiques en bloc et renvoient un JSON, qui sert de base aux éléments dépendants pour obtenir leurs données. Les métriques internes suivantes sont renvoyées par l'un ou l'autre des deux éléments :

- `zabbix[boottime]`
- `zabbix[hosts]`
- `zabbix[items]`
- `zabbix[items_unsupported]`
- `zabbix[preprocessing]` (serveur uniquement)
- `zabbix[preprocessing_queue]` (serveur uniquement)
- `zabbix[process,<type>,<mode>,<state>]` (statistiques basées uniquement sur le type de processus)
- `zabbix[rcache,<cache>,<mode>]`
- `zabbix[requiredperformance]`
- `zabbix[triggers]` (serveur uniquement)
- `zabbix[uptime]`
- `zabbix[vcache,buffer,<mode>]` (serveur uniquement)

- zabbix[vcache,cache,<parameter>]
- zabbix[version]
- zabbix[vmware,buffer,<mode>]
- zabbix[wcache,<cache>,<mode>] (type de cache « trends » sur le serveur uniquement)

Modèles

Des modèles sont disponibles pour la surveillance à distance des métriques internes du serveur Zabbix ou du proxy depuis une instance externe :

- État du serveur Zabbix distant
- État du proxy Zabbix distant

Notez que pour utiliser un modèle pour la surveillance à distance de plusieurs instances externes, un hôte distinct est requis pour la surveillance de chaque instance externe.

Processus trapper

La réception des requêtes de métriques internes provenant d'une instance Zabbix externe est gérée par le processus trapper, qui valide la requête, collecte les métriques, crée le tampon de données JSON et renvoie le JSON préparé, par exemple, depuis le serveur :

```
{
  "response": "success",
  "data": {
    "boottime": N,
    "uptime": N,
    "hosts": N,
    "items": N,
    "items_unsupported": N,
    "preprocessing": {
        "queued": {
          "count": N,
          "size": N
        },
        "direct": {
          "count": N,
          "size": N
        },
        "queue": N
    },
    "preprocessing_queue": N,
    "process": {
      "alert manager": {
        "busy": {
          "avg": N,
          "max": N,
          "min": N
        },
        "idle": {
          "avg": N,
          "max": N,
          "min": N
        },
        "count": N
      },
...
    },
    "queue": N,
    "rcache": {
      "total": N,
      "free": N,
      "pfree": N,
      "used": N,
      "pused": N
    },
```

```
    "requiredperformance": N,
    "triggers": N,
    "uptime": N,
    "vcache": {
      "buffer": {
        "total": N,
        "free": N,
        "pfree": N,
        "used": N,
        "pused": N
      },
      "cache": {
        "requests": N,
        "hits": N,
        "misses": N,
        "mode": N
      }
    },
    "vmware": {
      "total": N,
      "free": N,
      "pfree": N,
      "used": N,
      "pused": N
    },
    "version": "N",
    "wcache": {
      "values": {
        "all": N,
        "float": N,
        "uint": N,
        "str": N,
        "log": N,
        "text": N,
        "not supported": N
      },
      "history": {
        "pfree": N,
        "free": N,
        "total": N,
        "used": N,
        "pused": N
      },
      "index": {
        "pfree": N,
        "free": N,
        "total": N,
        "used": N,
        "pused": N
      },
      "trend": {
        "pfree": N,
        "free": N,
        "total": N,
        "used": N,
        "pused": N
      }
    }
  }
}
```


Éléments de file d’attente internes

Deux autres éléments permettent spécifiquement d'interroger à distance les statistiques de la file d'attente interne sur une autre instance Zabbix :

- Élément interne `zabbix[stats,<ip>,<port>,queue,<from>,<to>]` - pour interroger directement la file d'attente interne d'un serveur/proxy Zabbix distant
- Élément agent `zabbix.stats[<ip>,<port>,queue,<from>,<to>]` - pour interroger la file d'attente interne d'un serveur/proxy Zabbix distant via l'agent

Voir également : Éléments internes, Éléments Zabbix agent

Configuration de Kerberos avec Zabbix

Aperçu

L'authentification Kerberos peut être utilisée dans la surveillance web et les éléments HTTP dans Zabbix.

Cette page décrit un exemple de configuration de Kerberos pour que le serveur Zabbix effectue la surveillance web de `www.example.com` avec un principal Kerberos pour le processus Zabbix sur Debian/Ubuntu.

Configuration

1. Installer le KDC et les utilitaires client :

```
sudo apt update
sudo apt install krb5-kdc krb5-admin-server krb5-user
```

Lors de la configuration des paquets, répondre aux invites, par exemple :

```
Default Kerberos version 5 realm: EXAMPLE.COM
Kerberos servers for your realm: localhost (or your FQDN)
Administrative server for your Kerberos realm: localhost (or your FQDN)
```

2. Mapper un nom d'hôte convivial (facultatif, pour les tests locaux).

Modifier /etc/hosts et ajouter une entrée pour votre DC et votre serveur web si vous ne disposez pas de DNS :

```
sudo vi /etc/hosts
```

Exemple de ligne à ajouter :

```
192.168.1.100  dc01.example.com dc01
```

3. Configurer le client Kerberos et le realm KDC :

```
sudo vi /etc/krb5.conf
```

Exemple de paramètres :

```
[libdefaults]
    default_realm = EXAMPLE.COM
    dns_lookup_realm = false
    dns_lookup_kdc = false
    rdns = false
    ticket_lifetime = 24h
    renew_lifetime = 7d
    forwardable = true

[realms]
    EXAMPLE.COM = {
        kdc = dc01.example.com
        admin_server = dc01.example.com
    }

[domain_realm]
    .example.com = EXAMPLE.COM
    example.com = EXAMPLE.COM
```

Si vous prévoyez d'utiliser `.localdomain` ou d'autres noms non publics, ajoutez des mappages explicites domaine→realm afin que le mappage nom d'hôte→realm fonctionne. Les incohérences à ce niveau provoquent des erreurs `Server not found in Kerberos database`.

4. Initialiser la base de données Kerberos (une seule fois, sur l'hôte KDC). Définir un mot de passe principal sécurisé lorsque vous y êtes invité :

```
sudo krb5_newrealm
```

5. Créer le principal `HTTP/host.fqdn@REALM` en utilisant le nom d'hôte exact que les clients utiliseront ; privilégier les minuscules (par exemple, `HTTP/web.example.com@EXAMPLE.COM`). Une différence de casse ou de nom provoque l'erreur `Server not found in Kerberos database`.

```
sudo kadmin.local
```

Dans `kadmin.local` :

```
addprinc kerb-admin@EXAMPLE.COM     # principal administratif
addprinc -randkey HTTP/dc01.example.com@EXAMPLE.COM
ktadd -k /etc/apache2/http.keytab HTTP/dc01.example.com@EXAMPLE.COM
quit
```

Déplacer le keytab vers l'hôte web (ou le conserver localement s'il s'agit de la même machine) et définir des permissions utilisables par Apache :

```
chown www-data:www-data /etc/apache2/http.keytab
chmod 600 /etc/apache2/http.keytab
#### vérifier
sudo -u www-data -k /etc/apache2/http.keytab
```

6. Installer et activer le module GSSAPI d'Apache :

```
sudo apt install libapache2-mod-auth-gssapi
sudo a2enmod auth_gssapi
sudo a2enmod headers
sudo systemctl restart apache2
```

> **Attention:**
> Toutes les versions de mod_auth_gssapi ne prennent pas en charge chaque directive `Gssapi*`. Si Apache échoue avec `Invalid command 'GssapiCredStore'`, supprimer la directive non prise en charge ou mettre à niveau le module.

7. Configurer un VirtualHost (adapter `DocumentRoot` / le chemin vers votre interface Zabbix) :

```
sudo vi /etc/apache2/sites-available/zabbix.conf
```

Dans `zabbix.conf` :

```
<VirtualHost *:80>
    ServerName dc01.example.com
    DocumentRoot /usr/share/zabbix/ui
    <Directory /usr/share/zabbix/ui>
        Options FollowSymLinks
        AllowOverride None
        Require all granted
        AuthType GSSAPI
        AuthName "Kerberos Login"
        GssapiCredStore keytab:/etc/apache2/http.keytab
        GssapiLocalName On
        Require valid-user
    </Directory>
    RequestHeader set X-Remote-User %{REMOTE_USER}s env=REMOTE_USER
    RequestHeader unset Authorization
</VirtualHost>
```

Redémarrer Apache :

```
sudo systemctl restart apache2
```

8. Activer et démarrer les services KDC, puis vérifier les ports en écoute (hôte KDC) :

```
sudo systemctl enable --now krb5-kdc krb5-admin-server
ss -tnlp | grep :80    # ou : sudo netstat -tnlp | grep :80
```

9. Obtenir un TGT pour les tests (exécuter la commande avec l'utilisateur qui utilisera le ticket).

Vous devez voir krbtgt/EXAMPLE.COM@EXAMPLE.COM dans la liste des tickets. Exécuter kinit avec le même utilisateur du système d'exploitation que celui qui a besoin du ticket (par exemple, zabbix pour les vérifications web ou www-data/Apache pour les tests SSO interactifs dans le navigateur). Les tickets attribués à un autre utilisateur du système d'exploitation ne seront pas visibles, sauf si KRB5CCNAME et les permissions sont ajustés.

```
kinit kerb-admin@EXAMPLE.COM
klist
```

10. Tester l'échange SPNEGO avec curl (depuis un client disposant d'un TGT valide). Un résultat 200 OK (ou une redirection vers l'application) indique que SPNEGO a réussi :

```
curl -v --negotiate -u : http://dc01.example.com/
```

11. Facultatif : si l'interface Zabbix doit accepter les connexions authentifiées par HTTP, activer l'authentification HTTP dans l'interface Zabbix.

Dans le fichier de configuration de l'interface (zabbix.conf.php), définir :

```
$ZBX_FEATURE_FLAGS['http_auth_enabled'] = true;
```

Puis configurer l'authentification HTTP dans l'interface web.

12. Configuration du navigateur (Firefox est utilisé comme exemple) : définir network.negotiate-auth.trusted-uris sur le ou les hôtes qui effectuent la négociation (dc01.example.com) afin que le navigateur envoie automatiquement les jetons Kerberos.

Dans about:config :

```
network.negotiate-auth.trusted-uris = dc01.example.com
```

En accédant maintenant à http://dc01.example.com, vous devriez être directement connecté à Zabbix sans afficher le formulaire.

13. Maintenir les clés et les tickets à jour. La durée de vie par défaut d'un ticket Kerberos est d'environ 10 heures. Ajouter une tâche cron ou un minuteur systemd pour éviter les expirations :

```
####pour le service web
kinit -kt /etc/apache2/http.keytab HTTP/dc01.example.com@EXAMPLE.COM
####pour l'utilisateur de supervision
kinit -kt /var/lib/zabbix/kerb.keytab kerb-admin@EXAMPLE.COM
```

14. Vérifications de maintenance :

- klist -k /etc/apache2/http.keytab - vérifier que le principal de service est présent dans le keytab.
- sudo tail -f /var/log/apache2/error.log - surveiller les erreurs GSSAPI (gss_acquire_cred[_from]() failed to get server creds signifie un problème de keytab ou de permissions, ou l'absence du principal).
- Si curl --negotiate renvoie 401/403, cela signifie souvent que le principal est incorrect, qu'aucun ticket n'est disponible, que l'en-tête d'hôte ne correspond pas ou qu'il existe un problème de permissions du système de fichiers ; vérifier les journaux et les mappages de domaines dans /etc/krb5.conf.

Remarques sur la sécurité et les permissions des fichiers

Les fichiers keytab doivent être lisibles uniquement par le compte qui en a besoin. Exemples de permissions : 0400, avec zabbix:zabbix comme propriétaire pour un keytab utilisateur zabbix, ou 0440, avec root:www-data comme propriétaire pour un keytab Apache.

Évitez de stocker des mots de passe en clair à longue durée de validité sur l'hôte. Utilisez des keytabs ou des principaux de machine joints au domaine lorsque cela est possible.

Lors de l'exécution de tests ou de scripts qui définissent KRB5CCNAME ou copient des keytabs, vérifiez à nouveau le propriétaire et les permissions après l'opération : lorsqu'un serveur web rejette des identifiants, le problème provient souvent des permissions d'un fichier.

paramètres de modbus.get

Vue d'ensemble

Le tableau ci-dessous présente les détails des paramètres de l'élément modbus.get.

Paramètres

Parameter	Description	Defaults	Example
endpoint	Protocole et adresse du point de terminaison, définis comme `protocol://connection_string` Valeurs possibles du protocole : *rtu*, *ascii* (Agent 2 uniquement), *tcp* Format de la chaîne de connexion : avec *tcp* - `address:port` avec ligne série : *rtu*, *ascii* - `port_name:speed:params` où 'speed' - 1200, 9600, etc. 'params' - bits de données (5,6,7 ou 8), parité (n,e ou o pour none/even/odd), bits d'arrêt (1 ou 2)	protocol: none *rtu/ascii* protocol: port_name: none speed: 115200 params: 8n1 *tcp* protocol: address: none port: 502	tcp://192.168.6.1:511 tcp://192.168.6.2 tcp://[::1]:511 tcp://::1 tcp://localhost:511 tcp://localhost rtu://COM1:9600:8n ascii://COM2:1200:7o2 rtu://ttyS0:9600 ascii://ttyS1
slave id	Adresse Modbus du périphérique auquel il est destiné (de 1 à 247), voir MODBUS Messaging Implementation Guide (page 23) le périphérique tcp (pas la GW) ignorera ce champ	serial: 1 tcp: 255 (0xFF)	2
function	Vide ou valeur d'une fonction prise en charge : 1 - Read Coil, 2 - Read Discrete Input, 3 - Read Holding Registers, 4 - Read Input Registers	empty	3
address	Adresse du premier registre, coil ou input. Si 'function' est vide, alors 'address' doit être dans la plage suivante : Coil - 00001 - 09999 Discrete input - 10001 - 19999 Input register - 30001 - 39999 Holding register - 40001 - 49999 Si 'function' n'est pas vide, le champ 'address' ira de 0 à 65535 et sera utilisé sans modification (PDU)	empty function: 00001 non-empty function: 0	9999
count	Nombre de 'type' séquencés qui seront lus depuis le périphérique, où : pour Coil ou Discrete input, le 'type' = 1 bit dans les autres cas : (count*sizeof(type))/2 = nombre réel de registres à lire Si 'offset' n'est pas 0, la valeur sera ajoutée au 'real count' La plage acceptable pour 'real count' est 1:65535	1	2
type	Type de données : pour Read Coil et Read Discrete Input - *bit* pour Read Holding Registers et Read Input Registers : *int8* - 8bit *uint8* - 8bit (non signé) *int16* - 16bit *uint16* - 16bit (non signé) *int32* - 32bit *uint32* - 32bit (non signé) *float* - 32bit *uint64* - 64bit (non signé) *double* - 64bit	bit uint16	uint64

Parameter	Description	Defaults	Example
endianness	Type d'endianness : *be* - Big Endian *le* - Little Endian *mbe* - Mid-Big Endian *mle* - Mid-Little Endian Limitations : pour 1 bit - be pour 8 bits - be,le pour 16 bits - be,le	be	le
offset	Nombre de registres, à partir de 'address', dont le résultat sera ignoré. La taille de chaque registre est de 16bit (nécessaire pour prendre en charge les équipements qui ne prennent pas en charge l'accès en lecture aléatoire).	0	4

Création de noms personnalisés de compteurs de performance pour VMware

Vue d'ensemble

Le chemin du compteur de performance VMware a le format `group/counter[rollup]` où :

- `group` - le groupe du compteur de performance, par exemple, *cpu*
- `counter` - le nom du compteur de performance, par exemple, *usagemhz*
- `rollup` - le type de consolidation du compteur de performance, par exemple, *average*

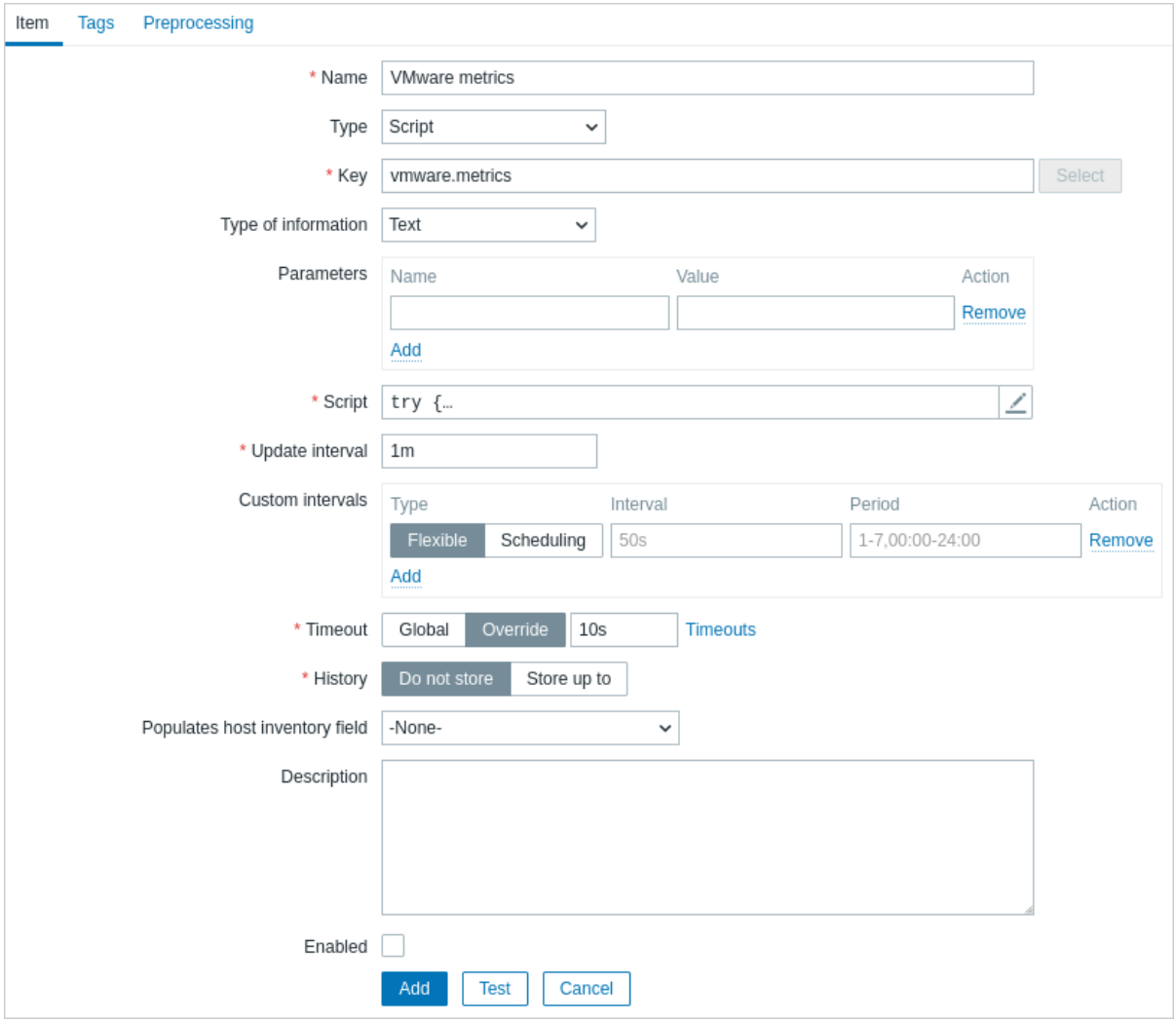
Ainsi, l'exemple ci-dessus donnerait le chemin de compteur suivant : `cpu/usagemhz[average]`

Les descriptions des groupes de compteurs de performance, les noms des compteurs et les types de consolidation peuvent être trouvés dans la documentation VMware.

Il est possible d'obtenir les noms internes et de créer des noms de compteurs de performance personnalisés en utilisant un élément de type script dans Zabbix.

Configuration

1. Créez un élément Script désactivé sur l'hôte VMware principal (où l'élément **eventlog[]** est présent) avec les paramètres suivants :



- *Name*: VMware metrics
- *Type*: Script
- *Key*: vmware.metrics
- *Type of information*: Text
- *Script*: copiez et collez le script fourni ci-dessous
- *Timeout*: 10
- *History*: Do not store
- *Enabled*: unmarked

Script

```
try {
    Zabbix.log(4, 'vmware metrics script');

    var result, resp,
    req = new HttpRequest();
    req.addHeader('Content-Type: application/xml');
    req.addHeader('SOAPAction: "urn:vim25/6.0"');

    login = '<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:urn="urn:vi
    <soapenv:Header/>\
    <soapenv:Body>\
        <urn:Login>\
            <urn:_this type="SessionManager">SessionManager</urn:_this>\
            <urn:userName>{$VMWARE.USERNAME}</urn:userName>\
            <urn:password>{$VMWARE.PASSWORD}</urn:password>\
        </urn:Login>\
    </soapenv:Body>\
</soapenv:Envelope>'
    resp = req.post("{$VMWARE.URL}", login);
    if (req.getStatus() != 200) {
```

```
        throw 'Response code: '+req.getStatus();
    }

    query = '<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:urn="urn:vi
<soapenv:Header/>\
    <soapenv:Body>\
        <urn:RetrieveProperties>\
            <urn:_this type="PropertyCollector">propertyCollector</urn:_this>\
            <urn:specSet>\
                <urn:propSet>\
                   <urn:type>PerformanceManager</urn:type>\
                   <urn:pathSet>perfCounter</urn:pathSet>\
                </urn:propSet>\
                <urn:objectSet>\
                   <urn:obj type="PerformanceManager">PerfMgr</urn:obj>\
                </urn:objectSet>\
            </urn:specSet>\
        </urn:RetrieveProperties>\
    </soapenv:Body>\
</soapenv:Envelope>'
    resp = req.post("{$VMWARE.URL}", query);
    if (req.getStatus() != 200) {
        throw 'Response code: '+req.getStatus();
    }
    Zabbix.log(4, 'vmware metrics=' + resp);
    result = resp;

    logout = '<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:urn="urn:v
    <soapenv:Header/>\
    <soapenv:Body>\
        <urn:Logout>\
            <urn:_this type="SessionManager">SessionManager</urn:_this>\
        </urn:Logout>\
    </soapenv:Body>\
</soapenv:Envelope>'

    resp = req.post("{$VMWARE.URL}",logout);
    if (req.getStatus() != 200) {
        throw 'Response code: '+req.getStatus();
    }

} catch (error) {
    Zabbix.log(4, 'vmware call failed : '+error);
    result = {};
}

return result;
```

Une fois l'élément configuré, cliquez sur le bouton *Test*, puis sur *Get value*.



Copiez le XML reçu dans n'importe quel formateur XML et trouvez la métrique souhaitée.

Exemple de XML pour une métrique :

```
<PerfCounterInfo xsi:type="PerfCounterInfo">
    <key>6</key>
```

```
    <nameInfo>
        <label>Usage in MHz</label>
        <summary>CPU usage in megahertz during the interval</summary>
        <key>usagemhz</key>
    </nameInfo>
    <groupInfo>
        <label>CPU</label>
        <summary>CPU</summary>
        <key>cpu</key>
    </groupInfo>
    <unitInfo>
        <label>MHz</label>
        <summary>Megahertz</summary>
        <key>megaHertz</key>
    </unitInfo>
    <rollupType>average</rollupType>
    <statsType>rate</statsType>
    <level>1</level>
    <perDeviceLevel>3</perDeviceLevel>
</PerfCounterInfo>
```

Utilisez XPath pour extraire le chemin du compteur à partir du XML reçu. Pour l'exemple ci-dessus, le XPath sera :

field	xPath	value
group	//groupInfo[../key=6]/key	cpu
counter	//nameInfo[../key=6]/key	usagemhz
rollup	//rollupType[../key=6]	average

Le chemin du compteur de performance résultant dans ce cas est : `cpu/usagemhz[average]`

Valeurs de retour pour system.sw.packages.get

Aperçu

Cette section fournit des détails sur la valeur de retour de l'élément d'agent Zabbix `system.sw.packages.get`.

Détails

La sortie de cet élément est un tableau d'objets contenant chacun les propriétés suivantes :

- **name** - nom du paquet
- **manager** - gestionnaire de paquets ayant signalé ces données (`rpm`, `dpkg`, `pacman`, `pkgtool` ou `portage`)
- **version** - version du paquet
- **size** - taille du paquet non compressé en octets (si elle n'est pas disponible, définir sur 0)
- **arch** - architecture du paquet
- **buildtime** - un objet avec 2 entrées :
  - **timestamp** - horodatage UNIX du moment où le paquet a été compilé (si non disponible, définir sur 0)
  - **value** - date et heure lisibles par l'humain du moment où le paquet a été compilé (si non disponibles, définir sur une chaîne vide)
- **installtime** - un objet avec 2 entrées :
  - **timestamp** - horodatage UNIX du moment où le paquet a été installé (si non disponible, définir sur 0)
  - **value** - date et heure lisibles par l'humain du moment où le paquet a été installé (si non disponibles, définir sur une chaîne vide)

Exemple :

```
[
    {
        "name": "util-linux-core",
        "manager": "rpm",
        "version": "2.37.4-3.el9",
        "size": 1296335,
        "arch": "x86_64",
        "buildtime": {
```

```
            "timestamp" : 1653552239,
            "value" : "Sep 20 01:39:40 2021 UTC"
        },
        "installtime": {
            "timestamp" : 1660780885,
            "value" : "Aug 18 00:01:25 2022 UTC"
        }
    },
    {
        "name": "xfonts-base",
        "manager": "dpkg",
        "version": "1:1.0.5",
        "size": 7337984,
        "arch": "all",
        "buildtime": {
            "timestamp": 0,
            "value": ""
        },
        "installtime": {
            "timestamp": 0,
            "value": ""
        }
    }
]
```

Valeurs de retour pour net.dns.get

Aperçu

Cette section fournit des détails sur la valeur de retour pour l'élément d'agent Zabbix 2 `net.dns.get`.

Détails

La sortie de cet élément est un objet contenant des informations sur l'enregistrement DNS, basées sur les paramètres fournis dans la clé de l'élément.

Par exemple, l'élément `net.dns.get[,example.com]` peut renvoyer le JSON suivant pour une requête refusée :

```
{
    "flags": [
        "RA"
    ],
    "query_time": "0.019030",
    "question_section": [
        {
            "qclass": "IN",
            "qname": "example.com.",
            "qtype": "SOA"
        }
    ],
    "response_code": "REFUSED",
    "zbx_error_code": 0
}
```

En spécifiant l'adresse IP du serveur DNS, l'élément `net.dns.get[192.0.2.1,example.com]` peut renvoyer le JSON suivant :

```
{
    "answer_section": [
        {
            "class": "IN",
            "name": "example.com.",
            "rdata": {
                "expire": 1209600,
                "mbox": "noc.dns.example.org.",
                "minttl": 3600,
```

```
                "ns": "ns.example.org.",
                "refresh": 7200,
                "retry": 3600,
                "serial": 2022091378
            },
            "rdlength": 44,
            "ttl": 1205,
            "type": "SOA"
        }
    ],
    "flags": [
        "RA"
    ],
    "query_time": "0.029556",
    "question_section": [
        {
            "qclass": "IN",
            "qname": "example.com.",
            "qtype": "SOA"
        }
    ],
    "response_code": "NOERROR",
    "zbx_error_code": 0
}
```


En cas de problème de connexion, l'élément `net.dns.get[192.0.2.1,example.com]` peut renvoyer une erreur :


```
{
    "zbx_error_code": -1,
    "zbx_error_msg": "Communication error: read udp 192.0.2.1:12345->192.0.2.1:53: i/o timeout"
}
```


Les types de codes d'erreur suivants sont possibles :

Scenario	"zbx_error_code"	"zbx_error_msg"
Aucune erreur et la réponse DNS a été reçue et analysée.	0	
Le DNS est indisponible.	-1	"Communication error"
Une erreur se produit pendant l'analyse JSON	-2	"Received unexpected response"

Avec des paramètres supplémentaires, l'élément `net.dns.get[192.0.2.1,example.com,ANY,5,5,tcp,"cdflag,rdflag,dnssec,n` peut renvoyer le JSON suivant :


```
{
    "additional_section": [
        {
            "extended_rcode": 32768,
            "name": ".",
            "rdata": {
                "options": [
                {
                    "code": 0,
                    "nsid": "67 70 64 6e 73 2d 6c 70 70"
                }
                ]
        },
            "rdlength": 13,
            "type": "OPT",
            "udp_payload": 512
        }
    ],
    "answer_section": [
        {
```

```
            "class": "IN",
            "name": "example.com.",
            "rdata": {
                "a": "192.0.2.1"
            },
            "rdlength": 4,
            "ttl": 19308,
            "type": "A"
        },
        {
            "class": "IN",
            "name": "example.com.",
            "rdata": {
                "algorithm": 13,
                "expiration": 1704715951,
                "inception": 1702910624,
                "key_tag": 21021,
                "labels": 2,
                "orig_ttl": 86400,
                "signature": "HVBOBcJJQy0S08J3f8kviPj8UkEUj7wmyiMyQqPSWgQIY9SCEJ5plq6KuxJmtAek1txZWXDo+6tp
                "signer_name": "example.com.",
                "type_covered": "A"
            },
            "rdlength": 95,
            "ttl": 19308,
            "type": "RRSIG"
        }
    ],
    "flags": [
        "RD",
        "RA",
        "AD",
        "CD"
    ],
    "query_time": "0.058221",
    "question_section": [
        {
            "qclass": "IN",
            "qname": "example.com.",
            "qtype": "ANY"
        }
    ],
    "response_code": "NOERROR",
    "zbx_error_code": 0
}
```


Voir aussi

Pour plus d'informations sur les enregistrements DNS, voir :

- Domain Names - Implementation and Specification
- Domain Name System (DNS) Parameters

Valeurs de retour pour net.if.get

Vue d'ensemble

Cette section fournit des détails sur les valeurs de retour pour l'élément `net.if.get`. L'élément est pris en charge sous Linux par Zabbix agent et Zabbix agent 2.

Détails

Cet élément renvoie des informations détaillées sur les interfaces réseau dont les noms correspondent à l'expression régulière donnée. La sortie est structurée comme un objet JSON contenant les tableaux `config` et `values`.

La sortie contient les propriétés suivantes :

- **name** - nom de l'interface, lu depuis `/proc/net/dev` ;
- **ifalias** - alias de l'interface, lu depuis `/sys/class/net/<name>/ifalias` ;
- **mac** - adresse MAC de l'interface, lue depuis `/sys/class/net/<name>/address` ;
- **type** - type d'interface (`physical`, `virtual`, `loopback`) ;
- **speed** - vitesse du lien en Mbits/sec, spécifique à Ethernet, lue depuis `/sys/class/net/<name>/speed` ;
- **duplex** - dernière valeur duplex ou valeur duplex actuelle de l'interface (`half`, `full`), spécifique à Ethernet, lue depuis `/sys/class/net/<name>/duplex` ;
- **administrative_state** - état configuré (`down`, `up`), lu depuis `/sys/class/net/<name>/flags` ;
- **operational_state** - état opérationnel RFC2863 de l'interface (`unknown`, `notpresent`, `down`, `lowerlayerdown`, `testing`, `dormant`, `up`), lu depuis `sys/class/net/<name>/operstate` ;
- **carrier** - présence du lien physique (0 - le lien physique est désactivé, 1 - le lien physique est activé), lue depuis `/sys/class/net/<name>/carrier` ;
- **carrier_changes** - nombre total de changements d'état du lien de up à down et inversement, lu depuis `/sys/class/net/<name>/car` ;
- **carrier_up_count** - nombre de fois où le lien est devenu up, lu depuis `/sys/class/net/<name>/carrier_up_count` ;
- **carrier_down_count** - nombre de fois où le lien est devenu down, lu depuis `/sys/class/net/<name>/carrier_down_count` ;
- **in** - trafic entrant, lu depuis `/proc/net/dev`. Contient les compteurs de l'interface :
  - **bytes** - nombre d'octets valides ;
  - **packets** - nombre de paquets valides ;
  - **errors** - nombre total de paquets erronés ;
  - **dropped** - nombre de paquets reçus mais non traités + nombre de paquets manqués par l'hôte ;
  - **overruns** - nombre d'événements de dépassement ;
  - **frame** - nombre de paquets avec erreurs de trame ;
  - **compressed** - nombre de paquets compressés correctement reçus ;
  - **multicast** - paquets multicast reçus.
- **out** - trafic sortant, lu depuis `/proc/net/dev`. Contient les compteurs de l'interface :
  - **bytes** - nombre d'octets valides ;
  - **packets** - nombre de paquets valides ;
  - **errors** - nombre total de paquets erronés ;
  - **dropped** - nombre de paquets reçus mais non traités + nombre de paquets manqués par l'hôte ;
  - **overruns** - nombre d'événements de dépassement ;
  - **collisions** - nombre de collisions pendant les transmissions de paquets ;
  - **carrier** - nombre d'erreurs de transmission de trame dues à une perte de porteuse pendant la transmission ;
  - **compressed** - nombre de paquets compressés transmis.

Les métriques non disponibles sont exclues de la sortie JSON. Dans l'exemple suivant, aucun alias n'est configuré pour l'interface "lo", la propriété `ifalias` est donc exclue.

Exemple :


```
{
    "config": [
        {
            "name": "lo",
            "mac": "00:00:00:00:00:00",
            "type": "loopback",
            "administrative_state": "up",
            "operational_state": "unknown",
            "carrier": 1
        },
        {
            "name": "enp3s0",
            "ifalias": "Wired Ethernet connection",
            "mac": "8c:16:45:e3:1d:58",
            "type": "physical",
            "duplex": "unknown",
            "administrative_state": "up",
            "operational_state": "down",
            "carrier": 0
        },
        {
            "name": "wlp5s0",
```

```
            "ifalias": "Wi-Fi network adapter",
            "mac": "34:e1:2d:72:67:c6",
            "type": "physical",
            "administrative_state": "up",
            "operational_state": "up",
            "carrier": 1
        }
    ],
    "values": [
        {
            "name": "lo",
            "mac": "00:00:00:00:00:00",
            "carrier": 1,
            "carrier_changes": 0,
            "carrier_up_count": 0,
            "carrier_down_count": 0,
            "in": {
                "bytes": 1040047,
                "packets": 10784,
                "errors": 0,
                "dropped": 0,
                "overruns": 0,
                "frame": 0,
                "compressed": 0,
                "multicast": 0
            },
            "out": {
                "bytes": 1040047,
                "packets": 10784,
                "errors": 0,
                "dropped": 0,
                "overruns": 0,
                "collisions": 0,
                "carrier": 0,
                "compressed": 0
            }
        },
        {
            "name": "enp3s0",
            "ifalias": "Wired Ethernet connection",
            "mac": "8c:16:45:e3:1d:58",
            "carrier": 0,
            "carrier_changes": 1,
            "carrier_up_count": 0,
            "carrier_down_count": 1,
            "in": {
                "bytes": 0,
                "packets": 0,
                "errors": 0,
                "dropped": 0,
                "overruns": 0,
                "frame": 0,
                "compressed": 0,
                "multicast": 0
            },
            "out": {
                "bytes": 0,
                "packets": 0,
                "errors": 0,
                "dropped": 0,
                "overruns": 0,
                "collisions": 0,
```

```
                "carrier": 0,
                "compressed": 0
            }
        },
        {
            "name": "wlp5s0",
            "ifalias": "Wi-Fi network adapter",
            "mac": "34:e1:2d:72:67:c6",
            "carrier": 1,
            "carrier_changes": 6,
            "carrier_up_count": 3,
            "carrier_down_count": 3,
            "in": {
                "bytes": 165537298,
                "packets": 216182,
                "errors": 0,
                "dropped": 291,
                "overruns": 0,
                "frame": 0,
                "compressed": 0,
                "multicast": 0
            },
            "out": {
                "bytes": 252574347,
                "packets": 246607,
                "errors": 0,
                "dropped": 0,
                "overruns": 0,
                "collisions": 0,
                "carrier": 0,
                "compressed": 0
            }
        }
    ]
}
```


Voir aussi

Pour plus d'informations sur l'ABI sysfs pour les périphériques réseau, voir :

- Documentation du noyau Linux

Valeurs de retour pour vfs.dev.get

Vue d'ensemble

Cette section fournit des détails sur la valeur de retour de l'élément `vfs.dev.get`. L'élément est pris en charge sous Linux par Zabbix agent et Zabbix agent 2, et sous AIX par Zabbix agent.

Détails

Cet élément renvoie des informations détaillées sur les périphériques de bloc dont les noms correspondent à l'expression régulière donnée.

La sortie est structurée sous forme d'objet JSON contenant `config` et, selon le paramètre <mode>, des tableaux `values`.

Le tableau `config` contient les propriétés disponibles pour le mode sélectionné. Le tableau `values` n'est renvoyé que pour les modes qui fournissent des statistiques sur les périphériques de bloc.

Les métriques indisponibles sont exclues de la sortie JSON.

Les sections suivantes décrivent les propriétés renvoyées pour chaque mode.

mode disques

`vfs.dev.get[disks,.*]` renvoie les données de configuration des périphériques de bloc.

Propriétés :

- **name** - nom du périphérique de bloc ;
- **devid** - identifiant du périphérique de bloc, par exemple un ID de périphérique basé sur WWN ou généré par le noyau ;
- **type** - type de périphérique de bloc (`disk`, `partition`, `rom`) ;
- **path** - chemin vers le nœud du périphérique de bloc ;
- **model** - modèle du périphérique ;
- **serial** - numéro de série du périphérique ;
- **wwn** - World Wide Name (WWN) du périphérique ;
- **size_bytes** - taille du périphérique en octets ;
- **logical_block_size** - taille logique de bloc en octets ;
- **physical_block_size** - taille physique de bloc en octets.

Exemple pour Linux :


```
{
  "config": [
    {
      "name": "sda",
      "devid": "ata-ST2000LM015-2E8174_ZDZ3YHZY",
      "type": "disk",
      "path": "/dev/sda",
      "model": " ST2000LM015-2E81",
      "wwn": "naa.5000c500b2791423",
      "size_bytes": 2000398934016,
      "logical_block_size": 512,
      "physical_block_size": 4096
    },
    {
      "name": "sdb",
      "devid": "usb-Kingston_DataTraveler_3.0_E0D55E6C382C15A158991CBD-0:0",
      "type": "disk",
      "path": "/dev/sdb",
      "model": "DataTraveler 3.0",
      "size_bytes": 61991813632,
      "logical_block_size": 512,
      "physical_block_size": 512
    },
    {
      "name": "nvme0n1",
      "devid": "nvme-SAMSUNG_MZVLW256HEHP-000L7_S35ENX1K753633",
      "type": "disk",
      "path": "/dev/nvme0n1",
      "model": "SAMSUNG MZVLW256HEHP-000L7",
      "serial": "S35ENX1K753633",
      "wwn": "eui.002538b781b7dc55",
      "size_bytes": 256060514304,
      "logical_block_size": 512,
      "physical_block_size": 512
    }
  ]
}
```


Exemple pour AIX :


```
{
  "config": [
    {
      "name": "cd0",
      "type": "disk",
      "path": "/dev/cd0",
      "size_bytes": 0,
      "logical_block_size": 512
    },
    {
      "name": "hdisk1",
      "type": "disk",
```

```
      "devid": "48333321360050768OC818022F000000000015EA404214503IBMfcp05VDASD03AIXvscsi",
      "path": "/dev/hdisk1",
      "size_bytes": 51136000000,
      "logical_block_size": 512
    },
    {
      "name": "hdisk0",
      "type": "disk",
      "devid": "48333321360050768OC818022F000000000015EA304214503IBMfcp05VDASD03AIXvscsi",
      "path": "/dev/hdisk0",
      "size_bytes": 0,
      "logical_block_size": 512
    }
  ]
}
```


mode périphériques

`vfs.dev.get[devices,.*]` renvoie les données de configuration des périphériques de bloc, y compris les partitions.

Propriétés :

- **name** - nom du périphérique de bloc.
- **devid** - identifiant du périphérique de bloc.
- **type** - type du périphérique de bloc (`disk`, `partition`, `rom`).
- **partitions** - partitions sur le périphérique. Contient les noms des partitions et leur taille en octets.

Exemple pour Linux :


```
{
  "config": [
    {
      "name": "sda",
      "devid": "ata-ST2000LM015-2E8174_ZDZ3YHZY",
      "type": "disk",
      "partitions": {
        "sda1": 16759808,
        "sda2": 2000381018112
      }
    },
    {
      "name": "sdb",
      "devid": "usb-Kingston_DataTraveler_3.0_E0D55E6C382C15A158991CBD-0:0",
      "type": "disk",
      "partitions": {
        "sdb1": 27631878144,
        "sdb2": 34359768576,
        "sdb3": 32256
      }
    },
    {
      "name": "nvme0n1",
      "devid": "nvme-SAMSUNG_MZVLW256HEHP-000L7_S35ENX1K753633",
      "type": "disk",
      "partitions": {
        "nvme0n1p1": 209715200,
        "nvme0n1p2": 16777216,
        "nvme0n1p3": 255060869120,
        "nvme0n1p4": 770703360
      }
    }
  ]
}
```


mode disk_stats

`vfs.dev.get[disk_stats,.*]` renvoie les données de configuration des périphériques de bloc et les statistiques de disque.

Propriétés :

- **name** - nom du périphérique de bloc.
- **devid** - identifiant du périphérique de bloc.
- **type** - type de périphérique de bloc (`disk`, `partition`, `rom`).
- **size_bytes** - taille du périphérique en octets.
- **logical_block_size** - taille logique du bloc en octets.
- **stats** - statistiques d'E/S du périphérique de bloc. Contient :
  - **reads_completed** - nombre d'opérations de lecture terminées ;
  - **writes_completed** - nombre d'opérations d'écriture terminées ;
  - **bytes_read** - nombre d'octets lus ;
  - **bytes_written** - nombre d'octets écrits ;
  - **io_time_ms** - temps total passé à effectuer des E/S, en millisecondes.

Exemple pour Linux


```
{
  "config": [
    {
      "name": "sda",
      "devid": "ata-ST2000LM015-2E8174_ZDZ3YHZY",
      "type": "disk",
      "size_bytes": 61991813632
    },
    {
      "name": "sdb",
      "devid": "usb-Kingston_DataTraveler_3.0_E0D55E6C382C15A158991CBD-0:0",
      "type": "disk",
      "size_bytes": 61991813632
    },
    {
      "name": "nvme0n1",
      "devid": "nvme-SAMSUNG_MZVLW256HEHP-000L7_S35ENX1K753633",
      "type": "disk",
      "size_bytes": 256060514304
    }
  ],
  "values": [
    {
      "name": "sda",
      "stats": {
        "reads_completed": 871,
        "writes_completed": 0,
        "bytes_read": 13045760,
        "bytes_written": 0,
        "io_time_ms": 1256
      }
    },
    {
      "name": "sdb",
      "stats": {
        "reads_completed": 51017,
        "writes_completed": 10682,
        "bytes_read": 10514627,
        "bytes_written": 1530472,
        "io_time_ms": 284988
      }
    },
    {
      "name": "nvme0n1",
      "stats": {
        "reads_completed": 1259,
        "writes_completed": 7,
```

```
        "bytes_read": 49698,
        "bytes_written": 6,
        "io_time_ms": 243
      }
    }
  ]
}
```


Exemple pour AIX :


```
{
  "config": [
    {
      "name": "cd0",
      "type": "disk",
      "path": "/dev/cd0",
      "size_bytes": 0,
      "logical_block_size": 2048
    },
    {
      "name": "hdisk1",
      "type": "disk",
      "devid": "48333321360050768OC818022F000000000015EA404214503IBMfcp05VDASD03AIXvscsi",
      "path": "/dev/hdisk1",
      "size_bytes": 51136000000,
      "logical_block_size": 512
    },
    {
      "name": "hdisk0",
      "type": "disk",
      "devid": "48333321360050768OC818022F000000000015EA304214503IBMfcp05VDASD03AIXvscsi",
      "path": "/dev/hdisk0",
      "size_bytes": 0,
      "logical_block_size": 512
    }
  ],
  "values": [
    {
      "name": "cd0",
      "stats": {
        "reads_completed": 21,
        "writes_completed": 0,
        "bytes_read": 75776,
        "bytes_written": 0,
        "io_time_ms": 0
      }
    },
    {
      "name": "hdisk1",
      "stats": {
        "reads_completed": 35888,
        "writes_completed": 1165147,
        "bytes_read": 593699840,
        "bytes_written": 10009646592,
        "io_time_ms": 69
      }
    },
    {
      "name": "hdisk0",
      "stats": {
        "reads_completed": 3,
        "writes_completed": 0,
        "bytes_read": 12800,
```

```
        "bytes_written": 0,
        "io_time_ms": 0
      }
    }
  ]
}
```

mode device_stats

`vfs.dev.get[device_stats,.*]` renvoie les données de configuration des périphériques de bloc et les statistiques des périphériques.

Propriétés :

- **name** - nom du périphérique de bloc.
- **devid** - identifiant du périphérique de bloc.
- **type** - type de périphérique de bloc (`disk`, `partition`, `rom`).
- **size_bytes** - taille du périphérique en octets.
- **stats** - statistiques d'E/S du périphérique de bloc. Contient :
  - **reads_completed** - nombre d'opérations de lecture terminées ;
  - **writes_completed** - nombre d'opérations d'écriture terminées ;
  - **bytes_read** - nombre d'octets lus ;
  - **bytes_written** - nombre d'octets écrits ;
  - **io_time_ms** - temps total passé à effectuer des E/S, en millisecondes.

Exemple pour Linux :

```
{
  "config": [
    {
      "name": "sda",
      "devid": "ata-ST2000LM015-2E8174_ZDZ3YHZY",
      "type": "disk",
      "size_bytes": 61991813632
    },
    {
      "name": "sda1",
      "type": "partition",
      "devid": "ata-ST2000LM015-2E8174_ZDZ3YHZY-part1",
      "size_bytes": 1234567890
    },
    {
      "name": "sda2",
      "type": "partition",
      "devid": "ata-ST2000LM015-2E8174_ZDZ3YHZY-part2",
      "size_bytes": 1234567890
    },
    {
      "name": "sdb",
      "devid": "usb-Kingston_DataTraveler_3.0_E0D55E6C382C15A158991CBD-0:0",
      "type": "disk",
      "size_bytes": 61991813632
    },
    {
      "name": "sdb1",
      "type": "partition",
      "devid": "usb-Kingston_DataTraveler_3.0_E0D55E6C382C15A158991CBD-0:0-part1",
      "size_bytes": 1234567890
    },
    {
      "name": "sdb2",
      "type": "partition",
      "devid": "usb-Kingston_DataTraveler_3.0_E0D55E6C382C15A158991CBD-0:0-part2",
      "size_bytes": 1234567890
    },
```

```
    {
      "name": "sdb3",
      "type": "partition",
      "devid": "usb-Kingston_DataTraveler_3.0_E0D55E6C382C15A158991CBD-0:0-part3",
      "size_bytes": 1234567890
    },
    {
      "name": "nvme0n1",
      "devid": "nvme-SAMSUNG_MZVLW256HEHP-000L7_S35ENX1K753633",
      "type": "disk",
      "size_bytes": 256060514304
    },
    {
      "name": "nvme0n1p1",
      "type": "partition",
      "devid": "nvme-SAMSUNG_MZVLW256HEHP-000L7_S35ENX1K753633-part1",
      "size_bytes": 1234567890
    },
    {
      "name": "nvme0n1p2",
      "type": "partition",
      "devid": "nvme-SAMSUNG_MZVLW256HEHP-000L7_S35ENX1K753633-part2",
      "size_bytes": 1234567890
    },
    {
      "name": "nvme0n1p3",
      "type": "partition",
      "devid": "nvme-SAMSUNG_MZVLW256HEHP-000L7_S35ENX1K753633-part3",
      "size_bytes": 1234567890
    },
    {
      "name": "nvme0n1p4",
      "type": "partition",
      "devid": "nvme-SAMSUNG_MZVLW256HEHP-000L7_S35ENX1K753633-part4",
      "size_bytes": 1234567890
    }
  ],
  "values": [
    {
      "name": "sda",
      "stats": {
        "reads_completed": 871,
        "writes_completed": 0,
        "bytes_read": 13045760,
        "bytes_written": 0,
        "io_time_ms": 1256
      }
    },
    {
      "name": "sda1",
      "stats": {
        "reads_completed": 123,
        "writes_completed": 123,
        "bytes_read": 123,
        "bytes_written": 123,
        "io_time_ms": 123
      }
    },
    {
      "name": "sda2",
      "stats": {
        "reads_completed": 123,
```

```
      "writes_completed": 123,
      "bytes_read": 123,
      "bytes_written": 123,
      "io_time_ms": 123
    }
  },
  {
    "name": "sdb",
    "stats": {
      "reads_completed": 51017,
      "writes_completed": 10682,
      "bytes_read": 10514627,
      "bytes_written": 1530472,
      "io_time_ms": 284988
    }
  },
  {
    "name": "sdb1",
    "stats": {
      "reads_completed": 123,
      "writes_completed": 123,
      "bytes_read": 123,
      "bytes_written": 123,
      "io_time_ms": 123
    }
  },
  {
    "name": "sdb2",
    "stats": {
      "reads_completed": 123,
      "writes_completed": 123,
      "bytes_read": 123,
      "bytes_written": 123,
      "io_time_ms": 123
    }
  },
  {
    "name": "sdb3",
    "stats": {
      "reads_completed": 123,
      "writes_completed": 123,
      "bytes_read": 123,
      "bytes_written": 123,
      "io_time_ms": 123
    }
  },
  {
    "name": "nvme0n1",
    "stats": {
      "reads_completed": 1259,
      "writes_completed": 7,
      "bytes_read": 49698,
      "bytes_written": 6,
      "io_time_ms": 243
    }
  },
  {
    "name": "nvme0n1p1",
    "stats": {
      "reads_completed": 123,
      "writes_completed": 123,
      "bytes_read": 123,
```

```
        "bytes_written": 123,
        "io_time_ms": 123
      }
    },
    {
      "name": "nvme0n1p2",
      "stats": {
        "reads_completed": 123,
        "writes_completed": 123,
        "bytes_read": 123,
        "bytes_written": 123,
        "io_time_ms": 123
      }
    },
    {
      "name": "nvme0n1p3",
      "stats": {
        "reads_completed": 123,
        "writes_completed": 123,
        "bytes_read": 123,
        "bytes_written": 123,
        "io_time_ms": 123
      }
    },
    {
      "name": "nvme0n1p4",
      "stats": {
        "reads_completed": 123,
        "writes_completed": 123,
        "bytes_read": 123,
        "bytes_written": 123,
        "io_time_ms": 123
      }
    }
  ]
}
```


Notes sur les éléments system.cpu.util sous Windows

L'élément system.cpu.util fournit le pourcentage d'utilisation du processeur.

Lorsqu'un processus collecteur est démarré sur l'agent Zabbix pour Windows, un tampon pour N éléments pour N processeurs logiques (threads) est initialisé. Les valeurs sont mises à jour périodiquement dans le tampon à l'aide des compteurs de performance Windows. Ces valeurs sont renvoyées lorsque `system.cpu.util[n]` est demandé, où 0 <= n < N.

Un élément supplémentaire dans le tampon est utilisé pour `system.cpu.util[total]`.

Zabbix sélectionne automatiquement différents compteurs de performance afin de prendre en charge à la fois les systèmes NUMA et les anciennes versions de Windows sans prise en charge des groupes de processeurs.

Si le nombre de processeurs logiques (threads) est inférieur ou égal à 64 et que le nombre de groupes de processeurs est égal à 1, alors :

```
N est le nombre de processeurs logiques (threads).

\Processor(_Total)\% Processor Time
\Processor(0)\% Processor Time
\Processor(1)\% Processor Time
\Processor(2)\% Processor Time
...
\Processor(N-1)\% Processor Time
```

Sinon :

```
G est le nombre de groupes de processeurs et N est le nombre de processeurs logiques (threads) dans le gro

\Processor Information(_Total)\% Processor Time
\Processor Information(0,0)\% Processor Time
\Processor Information(0,1)\% Processor Time
\Processor Information(0,2)\% Processor Time
...
\Processor Information(0,N-1)\% Processor Time
...
\Processor Information(G-1,0)\% Processor Time
\Processor Information(G-1,1)\% Processor Time
\Processor Information(G-1,2)\% Processor Time
...
\Processor Information(G-1,N-1)\% Processor Time
```

Prise en charge des grandes valeurs JSON

Pour garantir une gestion fiable des grandes valeurs avec le type de données JSON élément (1 MiB ou plus), examinez et appliquez les modifications de configuration suivantes selon les besoins :

1. Si vous utilisez MySQL ou MariaDB, augmentez la taille maximale autorisée des paquets dans `/etc/mysql/my.cnf` (elle peut être définie par défaut à 1 MB sur certains systèmes) :

```
[mysqld]
max_allowed_packet=128M

[mysqldump]
max_allowed_packet=1G
```

2. Ajustez, si nécessaire, d'autres variables système du serveur MySQL/MariaDB et les variables système InnoDB. Par exemple :

```
innodb_io_capacity=1900
connect_timeout=600
wait_timeout=57600
interactive_timeout=57600
net_read_timeout=7200
net_write_timeout=7200
net-buffer-length=32704
```

3. Augmentez `HistoryCacheSize` à 2G dans le fichier de configuration du serveur Zabbix.

4. Si vous utilisez un proxy Zabbix, augmentez `HistoryCacheSize` à 2G et `ProxyMemoryBufferSize` à 2G dans le fichier de configuration du proxy.

5. Si vous utilisez l'agent Zabbix, augmentez les délais de communication entre le serveur/proxy Zabbix et l'agent (par exemple, les délais d'attente des éléments ou la valeur du paramètre `Timeout` dans les fichiers de configuration du serveur/proxy Zabbix et de l'agent). Sinon, les messages entre eux risquent de ne pas aboutir, et vous pouvez voir des erreurs telles que *"message length does not match expected length"*. Vérifiez également, si nécessaire, d'autres délais d'attente, comme le délai d'attente du script ou le délai d'attente du test d'élément.

Fonctions prises en charge

Voir Expression de déclencheur et fonctions.

Macros

Il est possible d'utiliser les macros prises en charge prêtes à l'emploi et les macros utilisateur prises en charge par emplacement.

Macros prises en charge par l'emplacement

Vue d'ensemble

Le tableau macros prises en charge par emplacement contient une référence de toutes les macros (intégrées, d'expression et définissables par l'utilisateur) prises en charge à l'emplacement spécifique.

Si toutes les macros d'un groupe associé sont prises en charge, seul le nom du groupe est indiqué. Cliquez sur chaque nom de macro ou nom de groupe pour voir les détails complets des macros.

La liste détails des macros contient une liste détaillée de toutes les macros **intégrées**, regroupées par domaine d'application. Notez que pour personnaliser les valeurs des macros (par exemple, les raccourcir ou en extraire des sous-chaînes spécifiques), vous pouvez utiliser les fonctions de macro.

Macros prises en charge selon l'emplacement

Emplacement	Macros prises en charge	Commentaires
Action		
Condition de période, Durée de l'opération par défaut, Durée de l'étape	Macros définissables par l'utilisateur (globales uniquement)	Une seule macro doit remplir tout le champ ; plusieurs macros ou un mélange avec du texte ne sont pas pris en charge
Notifications et commandes		
Basé sur un déclencheur	Macros Action Macros Date/heure {ESC.HISTORY} Macros Événement, macros Événement de cause/symptôme Macros de mise à jour d'événement: {EVENT.UPDATE.HISTORY}, {EVENT.UPDATE.STATUS} Macro d'expression: {?EXPRESSION} {FUNCTION.VALUE} Macros d'hôte: {HOST.CONN}, {HOST.DESCRIPTION}, {HOST.DNS}, {HOST.HOST}, {HOST.ID}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Macros hôte cible (commandes uniquement) Macros inventaire d'hôte Macros élément (sauf {ITEM.STATE}, {ITEM.STATE.ERROR}) Macros proxy Macros de déclencheur: {TRIGGER.DESCRIPTION}, {TRIGGER.EVENTS.*}, {TRIGGER.EXPRESSION}, {TRIGGER.EXPRESSION.EXPLAIN}, {TRIGGER.EXPRESSION.RECOVERY}, {TRIGGER.ID}, {TRIGGER.HOSTGROUP.NAME}, {TRIGGER.NAME}, {TRIGGER.NAME.ORIG}, {TRIGGER.NSEVERITY}, {TRIGGER.SEVERITY}, {TRIGGER.STATUS}, {TRIGGER.TEMPLATE.NAME}, {TRIGGER.URL}, {TRIGGER.URL.NAME}, {TRIGGER.VALUE} Macros définissables par l'utilisateur	Voir aussi: Utilisation des macros dans les messages

Emplacement	Macros prises en charge	Commentaires
Mise à jour du problème	Macros Action Macros Date/heure {ESC.HISTORY} Macros Événement, macros Événement de cause/symptôme, macros Événement de récupération Macros Mise à jour d'événement: toutes (sauf {EVENT.UPDATE.NSEVERITY}, {EVENT.UPDATE.SEVERITY}) Macro d'expression: {?EXPRESSION} Macros Fonction Macros d'hôte: {HOST.CONN}, {HOST.DESCRIPTION}, {HOST.DNS}, {HOST.HOST}, {HOST.ID}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Macros hôte cible (commandes uniquement) Macros inventaire d'hôte Macros élément (sauf {ITEM.STATE}, {ITEM.STATE.ERROR}) Macros proxy Macros de déclencheur: {TRIGGER.DESCRIPTION}, {TRIGGER.EVENTS.*}, {TRIGGER.EXPRESSION}, {TRIGGER.EXPRESSION.EXPLAIN}, {TRIGGER.EXPRESSION.RECOVERY}, {TRIGGER.ID}, {TRIGGER.HOSTGROUP.NAME}, {TRIGGER.NAME}, {TRIGGER.NAME.ORIG}, {TRIGGER.NSEVERITY}, {TRIGGER.SEVERITY}, {TRIGGER.STATUS}, {TRIGGER.TEMPLATE.NAME}, {TRIGGER.URL}, {TRIGGER.URL.NAME}, {TRIGGER.VALUE} {USER.FULLNAME} Macros définissables par l'utilisateur	
Récupération du problème	Macros Événement de récupération Macros Événement {FUNCTION.RECOVERY.VALUE} {TRIGGER.EXPRESSION.RECOVERY.EXPLAIN}	
Découverte	Macros Action Macros Date/heure Macros Découverte Macros d'événement: {EVENT.AGE}, {EVENT.DATE}, {EVENT.ID}, {EVENT.OBJECT}, {EVENT.SOURCE}, {EVENT.TIME}, {EVENT.TIMESTAMP} Macros hôte cible (commandes uniquement) Macros proxy	
Auto-enregistrement	Macros Action Macros Date/heure Macros d'événement: {EVENT.AGE}, {EVENT.DATE}, {EVENT.ID}, {EVENT.OBJECT}, {EVENT.SOURCE}, {EVENT.TIME}, {EVENT.TIMESTAMP} Macros d'hôte: {HOST.HOST}, {HOST.IP}, {HOST.PORT}, {HOST.METADATA} Macros hôte cible (commandes uniquement) Macros proxy	
Basé sur un service	Macros Action Macros Date/heure {ESC.HISTORY} Macros d'événement: toutes sauf {EVENT.ACK.STATUS}, {EVENT.OPDATA} Macros Service Macros définissables par l'utilisateur	
Récupération de service	Macros Événement (sauf {EVENT.ACK.STATUS}, {EVENT.OPDATA}), macros Événement de récupération	

Emplacement	Macros prises en charge	Commentaires
Mise à jour de service	Macros Action Macros Date/heure {ESC.HISTORY} Macros Événement (sauf {EVENT.ACK.STATUS}, {EVENT.OPDATA}) Macros de mise à jour d'événement: {EVENT.UPDATE.DATE}, {EVENT.UPDATE.NSEVERITY}, {EVENT.UPDATE.SEVERITY}, {EVENT.UPDATE.STATUS}, {EVENT.UPDATE.TIME}, {EVENT.UPDATE.TIMESTAMP} Macros Service Macros définissables par l'utilisateur	
Notifications internes	Action macros Date/time macros {ESC.HISTORY} Événement macros (sauf {EVENT.NSEVERITY}, {EVENT.SEVERITY}), Événement de récupération macros Macros d'hôte: {HOST.CONN}, {HOST.DESCRIPTION}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Macros inventaire d'hôte Macros d'élément: {ITEM.DESCRIPTION}, {ITEM.DESCRIPTION.ORIG}, {ITEM.ID}, {ITEM.KEY}, {ITEM.KEY.ORIG}, {ITEM.NAME}, {ITEM.NAME.ORIG}, {ITEM.VALUETYPE} Macros proxy	
Basé sur un élément	Macros d'élément: {ITEM.STATE}, {ITEM.STATE.ERROR}	
Basé sur une règle LLD	Macros Notification de découverte de bas niveau	
Basé sur un déclencheur	Macros de déclencheur: {TRIGGER.DESCRIPTION}, {TRIGGER.EXPRESSION}, {TRIGGER.EXPRESSION.RECOVERY}, {TRIGGER.HOSTGROUP.NAME}, {TRIGGER.ID}, {TRIGGER.NAME}, {TRIGGER.NAME.ORIG}, {TRIGGER.NSEVERITY}, {TRIGGER.SEVERITY}, {TRIGGER.STATE}, {TRIGGER.STATE.ERROR}, {TRIGGER.TEMPLATE.NAME}, {TRIGGER.URL}, {TRIGGER.URL.NAME} Macros définissables par l'utilisateur	
Paramètres de script d'alerte	Macros Alerte Macros définissables par l'utilisateur (globales uniquement)	
Connecteur		
Champ URL, Nom d'utilisateur, Mot de passe, Jeton Bearer, proxy HTTP, certificat SSL, fichier de clé SSL, mot de passe de clé SSL	Macros définissables par l'utilisateur (globales uniquement)	
Champ Délai d'attente	Macros définissables par l'utilisateur (globales uniquement)	Une seule macro doit remplir tout le champ ; plusieurs macros ou un mélange avec du texte ne sont pas pris en charge.

Emplacement	Macros prises en charge	Commentaires
Nom du graphique	Macro d'expression: {?EXPRESSION}	Dans cet emplacement, seule une fonction **avg**, **last**, **max** ou **min** avec des secondes comme paramètre est prise en charge dans cette macro (des suffixes de temps peuvent être utilisés). La macro {HOST.HOST<1-9>} peut être utilisée comme hôte dans la macro (voir aussi macros indexées). Exemple d'utilisation: {?avg(/{HOST.HOST}/item.key,1h)} Utilisation non prise en charge: {?last(/host/item1)/last(/host/item2)} {?last(/host/item1)*10} {?count(/host/item1,5m)}
Hôte, prototype d'hôte		
IP/DNS de l'interface	Macros d'hôte: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME} Macros définissables par l'utilisateur [1]	[1] Pour l'adresse IP, une seule macro doit remplir tout le champ ; plusieurs macros ou un mélange avec du texte ne sont pas pris en charge
Port de l'interface	Macros définissables par l'utilisateur	Une seule macro doit remplir tout le champ ; plusieurs macros ou un mélange avec du texte ne sont pas pris en charge.
Communauté SNMP *Nom du contexte SNMPv3, Nom de sécurité, Phrase secrète d'authentification, Phrase secrète de confidentialité* *Nom d'utilisateur IPMI, mot de passe*	Macros définissables par l'utilisateur	
Élément, prototype d'élément, règle LLD		
Nom	Macros définissables par l'utilisateur	Non pris en charge dans les noms de règles LLD.

Emplacement	Macros prises en charge	Commentaires
Description	Macros définissables par l'utilisateur	
Paramètres de clé	Macros d'hôte: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Macros définissables par l'utilisateur	Les macros {HOST.*} prises en charge seront résolues vers l'interface sélectionnée pour l'élément. Lorsqu'elles sont utilisées dans des éléments sans interface, elles seront résolues vers l'interface de l'agent Zabbix, SNMP, JMX ou IPMI de l'hôte dans cet ordre de priorité, ou vers 'UNKNOWN' si l'hôte ne possède aucune interface. Les macros ne sont pas prises en charge dans la clé d'élément des éléments Zabbix trapper.
Intervalle de mise à jour, Intervalles personnalisés, Délai d'attente (disponible pour les types d'élément pris en charge), Conserver jusqu'à (pour l'historique/les tendances) champs	Macros définissables par l'utilisateur	Une seule macro doit remplir tout le champ ; plusieurs macros ou un mélange avec du texte ne sont pas pris en charge.
Supprimer les ressources perdues, Désactiver les ressources perdues champs	Macros définissables par l'utilisateur	Ces champs sont pris en charge uniquement pour les règles LLD. Une seule macro doit remplir tout le champ ; plusieurs macros ou un mélange avec du texte ne sont pas pris en charge.
Paramètres de l'étape de prétraitement (y compris les scripts personnalisés), *paramètres personnalisés de gestion des erreurs* (champs *Définir la valeur sur* et *Définir l'erreur sur*)	Macros définissables par l'utilisateur	
Expressions régulières de filtre	Macros d'hôte: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Macros définissables par l'utilisateur	Ces champs sont pris en charge uniquement pour les règles LLD.
Expressions régulières de filtre de remplacement	Macros définissables par l'utilisateur	Ces champs sont pris en charge uniquement pour les règles LLD.

Emplacement	Macros prises en charge	Commentaires
Opérations de remplacement: *Intervalle de mise à jour*, *Période de stockage de l'historique*, *Période de stockage des tendances* (pour les prototypes d'élément)	Macros définissables par l'utilisateur	Ces champs sont pris en charge uniquement pour les règles LLD. Une seule macro doit remplir tout le champ ; plusieurs macros ou un mélange avec du texte ne sont pas pris en charge.
Types d'élément		
Browser élément *noms et valeurs des paramètres*	Macros d'hôte: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Macros inventaire d'hôte Macros d'élément: {ITEM.ID}, {ITEM.KEY}, {ITEM.KEY.ORIG} Macros définissables par l'utilisateur	
Script élément *noms et valeurs des paramètres*		
Calculé/agrégé formule d'élément (*constantes d'expression*, *paramètres de fonction*, *paramètres de clé d'élément*, (élément agrégé uniquement) *conditions de filtre* (nom de groupe d'hôtes et nom de balise))	Macros définissables par l'utilisateur	
Database monitor élément *champs Nom d'utilisateur, Mot de passe, Requête SQL*	Macros définissables par l'utilisateur	
HTTP agent élément *champs URL, champs Requête, Corps de la requête, En-têtes (noms et valeurs), fichier de certificat SSL, fichier de clé SSL*	Macros d'hôte: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Macros d'élément: {ITEM.ID}, {ITEM.KEY}, {ITEM.KEY.ORIG} Macros définissables par l'utilisateur	Les URL contenant une macro secrète ne fonctionneront pas, car la macro qu'elles contiennent sera résolue en "******"
HTTP agent élément *champ Proxy HTTP*	Macros d'élément: {ITEM.ID}, {ITEM.KEY}, {ITEM.KEY.ORIG} Macros définissables par l'utilisateur	
HTTP agent élément *champs Codes d'état requis, Nom d'utilisateur/mot de passe d'authentification HTTP, mot de passe de clé SSL*	Macros définissables par l'utilisateur	
HTTP agent élément *champ Hôtes autorisés*	Macros d'hôte: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME} Macros définissables par l'utilisateur	
JMX agent élément *champ Point de terminaison*	Macros d'hôte: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.PORT} Macros définissables par l'utilisateur	

Emplacement	Macros prises en charge	Commentaires
Trapper élément *champ Hôtes autorisés*	Macros d'hôte: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP} Macros définissables par l'utilisateur	
SNMP agent élément *champ SNMP OID*	Macros définissables par l'utilisateur	
SSH and Telnet élément *script*	Macros d'hôte: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Macros définissables par l'utilisateur	
SSH agent élément *champs Nom d'utilisateur, Fichier de clé publique, Fichier de clé privée, Mot de passe*	Macros définissables par l'utilisateur	
Telnet agent élément *champs Nom d'utilisateur, Mot de passe*	Macros définissables par l'utilisateur	
Carte		
URL, Nom de l'URL	Macros d'hôte: {HOST.CONN}, {HOST.DESCRIPTION}, {HOST.DNS}, {HOST.HOST}, {HOST.ID}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} {HOSTGROUP.ID} Macros inventaire d'hôte Macros Carte {TRIGGER.ID}	
Libellé de l'élément	Macro d'expression: {?EXPRESSION} [1] Macros d'hôte: {HOST.CONN}, {HOST.DESCRIPTION}, {HOST.DNS}, {HOST.HOST}, {HOST.ID}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} {HOSTGROUP.ID} Macros inventaire d'hôte Macros Carte Macros de déclencheur: {TRIGGER.EVENTS.*}, {TRIGGER.ID}, {TRIGGER.PROBLEM.EVENTS.*}, {TRIGGERS.ACK}, {TRIGGERS.PROBLEM.ACK}, {TRIGGERS.PROBLEM.UNACK}, {TRIGGERS.UNACK}	[1] Seule une fonction **avg**, **last**, **max** ou **min** avec des secondes comme paramètre est prise en charge dans cette macro dans cet emplacement. La macro {HOST.HOST<1-9>} peut être utilisée comme hôte dans la macro. Exemple d'utilisation: {?avg(/{HOST.HOST}/item.key,1h)} Utilisation non prise en charge: {?last(/host/item1)/last(/host/item2)} {?last(/host/item1)*10} {?count(/host/item1,5m)}
Libellé du lien	Macro d'expression: {?EXPRESSION} [1]	
Libellé de la forme *Champ texte dans les formes de carte*	Macro d'expression: {?EXPRESSION} [1] {MAP.NAME}	
Découverte réseau		
Champ Intervalle de mise à jour	Macros définissables par l'utilisateur (globales uniquement)	Une seule macro doit remplir tout le champ ; plusieurs macros ou un mélange avec du texte ne sont pas pris en charge.
champs Communauté SNMP, SNMP OID	Macros définissables par l'utilisateur (globales uniquement)	

Emplacement	Macros prises en charge	Commentaires
champs Nom du contexte SNMPv3, Nom de sécurité, Phrase secrète d'authentification, Phrase secrète de confidentialité		
Proxy		
Adresse pour les agents actifs > *Port* (si le proxy appartient à un groupe) Pour le mode proxy passif: champs interface *Adresse* et *Port* champs *Délais d'attente pour les types d'élément* en mode de remplacement	Macros définissables par l'utilisateur (globales uniquement)	Une seule macro doit remplir tout le champ ; plusieurs macros ou un mélange avec du texte ne sont pas pris en charge.
Groupe de proxy		
Période de basculement, Nombre minimum de proxies champs	Macros définissables par l'utilisateur (globales uniquement)	Une seule macro doit remplir tout le champ ; plusieurs macros ou un mélange avec du texte ne sont pas pris en charge.
Scripts		
Commandes, texte de confirmation (Type: script, **action d'hôte** manuelle)	Macros d'hôte: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.ID} [1], {HOST.IP}, {HOST.NAME}, {HOST.PORT} Macros inventaire d'hôte [2] {MANUALINPUT} Macros Username Macros définissables par l'utilisateur	[1] Texte de confirmation uniquement [2] Pris en charge pour Zabbix server et Zabbix proxy
Commandes (Type: script, **action d'événement** manuelle)	Macros Date/heure Macros Événement, macros Événement de cause/symptôme, macros Événement de récupération (si la récupération a eu lieu) Macros de mise à jour d'événement: {EVENT.UPDATE.HISTORY}, {EVENT.UPDATE.STATUS} {FUNCTION.VALUE} Macros d'hôte: {HOST.CONN}, {HOST.DESCRIPTION}, {HOST.DNS}, {HOST.HOST}, {HOST.ID}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Macros inventaire d'hôte Macros d'élément: (sauf {ITEM.STATE}, {ITEM.STATE.ERROR}) {MANUALINPUT} Macros proxy Macros de déclencheur: {TRIGGER.DESCRIPTION}, {TRIGGER.EVENTS.*}, {TRIGGER.EXPRESSION}, {TRIGGER.EXPRESSION.EXPLAIN}, {TRIGGER.EXPRESSION.RECOVERY}, {TRIGGER.EXPRESSION.RECOVERY.EXPLAIN}, {TRIGGER.HOSTGROUP.NAME}, {TRIGGER.ID}, {TRIGGER.NAME}, {TRIGGER.NAME.ORIG}, {TRIGGER.NSEVERITY}, {TRIGGER.SEVERITY}, {TRIGGER.STATUS}, {TRIGGER.TEMPLATE.NAME}, {TRIGGER.URL}, {TRIGGER.URL.NAME}, {TRIGGER.VALUE} Macros Username Macros définissables par l'utilisateur	

Emplacement	Macros prises en charge	Commentaires
Texte de confirmation (Type: script, **action d'événement** manuelle)	Macros d'événement: {EVENT.ID}, {EVENT.NAME}, {EVENT.NSEVERITY}, {EVENT.SEVERITY}, {EVENT.STATUS}, {EVENT.VALUE} Macros d'hôte: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.ID}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Macros inventaire d'hôte {MANUALINPUT} Macros Username Macros définissables par l'utilisateur	
URL, texte de confirmation (Type: URL, **action d'hôte/événement** manuelle)		
Nom et valeur de balise	Macros d'hôte: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.ID}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Macros inventaire d'hôte Macros d'élément: {ITEM.LASTVALUE*}, {ITEM.LOG.*}, {ITEM.VALUE*} Macros de déclencheur: {TRIGGER.ID} (valeur de la balise du déclencheur uniquement) Macros définissables par l'utilisateur	Les macros de nom et de valeur de balise sont résolues uniquement pendant le processus de génération de l'événement.
Déclencheur		
Nom	Macros d'hôte: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Macros d'élément: {ITEM.LASTVALUE*}, {ITEM.LOG.*}, {ITEM.VALUE*} Macros/références positionnelles: $1...$9 Macros définissables par l'utilisateur	
Nom de l'événement	Macros Date/heure: {TIME}, {TIMESTAMP} Macro d'expression: {?EXPRESSION} Macros Fonction Macros d'hôte: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Macros d'élément: {ITEM.LASTVALUE*}, {ITEM.LOG.*}, {ITEM.VALUE*} Macros de déclencheur: {TRIGGER.EXPRESSION.EXPLAIN}	
Données opérationnelles	Macros d'hôte: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Macros d'élément: {ITEM.LASTVALUE*}, {ITEM.LOG.*}, {ITEM.VALUE*} Macros définissables par l'utilisateur	
Expression	{TRIGGER.VALUE} Macros définissables par l'utilisateur [1]	[1] Uniquement dans les constantes et les paramètres de fonction ; les macros secrètes ne sont pas prises en charge
champs Balise pour correspondance, Nom de l'entrée de menu, URL de l'entrée de menu	Macros définissables par l'utilisateur	Les URL contenant une macro secrète ne fonctionneront pas, car la macro qu'elles contiennent sera résolue en "******".
Description	Macros d'hôte: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Macros d'élément: {ITEM.LASTVALUE*}, {ITEM.LOG.*}, {ITEM.VALUE*} Macros définissables par l'utilisateur	

Emplacement	Macros prises en charge	Commentaires
URL	{EVENT.ID} Macros d'hôte: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.ID}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Macros d'élément: {ITEM.LASTVALUE*}, {ITEM.LOG.*}, {ITEM.VALUE*} {TRIGGER.ID}	
Commandes de paramètre utilisateur	Macros/références positionnelles: $1...$9	
Scénario web		
champs Nom, Agent, proxy HTTP	Macros définissables par l'utilisateur	
Champ Intervalle de mise à jour	Macros définissables par l'utilisateur	Une seule macro doit remplir tout le champ ; plusieurs macros ou un mélange avec du texte ne sont pas pris en charge.
champs Variables, En-têtes, certificat SSL, fichier de clé SSL	Macros d'hôte: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Macros définissables par l'utilisateur	
champs Utilisateur, Mot de passe, mot de passe de clé SSL	Macros définissables par l'utilisateur	
Étape du scénario web *champs Nom, Variables (valeurs uniquement)*	Macros définissables par l'utilisateur	
Étape du scénario web *champs Délai d'attente, Codes d'état requis*	Macros définissables par l'utilisateur	Une seule macro doit remplir tout le champ ; plusieurs macros ou un mélange avec du texte ne sont pas pris en charge.
Étape du scénario web *champs URL, En-têtes (noms et valeurs), Chaîne requise*	Macros d'hôte: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Macros définissables par l'utilisateur	
Étape du scénario web *champ Post*	Macros d'hôte: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME}, {HOST.PORT}	
Type de média webhook		
Valeurs des paramètres	Macros Alerte Toutes les macros prises en charge dans les notifications de problème basées sur un déclencheur (voir ci-dessus)	Une seule macro doit remplir tout le champ ; plusieurs macros ou un mélange avec du texte ne sont pas pris en charge.
script JavaScript, nom du paramètre du script *JavaScript*	Macros définissables par l'utilisateur (globales uniquement)	
Nom de l'entrée de menu, URL de l'entrée de menu	{EVENT.TAGS.<tag name>}	

Emplacement	Macros prises en charge	Commentaires
Widgets		
Gauge widget *champ Description*	Macros d'hôte: {HOST.CONN}, {HOST.DESCRIPTION}, {HOST.DNS}, {HOST.HOST}, {HOST.ID}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Macros inventaire d'hôte Macros élément (sauf {ITEM.STATE.ERROR}) Macros définissables par l'utilisateur (globales uniquement)	
Honeycomb widget *champ Libellé principal/secondaire* **Item value** widget *champ Description*		
Top hosts widget *colonne Données textuelles*	Macros d'hôte: {HOST.CONN}, {HOST.DESCRIPTION}, {HOST.DNS}, {HOST.HOST}, {HOST.ID}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Macros inventaire d'hôte Macros définissables par l'utilisateur (globales uniquement)	
URL (Dynamic) widget *champ URL*	Macros d'hôte: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.ID}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Macros définissables par l'utilisateur	
colonne* Texte personnalisé		
Monitoring > Problèmes	Macros hôte (sauf {HOST.METADATA}) Macros inventaire d'hôte Macros élément (sauf {ITEM.STATE.ERROR}) Macros de déclencheur: {TRIGGER.ID} Macros d'événement: {EVENT.ID} Macros définissables par l'utilisateur	
Monitoring > Hôtes	Macros hôte (sauf {HOST.METADATA}) Macros inventaire d'hôte Macros définissables par l'utilisateur	
Monitoring > Dernières données	Macros hôte (sauf {HOST.METADATA}) Macros inventaire d'hôte Macros élément (sauf {ITEM.STATE.ERROR}) Macros définissables par l'utilisateur	
Collecte de données > Modèles	Macros d'hôte: {HOST.DESCRIPTION}, {HOST.HOST}, {HOST.ID}, {HOST.NAME}	
Collecte de données > Hôtes	Macros hôte (sauf {HOST.METADATA}) Macros inventaire d'hôte Macros définissables par l'utilisateur	
Autres emplacements		
Champ Temps de travail	Macros définissables par l'utilisateur (globales uniquement)	Une seule macro doit remplir tout le champ ; plusieurs macros ou un mélange avec du texte ne sont pas pris en charge.
champ Délais d'attente pour les types d'élément (paramètre global)		
Média utilisateur *champ Quand actif*	Macros définissables par l'utilisateur (globales uniquement)	

Emplacement	Macros prises en charge	Commentaires
Type de média Email *champs Nom d'utilisateur, Mot de passe*	Macros définissables par l'utilisateur (globales uniquement)	Les macros secrètes sont recommandées.
Rapports planifiés *champs Objet, Message*	Macros Date/heure: {TIME}, {TIMESTAMP}	

Détails des macros

Cette liste contient des détails sur chaque macro intégrée, regroupée par domaine d'application :

- Macros d'action
- Macros d'alerte
- Macros de date/heure
- Macros de découverte
- Macros d'événement
  - Macros de mise à jour d'événement
  - Macros d'événement cause/symptôme
  - Macros d'événement de récupération
- Macros d'escalade
- Macros de fonction
- Macros d'hôte
  - Macros de l'hôte cible
- Macros de groupe d'hôtes
- Macros d'inventaire d'hôte
- Macros d'élément
- Macros de notification de découverte de bas niveau
- Macros de carte
- Macros de proxy
- Macros de script
- Macros de service
- Macros de déclencheur
- Macros de nom d'utilisateur

Macros d'action

{ACTION.ID}

L'ID numérique de l'action déclenchée.

{ACTION.NAME}

Le nom de l'action déclenchée.

Macros d'alerte

{ALERT.MESSAGE}

La valeur du *message par défaut* provenant de la configuration de l'action.

{ALERT.SENDTO}

La valeur *Send to* de la configuration des médias utilisateur.

{ALERT.SUBJECT}

La valeur du *sujet par défaut* provenant de la configuration de l'action.

Macros de date et d'heure

{DATE}

La date actuelle au format `yyyy.mm.dd`.

{TIME}

L'heure actuelle au format `hh:mm:ss`.

{TIMESTAMP}

L’heure actuelle au format d’horodatage UNIX.

Macros de découverte

{DISCOVERY.DEVICE.IPADDRESS}

L’adresse IP du périphérique découvert.
 Toujours disponible, ne dépend pas de l’ajout de l’hôte.

{DISCOVERY.DEVICE.DNS}

Le nom DNS du périphérique découvert.
 Toujours disponible, ne dépend pas de l’ajout de l’hôte.

{DISCOVERY.DEVICE.STATUS}

Le statut du périphérique découvert (UP/DOWN).

{DISCOVERY.DEVICE.UPTIME}

Le temps écoulé depuis le dernier changement de statut de découverte pour un appareil particulier.
 Par exemple : 1h 29m 01s.
 Pour les appareils avec le statut DOWN, il s’agit de la durée de leur indisponibilité.

{DISCOVERY.RULE.NAME}

Le nom de la règle de découverte qui a détecté la présence ou l’absence du périphérique/service.

{DISCOVERY.SERVICE.NAME}

Le nom du service découvert.
 Par exemple : HTTP.

{DISCOVERY.SERVICE.PORT}

Le port du service découvert.
 Par exemple : 80.

{DISCOVERY.SERVICE.STATUS}

Le statut du service découvert (UP/DOWN).

{DISCOVERY.SERVICE.UPTIME}

Le temps écoulé depuis le dernier changement de statut de découverte pour un service particulier.
 Par exemple : 1h 29m 01s.
 Pour les services ayant le statut DOWN, il s’agit de la durée de leur indisponibilité.

Macros d’escalade

{ESC.HISTORY}

Le journal des notifications précédemment envoyées, leur étape d’escalade et leur statut (*envoyé*, *en cours* ou *échoué*).

Macros d’événement

{EVENT.ACK.STATUS}

Le statut d’acquittement de l’événement (Oui/Non).

{EVENT.AGE}

L’âge de l’événement qui a déclenché une action, avec une précision à la seconde près.
 Utile dans les messages d’escalade. Par exemple : 9m 13s.

{EVENT.DATE}

La date de l’événement qui a déclenché une action au format `yyyy.mm.dd`.
 Par exemple : 2025.04.14.

{EVENT.DURATION}

La durée de l’événement (différence de temps entre les événements de problème et de rétablissement), avec une précision à la seconde près.
 Utile dans les messages de rétablissement de problème.

{EVENT.ID}

L’identifiant numérique de l’événement qui a déclenché une action.

{EVENT.NAME}

Le nom de l’événement qui a déclenché une action.

{EVENT.NSEVERITY}

La valeur numérique de la gravité de l’événement.
 Valeurs possibles : *0* - Non classé, *1* - Information, *2* - Avertissement, *3* - Moyen, *4* - Élevé, *5* - Désastre.

{EVENT.OBJECT}

La valeur numérique de l'objet d'événement.
 Valeurs possibles : *0* - Déclencheur, *1* - Hôte découvert, *2* - Service découvert, *3* - Enregistrement automatique, *4* - Élément, *5* - Règle de découverte de bas niveau.

{EVENT.OPDATA}

Les données opérationnelles du déclencheur sous-jacent du problème.

{EVENT.SEVERITY}

Le nom de la gravité de l'événement.

{EVENT.SOURCE}

La valeur numérique de la source de l'événement.
 Valeurs possibles : *0* - Déclencheur, *1* - Découverte, *2* - Enregistrement automatique, *3* - Interne, *4* - Service.

{EVENT.STATUS}

La valeur textuelle de l'événement qui a déclenché une action.

{EVENT.TAGS}

Une liste de tags d'événement séparés par des virgules.
Développée en une chaîne vide si aucun tag n'existe.

{EVENT.TAGSJSON}

Un tableau JSON contenant des objets de balise d'événement.
Développé en tableau vide si aucune balise n'existe.

{EVENT.TAGS.<nom de balise>}

La valeur de la balise de l'événement référencée par le nom de la balise.
 Un nom de balise contenant des caractères non alphanumériques (y compris des caractères multioctets UTF non anglais) doit être entouré de guillemets doubles. Les guillemets et les barres obliques inverses à l'intérieur d'un nom de balise entre guillemets doivent être échappés avec une barre oblique inverse.

{EVENT.TIME}

L'heure de l'événement qui a déclenché une action au format `hh:mm:ss`.
 Par exemple : 12:57:53.

{EVENT.TIMESTAMP}

L'horodatage UNIX de l'événement qui a déclenché une action.

{EVENT.VALUE}

La valeur numérique de l'événement qui a déclenché une action.
 Valeurs possibles : *1* - problème, *0* - récupération.

Macros d'événement de cause/symptôme

{EVENT.CAUSE.ACK.STATUS}

Le statut d'acquittement de l'événement cause (Oui/Non).
 Utilisé dans le contexte d'un événement symptôme.

{EVENT.CAUSE.AGE}

L'âge de l'événement cause, avec une précision allant jusqu'à la seconde.
 Utile dans les messages d'escalade.
 Utilisé dans le contexte d'un événement symptôme.

{EVENT.CAUSE.DATE}

La date de l'événement cause au format `yyyy.mm.dd`.
 Utilisé dans le contexte d'un événement symptôme.

{EVENT.CAUSE.DURATION}

La durée de l'événement cause (différence de temps entre les événements de problème et de rétablissement), avec une précision à la seconde près.
 Utile dans les messages de rétablissement de problème.
 Utilisé dans le contexte d'un événement symptôme.

{EVENT.CAUSE.ID}

L'identifiant numérique de l'événement cause.
 Utilisé dans le contexte d'un événement symptôme.

{EVENT.CAUSE.NAME}

Le nom de l'événement cause.
 Utilisé dans le contexte d'un événement symptôme.

{EVENT.CAUSE.NSEVERITY}

La valeur numérique de la sévérité de l'événement cause.
 Valeurs possibles : *0* - Non classé, *1* - Information, *2* - Avertissement, *3* - Moyen, *4* - Élevé, *5* - Désastre.
 Utilisé dans le contexte d'un événement symptôme.

{EVENT.CAUSE.OBJECT}

La valeur numérique de l'objet de l'événement cause.
 Valeurs possibles : *0* - Déclencheur, *1* - Hôte découvert, *2* - Service découvert, *3* - Enregistrement automatique, *4* - Élément, *5* - Règle de découverte de bas niveau.
 Utilisé dans le contexte d'un événement symptôme.

{EVENT.CAUSE.OPDATA}

Les données opérationnelles du déclencheur sous-jacent du problème à l'origine.
 Utilisé dans le contexte d'un événement symptôme.

{EVENT.CAUSE.SEVERITY}

Le nom de la sévérité de l'événement cause.
 Valeurs possibles : *Non classé*, *Information*, *Avertissement*, *Moyenne*, *Élevée*, *Désastre*.
 Utilisé dans le contexte d'un événement symptôme.

{EVENT.CAUSE.SOURCE}

La valeur numérique de la source de l'événement cause.
 Valeurs possibles : *0* - Déclencheur, *1* - Découverte, *2* - Enregistrement automatique, *3* - Interne.
 Utilisé dans le contexte d'un événement symptôme.

{EVENT.CAUSE.STATUS}

La valeur textuelle de l'événement cause.
 Utilisée dans le contexte d'un événement symptôme.

{EVENT.CAUSE.TAGS}

Une liste de tags de l'événement cause, séparés par des virgules.
Développé en une chaîne vide si aucun tag n'existe.
 Utilisé dans le contexte d'un événement symptôme.

{EVENT.CAUSE.TAGSJSON}

Un tableau JSON contenant des objets de balises de l'événement cause.
Développé en tableau vide si aucune balise n'existe.
 Utilisé dans le contexte d'un événement symptôme.

{EVENT.CAUSE.TAGS.<tag name>}

La valeur de la balise de l'événement cause référencée par le nom de la balise.
 Un nom de balise contenant des caractères non alphanumériques (y compris des caractères UTF multioctets non anglais) doit être placé entre guillemets doubles. Les guillemets et les barres obliques inverses à l'intérieur d'un nom de balise entre guillemets doivent être échappés avec une barre oblique inverse.
 Utilisé dans le contexte d'un événement symptôme.

{EVENT.CAUSE.TIME}

L'heure de l'événement cause au format `hh:mm:ss`.
 Utilisé dans le contexte d'un événement symptôme.

{EVENT.CAUSE.TIMESTAMP}

L'horodatage UNIX de l'événement cause.
 Utilisé dans le contexte d'un événement symptôme.

{EVENT.CAUSE.UPDATE.HISTORY}

Le journal des mises à jour du problème cause (acquittements, etc.).
 Utilisé dans le contexte d'un événement symptôme.

{EVENT.CAUSE.VALUE}

La valeur numérique de l'événement cause.
 Valeurs possibles : *1* - Problème, *0* - En cours de récupération.
 Utilisé dans le contexte d'un événement symptôme.

{EVENT.SYMPTOMS}

La liste des événements de symptôme. Inclut les détails suivants : nom de l'hôte, nom de l'événement, gravité, ancienneté, balises de service et valeurs.
 Cette macro est utilisée dans le contexte de l'événement cause et renvoie des informations sur les événements de symptôme.

Macros d'événement de récupération

{EVENT.RECOVERY.DATE}

La date de l'événement de récupération au format `yyyy.mm.dd`.

{EVENT.RECOVERY.ID}

L'ID numérique de l'événement de récupération.

{EVENT.RECOVERY.NAME}

Le nom de l'événement de récupération.

{EVENT.RECOVERY.STATUS}

La valeur textuelle de l'événement de récupération.

{EVENT.RECOVERY.TAGS}

Une liste de tags d'événement de récupération séparés par des virgules. Développée en une chaîne vide si aucun tag n'existe.

{EVENT.RECOVERY.TAGSJSON}

Un tableau JSON contenant des objets de balises d'événement de récupération. Développé en un tableau vide si aucune balise n'existe.

{EVENT.RECOVERY.TIME}

L'heure de l'événement de récupération au format `hh:mm:ss`.

{EVENT.RECOVERY.TIMESTAMP}

L'horodatage UNIX de l'événement de récupération.

{EVENT.RECOVERY.VALUE}

La valeur numérique de l'événement de récupération.

Macros de mise à jour d'événement

{EVENT.UPDATE.ACTION}

Nom lisible de l'action ou des actions effectuées lors de la mise à jour d'un problème.
 Prend les valeurs suivantes : *acknowledged*, *unacknowledged*, *commented*, *changed severity from (original severity) to (updated severity)* et *closed* (selon le nombre d'actions effectuées lors d'une seule mise à jour).

{EVENT.UPDATE.ACTIONJSON}

Un tableau JSON contenant les détails de la ou des action(s) effectuée(s) lors de la mise à jour d'un problème.
 Valeurs possibles des propriétés JSON :
- true (pour les propriétés `acknowledge`, `unacknowledge`, `close`, `unsuppress`, `cause` et `symptom`) ;
- <chaîne de message> (pour la propriété `message`) ;
- horodatage (pour les propriétés `suppress_until` et `timestamp`) ou 0 (pour la propriété `suppress_until` si la suppression est indéfinie) ;
- 0, 1, 2, 3, 4, 5 (pour les propriétés de gravité `old` et `new`).

 Par exemple : `{"acknowledge":true,"message":"Monthly maintenance.","severity":{"old":2,"new":1},"suppress_until":1730851199,"timestamp":1730822048}`.

{EVENT.UPDATE.DATE}

La date de la mise à jour de l'événement (accusé de réception, etc.) au format `yyyy.mm.dd`.

{EVENT.UPDATE.HISTORY}

Le journal des mises à jour du problème (acquittements, etc.).

{EVENT.UPDATE.MESSAGE}

Le message de mise à jour du problème.

{EVENT.UPDATE.NSEVERITY}

La valeur numérique de la nouvelle gravité de l'événement définie lors de l'opération de mise à jour du problème.

{EVENT.UPDATE.SEVERITY}

Le nom de la nouvelle sévérité de l'événement définie lors de l'opération de mise à jour du problème.

{EVENT.UPDATE.STATUS}

La valeur numérique du statut de mise à jour du problème.
 Valeurs possibles : *0* - Le webhook a été appelé en raison d'un événement de problème/rétablissement, *1* - Opération de mise à jour.

{EVENT.UPDATE.TIME}

L'heure de la mise à jour de l'événement (acquittement, etc.) au format `hh:mm:ss`.

{EVENT.UPDATE.TIMESTAMP}

L'horodatage UNIX de la mise à jour de l'événement (accusé de réception, etc.).

Macros de fonction

{FUNCTION.VALUE}

La valeur de la N-ième fonction basée sur un élément dans l'expression du déclencheur au moment de l'événement.
 Seules les fonctions dont le premier paramètre est `/host/key` sont prises en compte.

 Cette macro peut être utilisée avec un index numérique, par exemple {FUNCTION.VALUE<1-9>} pour désigner la première, la deuxième, la troisième, etc. fonction dans une expression de déclencheur. Voir macros indexées.

{FUNCTION.RECOVERY.VALUE}

La valeur de la N-ième fonction basée sur un élément dans l'expression de récupération au moment de l'événement.*
 Seules les fonctions dont le premier paramètre est un `/host/key` sont prises en compte.

 Cette macro peut être utilisée avec un index numérique, par exemple {FUNCTION.RECOVERY.VALUE<1-9>} pour désigner la première, deuxième, troisième, etc. fonction dans une expression de déclencheur. Voir macros indexées.

Macros d'hôte

{HOST.CONN}

L'adresse IP de l'hôte ou le nom DNS, selon les paramètres de l'hôte.
 Dans les scripts globaux, les champs IP/DNS de l'interface et les scénarios web, la macro sera résolue vers la principale interface agent. Si aucune interface agent n'est définie, la principale interface SNMP sera utilisée. Si aucune interface SNMP n'est également définie, la principale interface JMX sera utilisée. Si aucune interface JMX n'est définie non plus, la principale interface IPMI sera utilisée. Si l'hôte ne possède aucune interface, la macro est résolue en 'UNKNOWN'.

 Cette macro peut être utilisée avec un index numérique sous la forme {HOST.CONN<1-9>} afin de désigner le premier, le deuxième, le troisième, etc. hôte dans une expression de déclencheur. Voir macros indexées.

{HOST.DESCRIPTION}

La description de l'hôte.

 Cette macro peut être utilisée avec un index numérique sous la forme {HOST.DESCRIPTION<1-9>} pour désigner le premier, le deuxième, le troisième, etc. hôte dans une expression de déclencheur. Voir macros indexées.

{HOST.DNS}

Le nom DNS de l'hôte.
 Dans les scripts globaux, les champs IP/DNS de l'interface et les scénarios web, la macro sera résolue vers la principale interface agent. Si une interface agent n'est pas définie, la principale interface SNMP sera utilisée. Si une interface SNMP n'est pas non plus définie, la principale interface JMX sera utilisée. Si une interface JMX n'est pas définie non plus, la principale interface IPMI sera utilisée. Si l'hôte ne possède aucune interface, la macro est résolue en 'UNKNOWN'.

 Cette macro peut être utilisée avec un index numérique sous la forme {HOST.DNS<1-9>} pour désigner le premier, le deuxième, le troisième, etc. hôte dans une expression de déclencheur. Voir macros indexées.

{HOST.HOST}

Le nom technique de l'hôte.
 Les fonctions de macro ne sont pas prises en charge pour cette macro lorsqu'elle est utilisée comme espace réservé dans le premier paramètre d'une fonction d'historique, par exemple, `last(/{HOST.HOST}/{ITEM.KEY}).<br><br>` Cette macro peut être utilisée avec un index numérique sous la forme {HOST.HOST<1-9>} pour désigner le premier, le deuxième, le troisième, etc. hôte dans une expression de déclencheur. Voir les macros indexées.

{HOST.ID}

L'ID de l'hôte.

 Cette macro peut être utilisée avec un index numérique sous la forme {HOST.ID<1-9>} pour désigner le premier, le deuxième, le troisième, etc. hôte dans une expression de déclencheur. Voir macros indexées.

{HOST.IP}

L'adresse IP de l'hôte.
 Dans les scripts globaux, les champs IP/DNS de l'interface et les scénarios web, la macro sera résolue vers la principale interface agent. Si aucune interface agent n'est définie, la principale interface SNMP sera utilisée. Si aucune interface SNMP n'est également définie, la principale interface JMX sera utilisée. Si aucune interface JMX n'est définie non plus, la principale interface IPMI sera utilisée. Si l'hôte ne possède aucune interface, la macro est résolue en 'UNKNOWN'.

 Cette macro peut être utilisée avec un index numérique sous la forme {HOST.IP<1-9>} pour désigner le premier, le deuxième, le troisième, etc. hôte dans une expression de déclencheur. Voir macros indexées.

{HOST.METADATA}

Les métadonnées de l'hôte.

 Utilisé uniquement pour l'autoréenregistrement de l'agent actif.

{HOST.NAME}

Le nom visible de l'hôte.

 Cette macro peut être utilisée avec un index numérique sous la forme {HOST.NAME<1-9>} pour désigner le premier, le deuxième, le troisième, etc. hôte dans une expression de déclencheur. Voir macros indexées.

{HOST.PORT}

Le port de l'hôte (agent).
 Dans les scripts globaux, les champs IP/DNS de l'interface et les scénarios web, la macro sera résolue vers l'interface agent principale. Si aucune interface agent n'est définie, l'interface SNMP principale sera utilisée. Si aucune interface SNMP n'est également définie, l'interface JMX principale sera utilisée. Si aucune interface JMX n'est définie non plus, l'interface IPMI principale sera utilisée. Si l'hôte ne possède aucune interface, la macro est résolue en 'UNKNOWN'.

Cette macro peut être utilisée avec un index numérique sous la forme {HOST.PORT<1-9>} pour désigner le premier, le deuxième, le troisième, etc. hôte dans une expression de déclencheur. Voir indexed macros.

Macros d'hôte cible

{HOST.TARGET.CONN}

L'adresse IP ou le nom DNS de l'hôte cible.

{HOST.TARGET.DNS}

Le nom DNS de l'hôte cible.

{HOST.TARGET.HOST}

Le nom technique de l'hôte cible.

{HOST.TARGET.IP}

L'adresse IP de l'hôte cible.

{HOST.TARGET.NAME}

Le nom visible de l'hôte cible.

Macros de groupe d'hôtes

{HOSTGROUP.ID}

L'ID du groupe d'hôtes.

Macros d'inventaire de l'hôte

{INVENTORY.ALIAS}

Le champ *Alias* dans l'inventaire de l'hôte.

 Cette macro peut être utilisée avec un index numérique, par exemple {INVENTORY.ALIAS<1-9>} pour désigner le premier, le deuxième, le troisième, etc. hôte dans une expression de déclencheur. Voir macros indexées.

{INVENTORY.ASSET.TAG}

Le champ *Asset tag* dans l'inventaire de l'hôte.

 Cette macro peut être utilisée avec un index numérique, par exemple {INVENTORY.ASSET.TAG<1-9>} pour désigner le premier, le deuxième, le troisième, etc. hôte dans une expression de déclencheur. Voir macros indexées.

{INVENTORY.CHASSIS}

Le champ *Chassis* dans l'inventaire de l'hôte.

 Cette macro peut être utilisée avec un index numérique, par exemple {INVENTORY.CHASSIS<1-9>} pour désigner le premier, le deuxième, le troisième, etc. hôte dans une expression de déclencheur. Voir macros indexées.

{INVENTORY.CONTACT}

Le champ *Contact* dans l'inventaire de l'hôte.

 Cette macro peut être utilisée avec un index numérique, par exemple {INVENTORY.CONTACT<1-9>} pour désigner le premier, le deuxième, le troisième, etc. hôte dans une expression de déclencheur. Voir macros indexées.

{INVENTORY.CONTRACT.NUMBER}

Le champ *Numéro de contrat* dans l'inventaire de l'hôte.

 Cette macro peut être utilisée avec un index numérique, par exemple {INVENTORY.CONTRACT.NUMBER<1-9>} pour désigner le premier, le deuxième, le troisième, etc. hôte dans une expression de déclencheur. Voir macros indexées.

{INVENTORY.DEPLOYMENT.STATUS}

Le champ *Deployment status* dans l'inventaire de l'hôte.

 Cette macro peut être utilisée avec un index numérique, par exemple {INVENTORY.DEPLOYMENT.STATUS<1-9>} pour désigner le premier, le deuxième, le troisième, etc. hôte dans une expression de déclencheur. Voir indexed macros.

{INVENTORY.HARDWARE}

Le champ *Hardware* dans l'inventaire de l'hôte.

 Cette macro peut être utilisée avec un indice numérique, par exemple {INVENTORY.HARDWARE<1-9>} pour désigner le premier, le deuxième, le troisième, etc. hôte dans une expression de déclencheur. Voir macros indexées.

{INVENTORY.HARDWARE.FULL}

Le champ *Matériel (détails complets)* dans l'inventaire de l'hôte.

 Cette macro peut être utilisée avec un index numérique, par exemple {INVENTORY.HARDWARE.FULL<1-9>} pour désigner le premier, le deuxième, le troisième, etc. hôte dans une expression de déclencheur. Voir macros indexées.

{INVENTORY.HOST.NETMASK}

Le champ *Masque de sous-réseau de l'hôte* dans l'inventaire de l'hôte.

 Cette macro peut être utilisée avec un index numérique, par exemple {INVENTORY.HOST.NETMASK<1-9>} pour désigner le premier, le deuxième, le troisième, etc. hôte dans une expression de déclencheur. Voir macros indexées.

{INVENTORY.HOST.NETWORKS}

Le champ *Host networks* dans l'inventaire de l'hôte.

 Cette macro peut être utilisée avec un index numérique, par exemple {INVENTORY.HOST.NETWORKS<1-9>} pour désigner le premier, le deuxième, le troisième, etc. hôte dans une expression de déclencheur. Voir indexed macros.

{INVENTORY.HOST.ROUTER}

Le champ *routeur de l'hôte* dans l'inventaire de l'hôte.

 Cette macro peut être utilisée avec un index numérique, par exemple {INVENTORY.HOST.ROUTER<1-9>} pour désigner le premier, le deuxième, le troisième, etc. hôte dans une expression de déclencheur. Voir macros indexées.

{INVENTORY.HW.ARCH}

Le champ *Hardware architecture* dans l'inventaire de l'hôte.

 Cette macro peut être utilisée avec un index numérique, par exemple {INVENTORY.HW.ARCH<1-9>} pour désigner le premier, le deuxième, le troisième, etc. hôte dans une expression de déclencheur. Voir indexed macros.

{INVENTORY.HW.DATE.DECOMM}

Le champ *Date de mise hors service du matériel* dans l'inventaire de l'hôte.

 Cette macro peut être utilisée avec un index numérique, par exemple {INVENTORY.HW.DATE.DECOMM<1-9>} pour désigner le premier, le deuxième, le troisième, etc. hôte dans une expression de déclencheur. Voir macros indexées.

{INVENTORY.HW.DATE.EXPIRY}

Le champ *Date d'expiration de la maintenance matérielle* dans l'inventaire de l'hôte.

 Cette macro peut être utilisée avec un index numérique, par exemple {INVENTORY.HW.DATE.EXPIRY<1-9>} pour désigner le premier, le deuxième, le troisième, etc. hôte dans une expression de déclencheur. Voir macros indexées.

{INVENTORY.HW.DATE.INSTALL}

Le champ *Date d'installation du matériel* dans l'inventaire de l'hôte.

 Cette macro peut être utilisée avec un index numérique, par exemple {INVENTORY.HW.DATE.INSTALL<1-9>} pour désigner le premier, le deuxième, le troisième, etc. hôte dans une expression de déclencheur. Voir macros indexées.

{INVENTORY.HW.DATE.PURCHASE}

Le champ *Date d'achat du matériel* dans l'inventaire de l'hôte.

 Cette macro peut être utilisée avec un index numérique, par exemple {INVENTORY.HW.DATE.PURCHASE<1-9>} pour désigner le premier, le deuxième, le troisième, etc. hôte dans une expression de déclencheur. Voir macros indexées.

{INVENTORY.INSTALLER.NAME}

Le champ *Nom de l'installateur* dans l'inventaire de l'hôte.

 Cette macro peut être utilisée avec un index numérique, par exemple {INVENTORY.INSTALLER.NAME<1-9>} pour désigner le premier, le deuxième, le troisième, etc. hôte dans une expression de déclencheur. Voir macros indexées.

{INVENTORY.LOCATION}

Le champ *Location* dans l'inventaire de l'hôte.

 Cette macro peut être utilisée avec un index numérique, par exemple {INVENTORY.LOCATION<1-9>} pour désigner le premier, le deuxième, le troisième, etc. hôte dans une expression de déclencheur. Voir macros indexées.

{INVENTORY.LOCATION.LAT}

Le champ *Latitude de l'emplacement* dans l'inventaire de l'hôte.

 Cette macro peut être utilisée avec un index numérique, par exemple {INVENTORY.LOCATION.LAT<1-9>} pour désigner le premier, le deuxième, le troisième, etc. hôte dans une expression de déclencheur. Voir macros indexées.

{INVENTORY.LOCATION.LON}

Le champ *Longitude de l'emplacement* dans l'inventaire de l'hôte.

 Cette macro peut être utilisée avec un index numérique, par exemple {INVENTORY.LOCATION.LON<1-9>} pour désigner le premier, le deuxième, le troisième, etc. hôte dans une expression de déclencheur. Voir macros indexées.

{INVENTORY.MACADDRESS.A}

Le champ *adresse MAC A* dans l'inventaire de l'hôte.

 Cette macro peut être utilisée avec un index numérique, par exemple {INVENTORY.MACADDRESS.A<1-9>} pour désigner le premier, le deuxième, le troisième, etc. hôte dans une expression de déclencheur. Voir macros indexées.

{INVENTORY.MACADDRESS.B}

Le champ *adresse MAC B* dans l'inventaire de l'hôte.

 Cette macro peut être utilisée avec un index numérique, par exemple {INVENTORY.MACADDRESS.B<1-9>} pour désigner le premier, le deuxième, le troisième, etc. hôte dans une expression de déclencheur. Voir macros indexées.

{INVENTORY.MODEL}

Le champ *Modèle* dans l'inventaire de l'hôte.

 Cette macro peut être utilisée avec un index numérique, par exemple {INVENTORY.MODEL<1-9>} pour désigner le premier, le deuxième, le troisième, etc. hôte dans une expression de déclencheur. Voir macros indexées.

{INVENTORY.NAME}

Le champ *Name* dans l'inventaire de l'hôte.

 Cette macro peut être utilisée avec un index numérique, par exemple {INVENTORY.NAME<1-9>} pour désigner le premier, le deuxième, le troisième, etc. hôte dans une expression de déclencheur. Voir indexed macros.

{INVENTORY.NOTES}

Le champ *Notes* dans l'inventaire de l'hôte.

 Cette macro peut être utilisée avec un index numérique, par exemple {INVENTORY.NOTES<1-9>} pour désigner le premier, le deuxième, le troisième, etc. hôte dans une expression de déclencheur. Voir macros indexées.

{INVENTORY.OOB.IP}

Le champ *adresse IP OOB* dans l'inventaire de l'hôte.

 Cette macro peut être utilisée avec un index numérique, par exemple {INVENTORY.OOB.IP<1-9>} pour désigner le premier, le deuxième, le troisième, etc. hôte dans une expression de déclencheur. Voir macros indexées.

{INVENTORY.OOB.NETMASK}

Le champ *masque de sous-réseau OOB* dans l'inventaire de l'hôte.

 Cette macro peut être utilisée avec un index numérique, par exemple {INVENTORY.OOB.NETMASK<1-9>} pour désigner le premier, le deuxième, le troisième, etc. hôte dans une expression de déclencheur. Voir macros indexées.

{INVENTORY.OOB.ROUTER}

Le champ *routeur OOB* dans l'inventaire de l'hôte.

 Cette macro peut être utilisée avec un index numérique, par exemple {INVENTORY.OOB.ROUTER<1-9>} pour désigner le premier, le deuxième, le troisième, etc. hôte dans une expression de déclencheur. Voir macros indexées.

{INVENTORY.OS}

Le champ *OS* dans l'inventaire de l'hôte.

 Cette macro peut être utilisée avec un index numérique, par exemple {INVENTORY.OS<1-9>} pour désigner le premier, le deuxième, le troisième, etc. hôte dans une expression de déclencheur. Voir macros indexées.

{INVENTORY.OS.FULL}

Le champ *OS (détails complets)* dans l'inventaire de l'hôte.

 Cette macro peut être utilisée avec un index numérique, par exemple {INVENTORY.OS.FULL<1-9>} pour désigner le premier, le deuxième, le troisième, etc. hôte dans une expression de déclencheur. Voir macros indexées.

{INVENTORY.OS.SHORT}

Le champ *OS (court)* dans l'inventaire de l'hôte.

 Cette macro peut être utilisée avec un index numérique, par exemple {INVENTORY.OS.SHORT<1-9>} pour désigner le premier, le deuxième, le troisième, etc. hôte dans une expression de déclencheur. Voir macros indexées.

{INVENTORY.POC.PRIMARY.CELL}

Le champ *cellule POC principale* dans l'inventaire de l'hôte.

 Cette macro peut être utilisée avec un index numérique, par exemple {INVENTORY.POC.PRIMARY.CELL<1-9>} pour désigner le premier, le deuxième, le troisième, etc. hôte dans une expression de déclencheur. Voir macros indexées.

{INVENTORY.POC.PRIMARY.EMAIL}

Le champ *Primary POC email* dans l'inventaire de l'hôte.

 Cette macro peut être utilisée avec un index numérique, par exemple {INVENTORY.POC.PRIMARY.EMAIL<1-9>} pour désigner le premier, le deuxième, le troisième, etc. hôte dans une expression de déclencheur. Voir indexed macros.

{INVENTORY.POC.PRIMARY.NAME}

Le champ *Nom du contact principal* dans l'inventaire de l'hôte.

 Cette macro peut être utilisée avec un index numérique, par exemple {INVENTORY.POC.PRIMARY.NAME<1-9>} pour désigner le premier, le deuxième, le troisième, etc. hôte dans une expression de déclencheur. Voir macros indexées.

{INVENTORY.POC.PRIMARY.NOTES}

Le champ *Notes du POC principal* dans l'inventaire de l'hôte.

 Cette macro peut être utilisée avec un index numérique, par exemple {INVENTORY.POC.PRIMARY.NOTES<1-9>} pour désigner le premier, le deuxième, le troisième, etc. hôte dans une expression de déclencheur. Voir macros indexées.

{INVENTORY.POC.PRIMARY.PHONE.A}

Le champ *Téléphone A du contact principal* dans l'inventaire de l'hôte.

 Cette macro peut être utilisée avec un index numérique, par exemple {INVENTORY.POC.PRIMARY.PHONE.A<1-9>} pour désigner le premier, le deuxième, le troisième, etc. hôte dans une expression de déclencheur. Voir macros indexées.

{INVENTORY.POC.PRIMARY.PHONE.B}

Le champ *Téléphone B du contact principal* dans l'inventaire de l'hôte.

 Cette macro peut être utilisée avec un index numérique, par exemple {INVENTORY.POC.PRIMARY.PHONE.B<1-9>} pour désigner le premier, le deuxième, le troisième, etc. hôte dans une expression de déclencheur. Voir macros indexées.

{INVENTORY.POC.PRIMARY.SCREEN}

Le champ *Nom de l'écran du contact principal* dans l'inventaire de l'hôte.

 Cette macro peut être utilisée avec un index numérique, par exemple {INVENTORY.POC.PRIMARY.SCREEN<1-9>} pour désigner le premier, le deuxième, le troisième, etc. hôte dans une expression de déclencheur. Voir macros indexées.

{INVENTORY.POC.SECONDARY.CELL}

Le champ *cellule POC secondaire* dans l'inventaire de l'hôte.

 Cette macro peut être utilisée avec un index numérique, par exemple {INVENTORY.POC.SECONDARY.CELL<1-9>} pour désigner le premier, le deuxième, le troisième, etc. hôte dans une expression de déclencheur. Voir macros indexées.

{INVENTORY.POC.SECONDARY.EMAIL}

Le champ *e-mail du contact secondaire* dans l'inventaire de l'hôte.

 Cette macro peut être utilisée avec un index numérique, par exemple {INVENTORY.POC.SECONDARY.EMAIL<1-9>} pour désigner le premier, le deuxième, le troisième, etc. hôte dans une expression de déclencheur. Voir macros indexées.

{INVENTORY.POC.SECONDARY.NAME}

Le champ *Nom du contact secondaire* dans l'inventaire de l'hôte.

 Cette macro peut être utilisée avec un index numérique, par exemple {INVENTORY.POC.SECONDARY.NAME<1-9>} pour désigner le premier, le deuxième, le troisième, etc. hôte dans une expression de déclencheur. Voir macros indexées.

{INVENTORY.POC.SECONDARY.NOTES}

Le champ *Notes du contact secondaire* dans l'inventaire de l'hôte.

 Cette macro peut être utilisée avec un index numérique, par exemple {INVENTORY.POC.SECONDARY.NOTES<1-9>} pour désigner le premier, le deuxième, le troisième, etc. hôte dans une expression de déclencheur. Voir macros indexées.

{INVENTORY.POC.SECONDARY.PHONE.A}

Le champ *Secondary POC phone A* dans l'inventaire de l'hôte.

 Cette macro peut être utilisée avec un index numérique, par exemple {INVENTORY.POC.SECONDARY.PHONE.A<1-9>} pour désigner le premier, le deuxième, le troisième, etc. hôte dans une expression de déclencheur. Voir indexed macros.

{INVENTORY.POC.SECONDARY.PHONE.B}

Le champ *Secondary POC phone B* dans l'inventaire de l'hôte.

 Cette macro peut être utilisée avec un index numérique, par exemple {INVENTORY.POC.SECONDARY.PHONE.B<1-9>} pour désigner le premier, le deuxième, le troisième, etc. hôte dans une expression de déclencheur. Voir indexed macros.

{INVENTORY.POC.SECONDARY.SCREEN}

Le champ *Nom de l'écran du POC secondaire* dans l'inventaire de l'hôte.

 Cette macro peut être utilisée avec un index numérique, par exemple {INVENTORY.POC.SECONDARY.SCREEN<1-9>} pour désigner le premier, le deuxième, le troisième, etc. hôte dans une expression de déclencheur. Voir macros indexées.

{INVENTORY.SERIALNO.A}

Le champ *Numéro de série A* dans l'inventaire de l'hôte.

 Cette macro peut être utilisée avec un index numérique, par exemple {INVENTORY.SERIALNO.A<1-9>} pour désigner le premier, le deuxième, le troisième, etc. hôte dans une expression de déclencheur. Voir macros indexées.

{INVENTORY.SERIALNO.B}

Le champ *Numéro de série B* dans l'inventaire de l'hôte.

 Cette macro peut être utilisée avec un index numérique, par exemple {INVENTORY.SERIALNO.B<1-9>} pour désigner le premier, le deuxième, le troisième, etc. hôte dans une expression de déclencheur. Voir macros indexées.

{INVENTORY.SITE.ADDRESS.A}

Le champ *Site address A* dans l'inventaire de l'hôte.

 Cette macro peut être utilisée avec un index numérique, par exemple {INVENTORY.SITE.ADDRESS.A<1-9>} pour désigner le premier, le deuxième, le troisième, etc. hôte dans une expression de déclencheur. Voir macros indexées.

{INVENTORY.SITE.ADDRESS.B}

Le champ *Adresse du site B* dans l'inventaire de l'hôte.

 Cette macro peut être utilisée avec un index numérique, par exemple {INVENTORY.SITE.ADDRESS.B<1-9>} pour désigner le premier, le deuxième, le troisième, etc. hôte dans une expression de déclencheur. Voir macros indexées.

{INVENTORY.SITE.ADDRESS.C}

Le champ *Adresse du site C* dans l'inventaire de l'hôte.

 Cette macro peut être utilisée avec un index numérique, par exemple {INVENTORY.SITE.ADDRESS.C<1-9>} pour désigner le premier, le deuxième, le troisième, etc. hôte dans une expression de déclencheur. Voir macros indexées.

{INVENTORY.SITE.CITY}

Le champ *Ville du site* dans l'inventaire de l'hôte.

 Cette macro peut être utilisée avec un index numérique, par exemple {INVENTORY.SITE.CITY<1-9>} pour désigner le premier, le deuxième, le troisième, etc. hôte dans une expression de déclencheur. Voir macros indexées.

{INVENTORY.SITE.COUNTRY}

Le champ *Pays du site* dans l'inventaire de l'hôte.

 Cette macro peut être utilisée avec un index numérique, par exemple {INVENTORY.SITE.COUNTRY<1-9>} pour désigner le premier, le deuxième, le troisième, etc. hôte dans une expression de déclencheur. Voir macros indexées.

{INVENTORY.SITE.NOTES}

Le champ *Site notes* dans l'inventaire de l'hôte.

 Cette macro peut être utilisée avec un index numérique, par exemple {INVENTORY.SITE.NOTES<1-9>} pour désigner le premier, le deuxième, le troisième, etc. hôte dans une expression de déclencheur. Voir macros indexées.

{INVENTORY.SITE.RACK}

Le champ *Emplacement du rack du site* dans l'inventaire de l'hôte.

 Cette macro peut être utilisée avec un index numérique, par exemple {INVENTORY.SITE.RACK<1-9>} pour désigner le premier, le deuxième, le troisième, etc. hôte dans une expression de déclencheur. Voir macros indexées.

{INVENTORY.SITE.STATE}

Le champ *État/province du site* dans l'inventaire de l'hôte.

 Cette macro peut être utilisée avec un index numérique, par exemple {INVENTORY.SITE.STATE<1-9>} pour désigner le premier, le deuxième, le troisième, etc. hôte dans une expression de déclencheur. Voir macros indexées.

{INVENTORY.SITE.ZIP}

Le champ *Code postal du site* dans l'inventaire de l'hôte.

 Cette macro peut être utilisée avec un index numérique, par exemple {INVENTORY.SITE.ZIP<1-9>} pour désigner le premier, le deuxième, le troisième, etc. hôte dans une expression de déclencheur. Voir macros indexées.

{INVENTORY.SOFTWARE}

Le champ *Software* dans l'inventaire de l'hôte.

 Cette macro peut être utilisée avec un index numérique, par exemple {INVENTORY.SOFTWARE<1-9>} pour désigner le premier, le deuxième, le troisième, etc. hôte dans une expression de déclencheur. Voir macros indexées.

{INVENTORY.SOFTWARE.APP.A}

Le champ *Application logicielle A* dans l'inventaire de l'hôte.

 Cette macro peut être utilisée avec un index numérique, par exemple {INVENTORY.SOFTWARE.APP.A<1-9>} pour désigner le premier, le deuxième, le troisième, etc. hôte dans une expression de déclencheur. Voir macros indexées.

{INVENTORY.SOFTWARE.APP.B}

Le champ *Software application B* dans l'inventaire de l'hôte.

 Cette macro peut être utilisée avec un index numérique, par exemple {INVENTORY.SOFTWARE.APP.B<1-9>} pour désigner le premier, le deuxième, le troisième, etc. hôte dans une expression de déclencheur. Voir indexed macros.

{INVENTORY.SOFTWARE.APP.C}

Le champ *Application logicielle C* dans l'inventaire de l'hôte.

 Cette macro peut être utilisée avec un index numérique, par exemple {INVENTORY.SOFTWARE.APP.C<1-9>} pour désigner le premier, le deuxième, le troisième, etc. hôte dans une expression de déclencheur. Voir macros indexées.

{INVENTORY.SOFTWARE.APP.D}

Le champ *Software application D* dans l'inventaire de l'hôte.

 Cette macro peut être utilisée avec un index numérique, par exemple {INVENTORY.SOFTWARE.APP.D<1-9>} pour désigner le premier, le deuxième, le troisième, etc. hôte dans une expression de déclencheur. Voir macros indexées.

{INVENTORY.SOFTWARE.APP.E}

Le champ *Application logicielle E* dans l'inventaire de l'hôte.

 Cette macro peut être utilisée avec un index numérique, par exemple {INVENTORY.SOFTWARE.APP.E<1-9>} pour désigner le premier, le deuxième, le troisième, etc. hôte dans une expression de déclencheur. Voir macros indexées.

{INVENTORY.SOFTWARE.FULL}

Le champ *Logiciel (détails complets)* dans l'inventaire de l'hôte.

 Cette macro peut être utilisée avec un index numérique, par exemple {INVENTORY.SOFTWARE.FULL<1-9>} pour désigner le premier, le deuxième, le troisième, etc. hôte dans une expression de déclencheur. Voir macros indexées.

{INVENTORY.TAG}

Le champ *Tag* dans l'inventaire de l'hôte.

 Cette macro peut être utilisée avec un index numérique, par exemple {INVENTORY.TAG<1-9>} pour désigner le premier, le deuxième, le troisième, etc. hôte dans une expression de déclencheur. Voir macros indexées.

{INVENTORY.TYPE}

Le champ *Type* dans l'inventaire de l'hôte.

 Cette macro peut être utilisée avec un index numérique, par exemple {INVENTORY.TYPE<1-9>} pour désigner le premier, le deuxième, le troisième, etc. hôte dans une expression de déclencheur. Voir macros indexées.

{INVENTORY.TYPE.FULL}

Le champ *Type (détails complets)* dans l'inventaire de l'hôte.

 Cette macro peut être utilisée avec un indice numérique, par exemple {INVENTORY.TYPE.FULL<1-9>} pour désigner le premier, le deuxième, le troisième, etc. hôte dans une expression de déclencheur. Voir macros indexées.

{INVENTORY.URL.A}

Le champ *URL A* dans l'inventaire de l'hôte.

 Cette macro peut être utilisée avec un index numérique, par exemple {INVENTORY.URL.A<1-9>} pour désigner le premier, le deuxième, le troisième, etc. hôte dans une expression de déclencheur. Voir macros indexées.

{INVENTORY.URL.B}

Le champ *URL B* dans l'inventaire de l'hôte.

 Cette macro peut être utilisée avec un index numérique, par exemple {INVENTORY.URL.B<1-9>} pour désigner le premier, le deuxième, le troisième, etc. hôte dans une expression de déclencheur. Voir macros indexées.

{INVENTORY.URL.C}

Le champ *URL C* dans l'inventaire de l'hôte.

 Cette macro peut être utilisée avec un index numérique, par exemple {INVENTORY.URL.C<1-9>} pour désigner le premier, le deuxième, le troisième, etc. hôte dans une expression de déclencheur. Voir macros indexées.

{INVENTORY.VENDOR}

Le champ *Vendor* dans l'inventaire de l'hôte.

 Cette macro peut être utilisée avec un index numérique, par exemple {INVENTORY.VENDOR<1-9>} pour désigner le premier, le deuxième, le troisième, etc. hôte dans une expression de déclencheur. Voir indexed macros.

Macros d'élément

{ITEM.DESCRIPTION}

La description du N-ième élément dans l'expression du déclencheur qui a provoqué une notification.

 Cette macro peut être utilisée avec un indice numérique sous la forme {ITEM.DESCRIPTION<1-9>} pour désigner le premier, le deuxième, le troisième, etc. hôte dans une expression de déclencheur. Voir macros indexées.

{ITEM.DESCRIPTION.ORIG}

La description originale (avec les macros non résolues) du N-ième élément dans l'expression du déclencheur qui a provoqué une notification.

 Cette macro peut être utilisée avec un index numérique sous la forme {ITEM.DESCRIPTION.ORIG<1-9>} pour désigner le premier, le deuxième, le troisième, etc. hôte dans une expression de déclencheur. Voir macros indexées.

{ITEM.ID}

L'identifiant numérique du N-ième élément dans l'expression du déclencheur qui a provoqué une notification.

 Cette macro peut être utilisée avec un index numérique sous la forme {ITEM.ID<1-9>} pour désigner le premier, le deuxième, le troisième, etc. hôte dans une expression de déclencheur. Voir indexed macros.

{ITEM.KEY}

La clé du nième élément dans l'expression du déclencheur qui a provoqué une notification.
 Les fonctions de macro ne sont pas prises en charge pour cette macro si elle est utilisée comme espace réservé dans le premier paramètre d'une fonction d'historique, par exemple, `last(/{HOST.HOST}/{ITEM.KEY})`.

 Cette macro peut être utilisée avec un index numérique sous la forme {ITEM.KEY<1-9>} pour désigner le premier, le deuxième, le troisième, etc. hôte dans une expression de déclencheur. Voir les macros indexées.

{ITEM.KEY.ORIG}

La clé d'origine (avec les macros non résolues) du Ne élément dans l'expression du déclencheur qui a provoqué une notification.

 Cette macro peut être utilisée avec un index numérique sous la forme {ITEM.KEY.ORIG<1-9>} pour désigner le premier, le deuxième, le troisième, etc. hôte dans une expression de déclencheur. Voir macros indexées.

{ITEM.LASTVALUE}

La dernière valeur du N-ième élément dans l'expression du déclencheur qui a provoqué une notification.
 Elle sera résolue en *UNKNOWN* dans l'interface si la dernière valeur d'historique a été collectée il y a plus de *Max history display period* (défini dans la section de menu *Administration > General*).
 Lorsqu'elle est utilisée dans le nom du problème, la macro ne sera pas résolue à la dernière valeur de l'élément lors de l'affichage des événements de problème ; elle conservera plutôt la valeur de l'élément au moment où le problème s'est produit.
Lorsqu'elle est utilisée dans les notifications, dans certains cas, la macro peut ne pas être résolue à la dernière valeur de l'élément au moment où le déclencheur s'est déclenché. Par exemple, si un élément reçoit rapidement deux valeurs, "A" et "B", et que le déclencheur se déclenche pour "A", les notifications peuvent afficher "B" comme dernière valeur en raison d'un léger délai de traitement - la dernière valeur de l'élément a changé entre le moment où le déclencheur s'est déclenché et celui où la notification a été créée. Pour éviter cela, vous pouvez utiliser la macro {ITEM.VALUE}, qui se résout à la valeur au moment où le déclencheur se déclenche, garantissant ainsi que la valeur correcte est utilisée dans la notification.
 C'est un alias de `last(/{HOST.HOST}/{ITEM.KEY})`.
 La valeur résolue pour les éléments de type texte/journal est tronquée à 20 caractères par l'interface dans les emplacements suivants :
- Données opérationnelles;
- Description du déclencheur;
- URLs du déclencheur;
- Libellés des URL du déclencheur;
- Description du widget de valeur de l'élément.
Pour obtenir la valeur complète, vous pouvez utiliser des fonctions de macro, car aucune valeur n'est tronquée par le serveur. Par exemple : `{{ITEM.LASTVALUE}.regsub("(.*)", \1)}`.

 Cette macro peut être utilisée avec un index numérique, par exemple {ITEM.LASTVALUE<1-9>} pour désigner le premier, le deuxième, le troisième, etc. élément dans une expression de déclencheur. Voir macros indexées.

{ITEM.LASTVALUE.AGE}

Le temps écoulé entre la dernière collecte de la valeur de l'élément et l'évaluation de la macro.
 Affiché dans un format lisible par l'humain (par exemple 1m 45s).

 Cette macro peut être utilisée avec un index numérique, par exemple {ITEM.LASTVALUE.AGE<1-9>} pour désigner le premier, le deuxième, le troisième, etc. élément dans une expression de déclencheur. Voir macros indexées.

{ITEM.LASTVALUE.DATE}

La date à laquelle la dernière valeur de l'élément a été collectée.
 Affichée au format `YYYYMMDD`.
 Le fuseau horaire du serveur est utilisé dans les noms de déclencheur (dans la liste *Monitoring > Problems*), les noms d'événement, ainsi que les noms et valeurs des tags. Dans tous les autres cas, le fuseau horaire de l'utilisateur est utilisé.

 Cette macro peut être utilisée avec un index numérique, par exemple {ITEM.LASTVALUE.DATE<1-9>} pour désigner le premier, le deuxième, le troisième, etc. élément dans une expression de déclencheur. Voir macros indexées.

{ITEM.LASTVALUE.TIME}

L'heure à laquelle la dernière valeur de l'élément a été collectée.
 Affichée au format HHMMSS.
 Le fuseau horaire du serveur est utilisé dans les noms de déclencheur (dans la liste *Monitoring* > *Problems*), les noms d'événement, ainsi que les noms et valeurs des balises. Dans tous les autres cas, le fuseau horaire de l'utilisateur est utilisé.

 Cette macro peut être utilisée avec un index numérique, par exemple, {ITEM.LASTVALUE.TIME<1-9>} pour désigner le premier, le deuxième, le troisième, etc. élément dans une expression de déclencheur. Voir macros indexées.

{ITEM.LASTVALUE.TIMESTAMP}

L'horodatage UNIX auquel la dernière valeur de l'élément a été collectée.

 Cette macro peut être utilisée avec un index numérique, par exemple {ITEM.LASTVALUE.TIMESTAMP<1-9>} pour désigner le premier, le deuxième, le troisième, etc. élément dans une expression de déclencheur. Voir macros indexées.

{ITEM.LOG.AGE}

L'ancienneté de l'événement du journal.
Avec une précision à la seconde près.

 Cette macro peut être utilisée avec un index numérique, par exemple {ITEM.LOG.AGE<1-9>} pour désigner le premier, le deuxième, le troisième, etc. élément dans une expression de déclencheur. Voir indexed macros.

{ITEM.LOG.DATE}

La date à laquelle l'entrée de journal a été écrite dans le journal.

 Cette macro peut être utilisée avec un index numérique, par exemple {ITEM.LOG.DATE<1-9>} pour désigner le premier, le deuxième, le troisième, etc. élément dans une expression de déclencheur. Voir macros indexées.

{ITEM.LOG.EVENTID}

L'ID de l'événement dans le journal des événements.
 Pour la surveillance du journal des événements Windows uniquement.

 Cette macro peut être utilisée avec un index numérique, par exemple {ITEM.LOG.EVENTID<1-9>} pour pointer vers le premier, le deuxième, le troisième, etc. élément dans une expression de déclencheur. Voir macros indexées.

{ITEM.LOG.NSEVERITY}

La gravité numérique de l'événement dans le journal des événements.
 Pour la surveillance du journal des événements Windows uniquement.

 Cette macro peut être utilisée avec un index numérique, par exemple {ITEM.LOG.NSEVERITY<1-9>} pour désigner le premier, le deuxième, le troisième, etc. élément dans une expression de déclencheur. Voir macros indexées.

{ITEM.LOG.SEVERITY}

La gravité verbale de l'événement dans le journal des événements.
 Pour la surveillance du journal des événements Windows uniquement.

 Cette macro peut être utilisée avec un index numérique, par exemple {ITEM.LOG.SEVERITY<1-9>} pour désigner le premier, le deuxième, le troisième, etc. élément dans une expression de déclencheur. Voir macros indexées.

{ITEM.LOG.SOURCE}

La source de l'événement dans le journal des événements.
 Pour la surveillance du journal des événements Windows uniquement.

 Cette macro peut être utilisée avec un index numérique, par exemple {ITEM.LOG.SOURCE<1-9>} pour désigner le premier, le deuxième, le troisième, etc. élément dans une expression de déclencheur. Voir indexed macros.

{ITEM.LOG.TIME}

L'heure à laquelle l'entrée de journal a été écrite dans le journal.

 Cette macro peut être utilisée avec un indice numérique, par exemple {ITEM.LOG.TIME<1-9>} pour désigner le premier, le deuxième, le troisième, etc. élément dans une expression de déclencheur. Voir indexed macros.

{ITEM.LOG.TIMESTAMP}

L'horodatage UNIX auquel l'entrée de journal a été écrite dans le journal.

 Cette macro peut être utilisée avec un index numérique, par exemple {ITEM.LOG.TIMESTAMP<1-9>} pour désigner le premier, le deuxième, le troisième, etc. élément dans une expression de déclencheur. Voir macros indexées.

{ITEM.NAME}

Le nom de l'élément avec toutes les macros résolues.

 Cette macro peut être utilisée avec un index numérique, par exemple {ITEM.NAME<1-9>} pour désigner le premier, le deuxième, le troisième, etc. élément dans une expression de déclencheur. Voir macros indexées.

{ITEM.NAME.ORIG}

Le nom original (avec les macros non résolues) de l'élément.

 Cette macro peut être utilisée avec un indice numérique, par exemple {ITEM.NAME.ORIG<1-9>} pour désigner le premier, le deuxième, le troisième, etc. élément dans une expression de déclencheur. Voir indexed macros.

{ITEM.STATE}

Le dernier état du N-ième élément dans l'expression du déclencheur qui a provoqué une notification.
 Valeurs possibles : *Not supported*, *Normal*.

 Cette macro peut être utilisée avec un index numérique, par exemple {ITEM.STATE<1-9>} pour désigner le premier, le deuxième, le troisième, etc. élément dans une expression de déclencheur. Voir indexed macros.

{ITEM.STATE.ERROR}

Le message d'erreur contenant les détails expliquant pourquoi un élément est devenu non pris en charge.
 Si un élément passe à l'état non pris en charge puis redevient immédiatement pris en charge, le champ d'erreur peut être vide.

{ITEM.VALUE}

Résolu en l'une des deux valeurs suivantes :

- La valeur historique (au moment de l'événement) du N-ième élément dans l'expression du déclencheur, si elle est utilisée dans le contexte d'un changement d'état du déclencheur, par exemple lors de l'affichage d'événements ou de l'envoi de notifications ;
- La dernière valeur du N-ième élément dans l'expression du déclencheur, si elle est utilisée sans le contexte d'un changement d'état du déclencheur, par exemple lors de l'affichage d'une liste de déclencheurs dans une fenêtre de sélection contextuelle. Dans ce cas, elle fonctionne de la même manière que {ITEM.LASTVALUE}.

Dans le premier cas, elle sera résolue en *UNKNOWN* si la valeur historique a déjà été supprimée ou n'a jamais été stockée.
Dans le second cas, et uniquement dans l'interface, elle sera résolue en *UNKNOWN* si la dernière valeur historique a été collectée il y a plus de *Max history display period* (défini dans la section de menu *Administration > General*).
La valeur résolue pour les éléments de type texte/journal est tronquée à 20 caractères par l'interface dans les emplacements suivants :
- Données opérationnelles ;
- Description du déclencheur ;
- URL du déclencheur ;
- Libellés d'URL du déclencheur ;
- Description du widget de valeur de l'élément.
Pour obtenir la valeur complète, vous pouvez utiliser des fonctions de macro, car aucune valeur n'est tronquée par le serveur. Par exemple : `{{ITEM.VALUE}.regsub("(.*)", \1)}`.

Cette macro peut être utilisée avec un index numérique, par exemple {ITEM.VALUE<1-9>} pour désigner le premier, le deuxième, le troisième, etc. élément dans une expression du déclencheur. Voir les macros indexées.

{ITEM.VALUE.AGE}

Le temps écoulé entre la collecte de la valeur de l'élément et l'évaluation de la macro.
 Affiché dans un format lisible par l'utilisateur (par exemple 1m 45s).

 Cette macro peut être utilisée avec un index numérique, par exemple {ITEM.VALUE.AGE<1-9>} pour désigner le premier, le deuxième, le troisième, etc. élément dans une expression de déclencheur. Voir macros indexées.

{ITEM.VALUE.DATE}

La date à laquelle la valeur de l'élément a été collectée.
 Affichée au format `YYYYMMDD`.
 Le fuseau horaire du serveur est utilisé dans les noms de déclencheur (dans la liste *Monitoring* > *Problems*), les noms d'événements, ainsi que les noms et valeurs des balises. Dans tous les autres cas, le fuseau horaire de l'utilisateur est utilisé.

Cette macro peut être utilisée avec un index numérique, par exemple, {ITEM.VALUE.DATE<1-9>} pour désigner le premier, le deuxième, le troisième, etc. élément dans une expression de déclencheur. Voir macros indexées.

{ITEM.VALUE.TIME}

L'heure à laquelle la valeur de l'élément a été collectée.
 Affichée au format `HHMMSS`.
 Le fuseau horaire du serveur est utilisé dans les noms de déclencheur (dans la liste *Monitoring* > *Problems*), les noms d'événement, ainsi que les noms et valeurs des balises. Dans tous les autres cas, le fuseau horaire de l'utilisateur est utilisé.

Cette macro peut être utilisée avec un index numérique, par exemple {ITEM.VALUE.TIME<1-9>} pour désigner le premier, le deuxième, le troisième, etc. élément dans une expression de déclencheur. Voir macros indexées.

{ITEM.VALUE.TIMESTAMP}

L'horodatage UNIX auquel la valeur de l'élément a été collectée.

 Cette macro peut être utilisée avec un index numérique, par exemple {ITEM.VALUE.TIMESTAMP<1-9>} pour désigner le premier, le deuxième, le troisième, etc. élément dans une expression de déclencheur. Voir macros indexées.

{ITEM.VALUETYPE}

Le type de valeur du N-ième élément dans l'expression du déclencheur qui a provoqué une notification.
 Valeurs possibles : *0* - Nombre flottant, *1* - Caractère, *2* - Journal, *3* - Nombre entier non signé, *4* - Texte.

 Cette macro peut être utilisée avec un index numérique, par exemple {ITEM.VALUETYPE<1-9>} pour désigner le premier, le deuxième, le troisième, etc. élément dans une expression de déclencheur. Voir macros indexées.

Macros de notification de découverte de bas niveau

{LLDRULE.DESCRIPTION}

La description de la règle de découverte de bas niveau qui a provoqué une notification.

{LLDRULE.DESCRIPTION.ORIG}

La description d'origine (avec les macros non résolues) de la règle de découverte de bas niveau ayant provoqué une notification.

{LLDRULE.ID}

L'ID numérique de la règle de découverte de bas niveau qui a provoqué une notification.

{LLDRULE.KEY}

La clé de la règle de découverte de bas niveau qui a déclenché une notification.

{LLDRULE.KEY.ORIG}

La clé d'origine (avec les macros non résolues) de la règle de découverte de bas niveau qui a provoqué une notification.

{LLDRULE.NAME}

Le nom de la règle de découverte de bas niveau qui a déclenché une notification.

{LLDRULE.NAME.ORIG}

Le nom d'origine (avec les macros non résolues) de la règle de découverte de bas niveau ayant déclenché une notification.

{LLDRULE.STATE}

Le dernier état de la règle de découverte de bas niveau.
 Valeurs possibles: *Not supported*, *Normal*.

 Cette macro peut être utilisée avec un index numérique, par exemple {LLDRULE.STATE<1-9>} pour désigner le premier, le deuxième, le troisième, etc. élément dans une expression de déclencheur. Voir indexed macros.

{LLDRULE.STATE.ERROR}

Le message d'erreur contenant les détails expliquant pourquoi la règle de découverte de bas niveau est devenue non prise en charge.
 Si une règle de découverte de bas niveau passe à l'état non pris en charge puis redevient immédiatement prise en charge, le champ d'erreur peut être vide.

Macros de carte

{MAP.ID}

L'ID de la carte réseau.

{MAP.NAME}

Le nom de la carte réseau.

Macros de proxy

{PROXY.DESCRIPTION}

La description du proxy.
 Résout l'une des valeurs suivantes :

- Proxy du nième élément dans l'expression du déclencheur (dans les notifications basées sur les déclencheurs). Vous pouvez utiliser ici des macros indexées ;
- Proxy qui a exécuté la découverte (dans les notifications de découverte). Utilisez {PROXY.DESCRIPTION} ici, sans indexation ;
- Proxy auquel un agent actif s'est enregistré (dans les notifications d'auto-enregistrement). Utilisez {PROXY.DESCRIPTION} ici, sans indexation.

Cette macro peut être utilisée avec un index numérique, par exemple {PROXY.DESCRIPTION<1-9>} pour désigner le premier, le deuxième, le troisième, etc. élément dans une expression de déclencheur. Voir les macros indexées.

{PROXY.NAME}

Le nom du proxy.
 Se résout en l'un des cas suivants :

- Proxy du nième élément dans l'expression du déclencheur (dans les notifications basées sur un déclencheur). Vous pouvez utiliser ici des macros indexées ;
- Proxy qui a exécuté la découverte (dans les notifications de découverte). Utilisez {PROXY.NAME} ici, sans indexation ;
- Proxy auquel un agent actif s'est enregistré (dans les notifications d'enregistrement automatique). Utilisez {PROXY.NAME} ici, sans indexation.

Cette macro peut être utilisée avec un index numérique, par exemple {PROXY.NAME<1-9>} pour désigner le premier, le deuxième, le troisième, etc. élément dans une expression de déclencheur. Voir les macros indexées.

Macros de script

{MANUALINPUT}

La valeur d'entrée manuelle spécifiée par l'utilisateur au moment de l'exécution du script.

Macros de service

{SERVICE.DESCRIPTION}

La description du service avec les macros résolues.

{SERVICE.ID}

L'identifiant numérique du service qui a déclenché une action.

{SERVICE.NAME}

Le nom du service avec les macros résolues.

{SERVICE.ROOTCAUSE}

La liste des événements de problème de déclencheur ayant provoqué la défaillance d'un service, triée par gravité et par nom d'hôte.
 Comprend les détails suivants : nom d'hôte, nom de l'événement, gravité, ancienneté, balises de service et valeurs.

{SERVICE.TAGS}

Une liste de balises d'événement de service séparées par des virgules.
 Les balises d'événement de service peuvent être définies dans la section des balises de configuration du service. Développé en une chaîne vide si aucune balise n'existe.

{SERVICE.TAGSJSON}

Un tableau JSON contenant des objets de balises d'événement de service.
 Les balises d'événement de service peuvent être définies dans la section des balises de configuration du service. Développé en un tableau vide si aucune balise n'existe.

{SERVICE.TAGS.<tag name>}

La valeur de la balise d'événement de service référencée par le nom de la balise.
 Les balises d'événement de service peuvent être définies dans la section des balises de configuration du service.
 Un nom de balise contenant des caractères non alphanumériques (y compris des caractères multioctets UTF non anglais) doit être placé entre guillemets doubles. Les guillemets et les antislashs à l'intérieur d'un nom de balise entre guillemets doivent être échappés avec un antislash.

Macros de déclencheur

{TRIGGER.DESCRIPTION}

La description du déclencheur.
 Toutes les macros prises en charge dans une description de déclencheur seront développées si `{TRIGGER.DESCRIPTION}` est utilisée dans le texte de notification.

{TRIGGER.EVENTS.ACK}

Le nombre d'événements acquittés pour un élément de carte dans les cartes, ou pour le déclencheur ayant généré l'événement actuel dans les notifications.

{TRIGGER.EVENTS.PROBLEM.ACK}

Le nombre d'événements de problème acquittés pour tous les déclencheurs, sans tenir compte de leur état.

{TRIGGER.EVENTS.PROBLEM.UNACK}

Le nombre d'événements de problème non acquittés pour tous les déclencheurs, sans tenir compte de leur état.

{TRIGGER.EVENTS.UNACK}

Le nombre d'événements non acquittés pour un élément de carte dans les cartes, ou pour le déclencheur qui a généré l'événement actuel dans les notifications.

{TRIGGER.EXPRESSION}

L'expression du déclencheur.

{TRIGGER.EXPRESSION.EXPLAIN}

Une expression de déclencheur partiellement évaluée.
 Les fonctions basées sur des éléments sont évaluées et remplacées par leurs résultats au moment de la génération de l'événement, tandis que toutes les autres fonctions sont affichées telles qu'elles sont écrites dans l'expression. Peut être utilisé pour déboguer les expressions de déclencheur.

{TRIGGER.EXPRESSION.RECOVERY}

L'expression de récupération du déclencheur si la *génération d'événements OK* dans la configuration du déclencheur est définie sur « Expression de récupération » ; sinon, une chaîne vide est renvoyée.

{TRIGGER.EXPRESSION.RECOVERY.EXPLAIN}

Une expression de récupération de déclencheur partiellement évaluée.
 Les fonctions basées sur des éléments sont évaluées et remplacées par les résultats au moment de la génération de l'événement, tandis que toutes les autres fonctions sont affichées telles qu'elles sont écrites dans l'expression. Peut être utilisée pour déboguer les expressions de récupération de déclencheur.

{TRIGGER.HOSTGROUP.NAME}

Une liste triée (par requête SQL), séparée par des virgules et des espaces, des groupes d'hôtes dans lesquels le déclencheur est défini.

{TRIGGER.ID}

L'identifiant numérique du déclencheur qui a déclenché cette action.

{TRIGGER.NAME}

Le nom du déclencheur avec les macros résolues.
 Notez que `{EVENT.NAME}` peut être utilisé dans les actions pour afficher le nom de l'événement/problème déclenché avec les macros résolues.

{TRIGGER.NAME.ORIG}

Le nom d'origine du déclencheur (avec les macros non résolues).

{TRIGGER.NSEVERITY}

La sévérité numérique du déclencheur.
 Valeurs possibles : *0* - Non classé, *1* - Information, *2* - Avertissement, *3* - Moyenne, *4* - Haute, *5* - Désastre.

{TRIGGER.PROBLEM.EVENTS.PROBLEM.ACK}

Le nombre d'événements de problème reconnus pour les déclencheurs à l'état de problème.

{TRIGGER.PROBLEM.EVENTS.PROBLEM.UNACK}

Le nombre d'événements de problème non acquittés pour les déclencheurs en état de problème.

{TRIGGER.SEVERITY}

Le nom de la sévérité du déclencheur.
 Peut être défini dans *Administration > General > Trigger displaying options*.

{TRIGGER.STATE}

Le dernier état de l'expression du déclencheur.
 Valeurs possibles : *Inconnu*, *Normal*.

{TRIGGER.STATE.ERROR}

Le message d'erreur contenant les détails expliquant pourquoi un déclencheur est devenu non pris en charge.
 Si un déclencheur passe à l'état non pris en charge puis redevient immédiatement pris en charge, le champ d'erreur peut être vide.

{TRIGGER.STATUS}

La valeur du déclencheur au moment de l'exécution de l'étape de l'opération.
 Valeurs possibles : *Problem*, *OK*.

{TRIGGER.TEMPLATE.NAME}

Une liste triée (par requête SQL), séparée par des virgules et des espaces, des modèles dans lesquels le déclencheur est défini, ou *UNKNOWN* si le déclencheur est défini dans un hôte.

{TRIGGER.URL}

L'URL du déclencheur.

{TRIGGER.URL.NAME}

Le libellé de l'URL du déclencheur.

{TRIGGER.VALUE}

La valeur numérique actuelle du déclencheur.
 Valeurs possibles : *0* - le déclencheur est dans l'état OK, *1* - le déclencheur est dans l'état Problème.

{TRIGGERS.ACK}

Le nombre de déclencheurs acquittés pour un élément de carte, sans tenir compte de l'état du déclencheur.
 Un déclencheur est considéré comme acquitté si tous ses événements de problème sont acquittés.

{TRIGGERS.PROBLEM.ACK}

Le nombre de déclencheurs de problème acquittés pour un élément de carte.
 Un déclencheur est considéré comme acquitté si tous ses événements de problème sont acquittés.

{TRIGGERS.PROBLEM.UNACK}

Le nombre de déclencheurs de problème non acquittés pour un élément de carte.
 Un déclencheur est considéré comme non acquitté si au moins un de ses événements de problème n'est pas acquitté.

{TRIGGERS.UNACK}

Le nombre de déclencheurs non acquittés pour un élément de carte, sans tenir compte de l'état du déclencheur.
 Un déclencheur est considéré comme non acquitté si au moins un de ses événements de problème n'est pas acquitté.

Macros de nom d'utilisateur

{USER.FULLNAME}

Le nom, le prénom et le nom d'utilisateur de l'utilisateur qui a ajouté l'accusé de réception de l'événement ou lancé le script.

{USER.NAME}

Le nom de l'utilisateur qui a lancé le script.

{USER.SURNAME}

Le nom de famille de l'utilisateur qui a lancé le script.

{USER.USERNAME}

Le nom d'utilisateur de l'utilisateur qui a lancé le script.

Macros indexées

La syntaxe des macros indexées **{MACRO<1-9>}** ne peut référencer que le N-ième élément, fonction ou hôte dans le champ *Expression* d'un déclencheur :

- **{HOST.IP1}, {HOST.IP2}, {HOST.IP3}** sont résolues en adresses IP du premier, du deuxième et du troisième hôte dans l'expression du déclencheur (si présents).
- **{ITEM.VALUE1}, {ITEM.VALUE2}, {ITEM.VALUE3}** sont résolues en valeurs du premier, du deuxième et du troisième élément dans l'expression du déclencheur au moment de l'événement (si présents).
- **{FUNCTION.VALUE1}, {FUNCTION.VALUE2}, {FUNCTION.VALUE3}** sont résolues en valeurs des première, deuxième et troisième fonctions basées sur des éléments au moment de l'événement (si présentes).

Dans le contexte des déclencheurs, les macros indexées font toujours référence au champ *Expression* de la configuration du déclencheur, et non à l'*Expression de récupération*. Par exemple, dans un événement de récupération, **{ITEM.VALUE2}** sera résolue en la valeur du deuxième élément de l'expression du problème au moment de la récupération.

La macro **{HOST.HOST<1-9>}** est également prise en charge dans la macro d'expression `{?func(/host/key,param)}` dans les **noms de graphique**. Par exemple, `{?func(/{HOST.HOST2}/key,param)}` dans un nom de graphique sera résolu en l'hôte du deuxième élément du graphique.

Warning:
Les macros indexées ne seront résolues dans aucun autre contexte que les cas mentionnés ici. Pour les autres contextes, utilisez des macros sans index (**{HOST.HOST}, {HOST.IP}**, etc.).

Macros utilisateur pris en charge par emplacement

Aperçu

Cette section contient une liste des emplacements où les macros définies par l'utilisateur sont prises en charge.

Note:
Seules les macros utilisateur de niveau global sont prises en charge pour les *Actions*, la *Découverte réseau*, les *Proxys* et tous les emplacements répertoriés dans la section *Autres emplacements* de cette page. Dans les emplacements mentionnés, les macros de niveau hôte et de niveau modèle ne seront pas résolues.

Note:
Pour personnaliser les valeurs des macros (par exemple, raccourcir ou extraire des sous-chaînes spécifiques), vous pouvez utiliser les fonctions de macro.

Actions

Dans les actions, les macros utilisateur peuvent être utilisées dans les champs suivants :

Emplacement		Plusieurs macros/mix avec du texte[1]
Notifications et commandes basées sur les déclencheurs		oui
Notifications internes basées sur les déclencheurs		oui
Notifications de mise à jour de problème		oui
Notifications et commandes basées sur les services		oui
Notifications de mise à jour des services		oui
Condition de période de temps		non
Opérations		
	Durée de l'étape d'opération par défaut	non
	Durée de l'étape	non

Hôtes/prototypes d'hôte

Dans la configuration d'un hôte et d'un prototype d'hôte, les macros utilisateur peuvent être utilisées dans les champs suivants :

Emplacement		Macros multiples/mélange avec du texte[1]
IP/DNS de l'interface		DNS uniquement
Port de l'interface		non
SNMP v1, v2		
	Communauté SNMP	oui
SNMP v3		
	Nom de contexte	oui
	Nom de sécurité	oui
	Phrase secrète d'authentification	oui
	Phrase secrète de confidentialité	oui
IPMI		
	Nom d'utilisateur	oui
	Mot de passe	oui
Tags[2]		
	Noms des tags	oui
	Valeurs des tags	oui

Éléments / prototypes d'éléments

Dans la configuration d'un élément ou d'un prototype d'élément, les macros utilisateur peuvent être utilisées dans les champs suivants :

Emplacement	Plusieurs macros/mélange avec du texte[1]
Nom de l'élément	oui
Paramètres de la clé de l'élément	oui[4]
Intervalle de mise à jour	non
Intervalles per-son-nal-isés	non

Emplacement	Plusieurs macros/mélange avec du texte[1]
Délai d'expiration (disponible pour les types d'éléments pris en charge)	non
Stocker jusqu'à (pour l'historique et les tendances)	non
Description	oui
Élément calculé/agrégé	
Formule (constantes d'expression et paramètres de fonction ; paramètres de clé de l'élément ; (*élément agrégé uniquement*) conditions de filtrage (nom du groupe d'hôtes et nom de l'étiquette))	oui
Moniteur de base de données	
Nom d'utilisateur	oui
Mot de passe	oui
Requête SQL	oui
Agent HTTP	
URL[3]	oui
Champs de requête	oui
Corps de la requête	oui
En-têtes (noms et valeurs)	oui
Codes d'état requis	oui
Proxy HTTP	oui
Nom d'utilisateur pour l'authentification HTTP	oui
Mot de passe pour l'authentification HTTP	oui
Fichier de certificat SSl	oui
Fichier de clé SSl	oui
Mot de passe de la clé SSl	oui
Hôtes autorisés	oui
Agent JMX	
Point de terminaison JMX	oui
Élément de type script	
Noms et valeurs des paramètres	oui

Emplacement	Plusieurs macros/mélange avec du texte[1]
Élément de type navigateur	
Noms et valeurs des paramètres	oui
Élément de requête de télémétrie	
Décalage temporel	non
Limite de rétrospection	non
Granularité	non
Agent SNMP	
OID SNMP	oui
Agent SSH	
Nom d'utilisateur	oui
Fichier de clé publique	oui
Fichier de clé privée	oui
Mot de passe	oui
Script	oui
Agent TELNET	
Nom d'utilisateur	oui
Mot de passe	oui
Script	oui
Trappeur Zabbix	
Hôtes autorisés	oui
Étiquettes[2]	
Noms des étiquettes	oui
Valeurs des étiquettes	oui
Étapes de prétraitement	
Paramètres (y compris les scripts personnalisés)	oui
Paramètres personnalisés de gestion des erreurs (champs *Définir la valeur sur* et *Définir l'erreur sur*)	oui

Découverte de bas niveau

Dans une règle de découverte de bas niveau, les macros utilisateur peuvent être utilisées dans les champs suivants :

Emplacement	Macros multiples/mélange avec du texte[1]
Paramètres de clé	oui

Emplacement	Macros multiples/mélange avec du texte[1]
Intervalle de mise à jour	non
Intervalle per-son-nal-isé	non
Délai d’expiration (disponible pour les types d’élément pris en charge)	non
Supprimer les ressources per-dues	non
Désactiver les ressources per-dues	non
Description	oui
agent SNMP	
OID SNMP	oui
agent SSH	
Nom d’utilisateur	oui
Fichier de clé publique	oui
Fichier de clé privée	oui
Mot de passe	oui
Script	oui
agent TEL-NET	
Nom d’utilisateur	oui
Mot de passe	oui
Script	oui
Zabbix trap-per	
Hôtes autorisés	oui
Moniteur de base de don-nées	
Nom d’utilisateur	oui
Mot de passe	oui
Requête SQL	oui

Emplacement	Macros multiples/mélange avec du texte[1]
agent	
JMX	
Point de terminaison JMX	oui
agent	
HTTP	
URL[3]	oui
Champs de requête	oui
Corps de la requête	oui
En-têtes (noms et valeurs)	oui
Codes d'état requis	oui
Nom d'utilisateur pour l'authentification HTTP	oui
Mot de passe pour l'authentification HTTP	oui
Filtres	
Expression régulière	oui
Remplacements	
Filtres : expression régulière	oui
Opérations : intervalle de mise à jour (pour les prototypes d'élément)	non
Opérations : période de stockage de l'historique (pour les prototypes d'élément)	non
Opérations : période de stockage des tendances (pour les prototypes d'élément)	non

Découverte réseau

Dans une règle de découverte de réseau, les macros utilisateur peuvent être utilisées dans les champs suivants :

Emplacement		Plusieurs macros/mix avec du texte[1]
Intervalle de mise à jour		non
SNMP v1, v2		
	Communauté SNMP	oui
	OID SNMP	oui
SNMP v3		
	Nom du contexte	oui
	Nom de sécurité	oui
	Phrase secrète d'authentification	oui
	Phrase secrète de confidentialité	oui
	OID SNMP	oui

Proxies

Dans une configuration de proxy, les macros utilisateur peuvent être utilisées dans les champs suivants :

Location	Multiple macros/mix with text[1]
Address for active agents > *Port* (if the proxy belongs to a group)	no
Interface address and port (for passive proxy)	no
Timeouts for item types	no

Groupes de proxy

Dans une configuration de groupe de proxy, les macros utilisateur peuvent être utilisées dans les champs suivants :

Emplacement	Macros multiples/mélange avec du texte[1]
Période de basculement	non
Nombre minimum de proxy	non

Modèles

Dans une configuration de modèle, les macros utilisateur peuvent être utilisées dans les champs suivants :

Emplacement		Plusieurs macros/mix avec du texte[1]
Tags[2]		
	Noms de Tag	oui
	Valeurs de Tag	oui

Déclencheurs

Dans la configuration d'un déclencheur, les macros utilisateur peuvent être utilisées dans les champs suivants :

Emplacement	Macros multiples/mélange avec du texte[1]
Nom	oui
Données opérationnelles	oui
Expression (uniquement dans les constantes et les paramètres de fonction ; les macros secrètes ne sont pas prises en charge)	oui
Tag pour la correspondance	oui
Nom de l'entrée de menu	oui
URL de l'entrée de menu[3]	oui
Description	oui
Tags[2]	
Noms des tags	oui
Valeurs des tags	oui

Scénario Web

Dans une configuration de scénario Web, les macros utilisateur peuvent être utilisées dans les champs suivants :

Emplacement		Plusieurs macros/mix avec du texte[1]
Nom		oui
Intervalle de mise à jour		non
Agent		oui
Proxy HTTP		oui
Variables (valeurs uniquement)		oui
En-têtes (noms et valeurs)		oui
Étapes		
	Nom	oui
	URL[3]	oui
	Variables (valeurs uniquement)	oui
	En-têtes (noms et valeurs)	oui
	Timeout	non
	Chaîne requise	oui
	Codes d'état requis	non
Authentification		
	Utilisateur	oui
	Mot de passe	oui
	Certificat SSL	oui
	Fichier de clé SSL	oui
	Mot de passe de la clé SSL	oui
Tags[2]		
	Noms des tags	oui
	Valeurs des tags	oui

Autres emplacements

En plus des emplacements répertoriés ici, les macros utilisateur peuvent être utilisées dans les champs suivants :

Emplacement	Macros multiples/mélange avec du texte[1]
Scripts globaux (URL, script, SSH, Telnet, IPMI), y compris le texte de confirmation	yes
Webhooks	
Script JavaScript	no
Nom du paramètre du script JavaScript	no
Valeur du paramètre du script JavaScript	yes
Tableaux de bord	
Colonne de type de données *Texte* dans le widget de tableau de bord *Top hosts*	yes
Paramètre *Description* dans le widget de tableau de bord *Item value* et *Gauge*	yes
Paramètre *Text* de l'étiquette principale/secondaire dans le widget de tableau de bord *Honeycomb*	yes

Emplacement	Macros multiples/mélange avec du texte[1]
Paramètre *URL*[3] dans le widget de tableau de bord *URL*	yes
Surveillance > *Problèmes*, *Hôtes*, *Dernières données*	
Colonne *Texte personnalisé*	yes
Collecte de données > *Modèles*, *Hôtes*	
Colonne *Texte personnalisé*	yes
Alertes > *Types de média* > *Modèles de message*	
Objet	yes
Message	yes
Alertes > *Types de média* > *Script*	
Paramètres du script	yes
Alertes > *Types de média* > *Type de média*	
Champs *Username* et *Password* pour le type de média *Email* (lorsque *Authentication* est défini sur ”Username and password”; macros secrètes recommandées)	yes

Emplacement	Macros multiples/mélange avec du texte[1]
Utilisateurs > *Utilisateurs* > *Média*	
Lorsque actif	yes
Administration > *Général* > *GUI*	
Temps de travail	no
Administration > *Général* > *Délais d'attente*	
Délais d'attente pour les types d'élément	no
Administration > *Général* > *Connecteurs*	
URL	yes
Nom d'utilisateur	yes
Mot de passe	yes
Jeton Bearer	yes
Délai d'attente	no
Proxy HTTP	yes
Fichier de certificat SSL	yes
Fichier de clé SSL	yes
Mot de passe de la clé SSL	yes

Pour obtenir la liste complète de toutes les macros prises en charge dans Zabbix, consultez les macros prises en charge.

Notes de bas de page

[1] Si plusieurs macros dans un champ ou des macros mélangées avec du texte ne sont pas pris en charge pour l'emplacement, une seule macro doit remplir tout le champ.

[2] Les macros utilisées dans les noms et les valeurs des balises ne sont résolues que pendant le processus de génération des événements.

[3] Les URL qui contiennent une macro secrète ne fonctionneront pas, car la macro qu'elles contiennent sera résolue en "******".

[4] Les macros ne sont pas prises en charge dans la clé d'élément des éléments Zabbix trapper.

Compatibilité des versions

Agents pris en charge

Pour être compatible avec Zabbix 8.0, l'agent Zabbix ne doit pas être antérieur à la version 1.6 ni ultérieur à la version 8.0.

Vous devrez peut-être vérifier la configuration des agents plus anciens, car certains paramètres ont changé, par exemple les paramètres liés à la journalisation pour les versions antérieures à la version 3.0.

Pour tirer pleinement parti des dernières fonctionnalités, des métriques, des performances améliorées et de la réduction de l'utilisation de la mémoire, utilisez la dernière version prise en charge de l'agent.

Remarques pour Windows XP

- Sur Windows XP 32 bits, n'utilisez pas d'agents Zabbix plus récents que la version 6.0.x ;
- Sur Windows XP/Server 2003, n'utilisez pas de modèles d'agent plus récents que Zabbix 4.0.x. Les modèles plus récents utilisent des compteurs de performances en anglais, qui ne sont pris en charge que depuis Windows Vista/Server 2008.

Agents 2 pris en charge

Les anciennes versions de Zabbix agent 2 à partir de la version 4.4 sont compatibles avec Zabbix 8.0 ; Zabbix agent 2 ne doit pas être plus récent que 8.0.

Notez que lors de l'utilisation de Zabbix agent 2 versions 4.4 et 5.0, l'intervalle par défaut de 10 minutes est utilisé pour l'actualisation des éléments non pris en charge.

Pour tirer pleinement parti des dernières fonctionnalités, des métriques, des performances améliorées et de la réduction de l'utilisation de la mémoire, utilisez la dernière version prise en charge de agent 2.

Proxys Zabbix pris en charge

Pour une compatibilité totale avec Zabbix 8.0, les proxys doivent correspondre à la version majeure du serveur. Seuls les proxys Zabbix 8.0.x sont entièrement compatibles avec un serveur Zabbix 8.0.x.

Les proxys obsolètes sont partiellement pris en charge : ils peuvent toujours collecter des données et exécuter des scripts, mais ne peuvent pas recevoir de mises à jour de configuration, telles que de nouveaux éléments.

Par rapport au serveur Zabbix, les proxies peuvent être :

- *Actuels* (le proxy et le serveur ont la même version majeure) ;
- *Obsolètes* (la version du proxy est antérieure à celle du serveur, mais elle est partiellement prise en charge) ;
- *Non pris en charge* (la version du proxy est antérieure à la version de la précédente version LTS du serveur *ou* la version du proxy est plus récente que la version majeure du serveur).

Exemples :

Version du serveur	Version de proxy *actuelle*	Version de proxy *obsolète*	Version de proxy *non prise en charge*
6.4	6.4	6.0, 6.2	Antérieure à 6.0 ; supérieure à 6.4
7.0	7.0	6.0, 6.2, 6.4	Antérieure à 6.0 ; supérieure à 7.0
7.2	7.2	7.0	Antérieure à 7.0 ; supérieure à 7.2
7.4	7.4	7.0	Antérieure à 7.0 ; supérieure à 7.4
8.0	8.0	7.0, 7.2, 7.4	Antérieure à 7.0 ; supérieure à 8.0

Fonctionnalités prises en charge par les proxies :

Version du proxy	Mise à jour des données	Mise à jour de la configuration	Tâches
Current	Yes	Yes	Yes
Outdated	Yes	No	Commandes à distance (par exemple, scripts shell); Vérifications immédiates de la valeur de l'élément (c.-à-d. *Exécuter maintenant*); Remarque: les tests de prétraitement avec une valeur réelle ne sont pas pris en charge.
Unsupported	No	No	No

Des avertissements concernant l'utilisation de versions incompatibles du démon Zabbix sont consignés dans le journal.

Fichiers XML pris en charge

Les fichiers XML dont la version n'est pas antérieure à 1.8 sont pris en charge pour l'importation dans Zabbix 8.0.

Attention:
Dans le format d'export XML, les dépendances des déclencheurs sont stockées uniquement par nom. S'il existe plusieurs déclencheurs portant le même nom (par exemple, avec des sévérités et des expressions différentes) pour lesquels une dépendance est définie entre eux, il n'est pas possible de les importer. Ces dépendances doivent être supprimées manuellement du fichier XML puis rajoutées après l'importation.

Syntaxe de la période de temps

Aperçu

Pour définir une période, le format suivant doit être utilisé :

`d-d,hh:mm-hh:mm`

où les symboles signifient ce qui suit :

Symbole	Description
d	Jour de la semaine : 1 - lundi, 2 - mardi, ... , 7 - dimanche
hh	Heures : 00-24
mm	Minutes : 00-59

Vous pouvez spécifier plus d'une période en utilisant un séparateur point-virgule (;) :

`d-d,hh:mm-hh:mm;d-d,hh:mm-hh:mm...`

Laisser la période vide équivaut à 1-7,00:00-24:00, qui est la valeur par défaut.

Attention:
La limite supérieure d'une période n'est pas incluse. Ainsi, si vous spécifiez 09:00-18:00, la dernière seconde incluse dans la période est 17:59:59.

Exemples

Heures d'ouverture. Lundi - Vendredi de 9h00 à 18h00 :

`1-5,09:00-18:00`

Horaires de travail plus week-end. Lundi - Vendredi de 9h00 à 18h00 et Samedi, Dimanche de 10h00 à 16h00 :

`1-5,09:00-18:00;6-7,10:00-16:00`

Symboles d'unité

Aperçu

Travailler avec de grandes valeurs telles que 86400, 104857600 ou 1000000 peut être difficile et entraîner des erreurs. Pour simplifier la configuration et améliorer la lisibilité, Zabbix prend en charge les symboles d'unité (suffixes) qui servent de multiplicateurs de valeur.

Expressions de déclencheur sans suffixes :

```
last(/host/system.uptime)<86400
avg(/host/system.cpu.load,600s)<10
last(/host/vm.memory.size[available])<20971520
```

Expressions de déclencheur avec suffixes :

```
last(/host/system.uptime)<1d
avg(/host/system.cpu.load,10m)<10
last(/host/vm.memory.size[available])<20M
```

Les suffixes peuvent également simplifier la configuration d'autres entités, telles que les éléments, les widgets, etc., et aider à afficher les valeurs des éléments dans un format lisible par l'homme.

> **Note:**
> Pour savoir si un champ de configuration d'entité prend en charge les suffixes, consultez toujours la page correspondante à l'entité en cours de configuration.

Suffixes de temps

L'interface Zabbix prend en charge les suffixes de temps suivants dans la configuration des entités :
 s - secondes *(lorsqu'il est utilisé, il fonctionne de la même manière que la valeur brute)*;
 m - minutes;
 h - heures;
 d - jours;
 w - semaines;
 M - mois *(uniquement pour les fonctions de tendance)*;
 y - années *(uniquement pour les fonctions de tendance)*.

Les suffixes de temps ne sont pris en charge qu'avec des nombres entiers. Par exemple, 1h est pris en charge, mais 1,5h ou 1.5h ne sont pas pris en charge ; utilisez 90m à la place.

Suffixes de taille de mémoire

Zabbix prend en charge les suffixes de taille de mémoire suivants :
 K - kilooctets
 M - mégaoctets
 G - gigaoctets
 T - téraoctets

Suffixes de valeur d'élément

Les suffixes peuvent également être utilisés pour afficher les valeurs numériques des éléments dans un format lisible par l'humain.

Pour activer cette fonctionnalité, utilisez l'un des suffixes suivants dans le champ *Units* lors de la configuration d'un élément :
 B - octets;
 Bps - octets par seconde;
 s - secondes, affichées en utilisant jusqu'à trois unités de temps non nulles les plus grandes;
 uptime - temps écoulé au format hh:mm:ss ou N days, hh:mm:ss;
 unixtime - horodatage Unix, formaté comme yyyy.mm.dd hh:mm:ss.

Les règles supplémentaires suivantes s'appliquent à la manière dont ces suffixes sont interprétés et affichés :

- Pour B et Bps, Zabbix utilise une conversion en base 2 (1K = 1024B), conformément à la norme JEDEC.
- Pour les autres unités (telles que Hz, W, etc.), Zabbix utilise une conversion en base 10 (1K = 1000).
- Pour s (secondes) :
  - Le format inclut yyy mmm ddd hhh mmm sss ms ; seules jusqu'à trois unités de temps non nulles les plus grandes sont affichées (par exemple, 1M 10d 4h).
  - Si une unité est nulle et se trouve entre deux unités non nulles, elle est omise (par exemple, 10d 56m au lieu de 10d 0h 56m).

Lorsque *Units* sont utilisés, les suffixes multiplicateurs suivants sont automatiquement appliqués aux valeurs d'élément :
 K, **M**, **G**, **T** - kilo, méga, giga, téra;
 P, **E**, **Z**, **Y** - péta, exa, zetta, yotta *(ils sont appliqués uniquement dans l'interface)*.

Pour empêcher la conversion d'unité, utilisez le préfixe `!` (par exemple, !B ou !s).

Les exemples suivants montrent comment les valeurs d'élément reçues sont converties en fonction des unités spécifiées :

```
1 B → 1 B
1024 B → 1 KB
1536 B → 1.5 KB
881764 B → 881.76 KB
881764 !B → 881764 B

0.0000155 s → 0.016ms
3470400 s → 1M 10d 4h
2606400 s → 1M 4h
2592000 s → 1M
2592001 s → 1M
2592001 !s → 2592001 s

17764 uptime → 04:56:04
86400 uptime → 1 day, 00:00:00
881764 uptime → 10 days, 04:56:04
32417764 uptime → 375 days, 04:56:04
32417764 !uptime → 32417764 uptime

881764 unixtime → 1970-01-11 04:56:04 AM
```

```
17764 Hz → 17.76 KHz
86400 Hz → 86.4 KHz
881764 Hz → 881.76 KHz
32417764 Hz → 32.42 MHz

0 ! → 0
0 !! → 0 !
```

Note:
Avant Zabbix 4.0, il existait une liste d'exclusion d'unités codée en dur, composée de `ms`, `rpm`, `RPM`, `%`. Cette liste d'exclusion est désormais obsolète ; la bonne manière d'empêcher la conversion de ces unités est donc `!ms`, `!rpm`, `!RPM`, `!%`.

Exécution de commandes

Zabbix utilise une fonctionnalité commune pour les vérifications externes, les paramètres utilisateur, les éléments system.run, les scripts d'alerte personnalisés, les commandes à distance et les scripts globaux.

Étapes d'exécution

Note:
Par défaut, tous les scripts dans Zabbix sont exécutés à l'aide du shell *sh*, il n'est pas possible de modifier le shell par défaut. Pour utiliser un autre shell, vous pouvez utiliser une solution de contournement : créez un fichier script et appelez ce script pendant l'exécution de la commande.

La commande/le script est exécuté de la même manière sur les plates-formes Unix et Windows :

1. Zabbix (le processus parent) crée un canal pour la communication
2. Zabbix définit le canal comme sortie pour le processus enfant à créer
3. Zabbix crée le processus enfant (exécute la commande/le script)
4. Un nouveau groupe de processus (sous Unix) ou une tâche (sous Windows) est créé pour le processus enfant
5. Zabbix lit à partir du canal jusqu'à ce que le délai d'attente se produise ou que personne n'écrive à l'autre extrémité (TOUS les handles/descripteurs de fichiers ont été fermés). Notez que le processus enfant peut créer plus de processus et quitter avant qu'ils ne quittent ou ne ferment le handle/descripteur de fichier.
6. Si le délai d'attente n'a pas été atteint, Zabbix attend que le processus enfant initial se termine ou que le délai d'attente se produise
7. Si le processus enfant initial s'est terminé et que le délai d'attente n'a pas été atteint, Zabbix vérifie le code de sortie du processus enfant initial et le compare à 0 (une valeur différente de zéro est considérée comme un échec d'exécution, uniquement pour les scripts d'alerte personnalisés, les commandes distantes et les scripts utilisateur exécutés sur le serveur Zabbix et le proxy Zabbix)
8. À ce stade, il est supposé que tout est terminé et que l'ensemble de l'arborescence des processus (c'est-à-dire le groupe de processus ou le travail) est terminé

Attention:
Zabbix suppose qu'une commande/un script a terminé le traitement lorsque le processus enfant initial est terminé ET qu'aucun autre processus ne maintient toujours le handle/descripteur de fichier de sortie ouvert. Lorsque le traitement est terminé, TOUS les processus créés sont terminés.

Tous les guillemets doubles et les barres obliques inverses de la commande sont échappés avec des barres obliques inverses et la commande est entourée de guillemets doubles.

Vérification du code de sortie

Les codes de sortie sont vérifiés selon les conditions suivantes :

- Uniquement pour les scripts d'alerte personnalisés, les commandes à distance et les scripts utilisateur exécutés sur le serveur Zabbix et le proxy Zabbix.
- Tout code de sortie différent de 0 est considéré comme un échec d'exécution.
- Le contenu de la sortie d'erreur standard et de la sortie standard pour les exécutions échouées est collecté et disponible dans l'interface (où le résultat de l'exécution est affiché).
- Une entrée de journal supplémentaire peut être créée pour les commandes à distance exécutées sur Zabbix agent/proxy en activant le paramètre LogRemoteCommands sur agent/proxy.

Messages possibles dans l'interface et entrées de journal pour les commandes/scripts en échec :

- Le contenu de la sortie d'erreur standard et de la sortie standard pour les exécutions échouées (le cas échéant).
- "Process exited with code: N." (pour une sortie vide, et un code de sortie différent de 0).
- "Process killed by signal: N." (pour un processus terminé par un signal, sous Linux uniquement).
- "Process terminated unexpectedly." (pour un processus terminé pour des raisons inconnues).

Voir aussi

- Vérifications externes
- Paramètres utilisateur
- éléments system.run
- Scripts d'alerte personnalisés
- Commandes distantes
- Scripts globaux

Exemples d'échappement

Aperçu

Cette page fournit des exemples d'utilisation de l'échappement correct lors de l'utilisation d'expressions régulières dans différents contextes.

> **Note:**
> Lors de l'utilisation du constructeur d'expressions de déclencheur, l'échappement correct dans les expressions régulières est ajouté automatiquement.

Exemples

Macro utilisateur avec contexte

Expression régulière : `\.+\"[a-z]+<br>` Macro utilisateur avec contexte : `{$MACRO:regex:"\.+\\"[a-z]+"}`

Remarque :

- les barres obliques inverses ne sont pas échappées ;
- les guillemets sont échappés.

Fonction de macro dans le paramètre de clé d'élément

Expression régulière : `.+:(\d+)$<br>` Clé d'élément : `net.tcp.service[tcp,,"{{$ENDPOINT}.regsub(\".+:(\d+)$\",\1)}"]`

Remarque :

- l'expression régulière à l'intérieur de la fonction de macro `regsub` est entre guillemets doubles (car elle contient une parenthèse fermante) ;
- les guillemets autour de l'expression régulière sont échappés (car l'ensemble du troisième paramètre de l'élément est entre guillemets doubles) ;
- le troisième paramètre de la clé d'élément est entre guillemets doubles car il contient une virgule.

Fonction de macro LLD

Expression régulière : `\.+\"([a-z]+)<br>` Macro LLD : `{{#MACRO}.iregsub("\.+\\"([a-z]+)", \1)}`

Remarque :

- les barres obliques inverses ne sont pas échappées ;
- les guillemets sont échappés.

Fonction de macro LLD dans le contexte d'une macro utilisateur

Expression régulière : `\.+\"([a-z]+)<br>` Macro LLD : `{{#MACRO}.iregsub("\.+\\"([a-z]+)", \1)}<br>` Macro utilisateur avec contexte : `{$MACRO:"{{#MACRO}.iregsub(\"\.+\\\"[a-z]+\", \1)}"}`

Remarque :

- l'échappement des barres obliques inverses pour LLD ne change pas ;
- lors de l'insertion de la macro LLD dans le contexte d'une macro utilisateur, il faut la placer dans une chaîne :

1. Des guillemets sont ajoutés autour de l'expression de macro ;
2. Les guillemets sont échappés ; au total, 3 nouvelles barres obliques inverses sont introduites.

Paramètre de chaîne d'une fonction (quelconque)

`concat` est utilisé comme exemple.

Contenu de la chaîne : `\.+\"[a-z]+<br>` Expression : `concat("abc", "\\.\\\"[a-z]+")`

Remarque :

- Les paramètres de chaîne nécessitent l'échappement à la fois des barres obliques inverses et des guillemets.

Fonction de macro LLD dans le paramètre de chaîne d'une fonction

Expression régulière : `\.+\"([a-z]+)<br>` Macro LLD : `{{#MACRO}.iregsub("\.+\\"([a-z]+)", \1)}<br>` Expression : `concat("abc, "{{#MACRO}.iregsub(\"\\.+\\\\\"([a-z]+)\", \\1)}")`

Remarque :

- Les paramètres de chaîne nécessitent l'échappement à la fois des barres obliques inverses et des guillemets ;
- Une couche supplémentaire d'échappement est ajoutée, car la macro ne sera résolue qu'après la suppression des guillemets de la chaîne ;

Macro utilisateur avec contexte dans le paramètre de chaîne d'une fonction

Expression régulière : `\.+\"[a-z]+<br>` Macro utilisateur avec contexte : `{$MACRO:regex:"\.+\\"[a-z]+"}<br>` Expression : `concat("abc, "{$MACRO:regex:\"\\.+\\\\\"[a-z]+\"}")`

Remarque :

- Comme dans l'exemple précédent, une couche supplémentaire d'échappement est nécessaire ;
- Les barres obliques inverses et les guillemets ne sont échappés que pour l'échappement de niveau supérieur (du fait qu'il s'agit d'un paramètre de chaîne).

Fonction de macro LLD dans le contexte d'une macro utilisateur dans une fonction

Expression régulière : `\.+\"([a-z])+<br>` Macro LLD : `{{#MACRO}.iregsub("\.+\\"([a-z]+)", \1)}<br>` Macro utilisateur avec contexte : `{$MACRO:"{{#MACRO}.iregsub(\"\.+\\\"([a-z]+)\", \1)}"}<br>` Expression : `concat("abc, "{$MACRO:\"{{#MACRO}.iregsub(\\\"\.+\\\\\\\"([a-z]+)\\\", \\1)}\"}")`

Remarque sur les trois couches d'échappement :

1. Pour la fonction de macro LLD, sans échappement des barres obliques inverses ;
2. Pour la macro utilisateur avec contexte, sans échappement des barres obliques inverses ;
3. Pour le paramètre de chaîne d'une fonction, avec échappement des barres obliques inverses.

Macro utilisateur avec contexte simplement dans une chaîne

Expression régulière : `\.+\"[a-z]+<br>` Macro utilisateur avec contexte : `{$MACRO:regex:"\.+\\"[a-z]+"}<br>` Dans la chaîne d'une expression quelconque, par exemple : `func(arg1, arg2, arg3)="{$MACRO:regex:\"\\.+\\\\\"[a-z]+\"}"`

Remarque :

- Les chaînes nécessitent également l'échappement des barres obliques inverses ;
- Les chaînes nécessitent également l'échappement des guillemets ;
- Là encore, il s'agit d'un cas avec 2 niveaux d'échappement :

1. Échappement pour le contexte de macro utilisateur sans échappement des barres obliques inverses ;
2. Échappement dû au fait qu'il s'agit d'une chaîne avec échappement des barres obliques inverses.

Qu'est-ce que Zabbix

Zabbix a été créé par Alexei Vladishev et est actuellement activement développé et pris en charge par Zabbix SIA.

Zabbix est une solution de supervision distribuée open source de niveau entreprise.

Zabbix est un logiciel qui supervise de nombreux paramètres d'un réseau, ainsi que l'état et l'intégrité des serveurs, machines virtuelles, applications, services, bases de données, sites web, environnements cloud et bien plus encore. Zabbix utilise un mécanisme de notification flexible qui permet aux utilisateurs de configurer des alertes par e-mail pour pratiquement tout événement. Cela permet de réagir rapidement aux problèmes des serveurs. Zabbix offre d'excellentes fonctionnalités de création de rapports et de visualisation des données basées sur les données stockées. Cela fait de Zabbix une solution idéale pour la planification des capacités.

Zabbix prend en charge à la fois l'interrogation et les traps. Tous les rapports et statistiques de Zabbix, ainsi que les paramètres de configuration, sont accessibles via une interface web. Une interface web garantit que l'état de votre réseau et la santé de vos

serveurs peuvent être évalués depuis n'importe quel endroit. Correctement configuré, Zabbix peut jouer un rôle important dans la supervision de l'infrastructure informatique. Cela est tout aussi vrai pour les petites organisations disposant de quelques serveurs que pour les grandes entreprises possédant une multitude de serveurs.

Zabbix est disponible gratuitement. Zabbix est écrit et distribué sous licence AGPL-3.0. Cela signifie que son code source est librement distribué et accessible au grand public.

Un support commercial est disponible et fourni par Zabbix Company et ses partenaires dans le monde entier.

En savoir plus sur les fonctionnalités de Zabbix.

Utilisateurs de Zabbix De nombreuses organisations de tailles différentes dans le monde entier s'appuient sur Zabbix comme plateforme de supervision principale.

Architecture Zabbix se compose de plusieurs composants logiciels principaux. Leurs responsabilités sont décrites ci-dessous.

Serveur

Zabbix server est le composant central auquel les agent signalent les informations et statistiques de disponibilité et d'intégrité. Le serveur est le référentiel central dans lequel sont stockées toutes les données de configuration, statistiques et opérationnelles.

Stockage de la base de données

Toutes les informations de configuration ainsi que les données collectées par Zabbix sont stockées dans une base de données.

Interface web

Pour un accès facile à Zabbix depuis n'importe où et depuis n'importe quelle plateforme, une interface web est fournie. L'interface fait partie du serveur Zabbix et s'exécute généralement (mais pas nécessairement) sur la même machine physique que celle sur laquelle s'exécute le serveur.

Proxy

Zabbix proxy peut collecter des données de performance et de disponibilité au nom du serveur Zabbix. Un proxy est une composante facultative du déploiement Zabbix ; toutefois, il peut être très utile pour répartir la charge d'un seul serveur Zabbix.

Agent

Les agents Zabbix sont déployés sur les cibles de supervision afin de surveiller activement les ressources et applications locales et de transmettre les données collectées au serveur Zabbix. Depuis Zabbix 4.4, deux types d'agents sont disponibles: le Zabbix agent (léger, pris en charge sur de nombreuses plateformes, écrit en C) et le Zabbix agent 2 (très flexible, facilement extensible avec des plugins, écrit en Go).

Flux de données De plus, il est important de prendre du recul et d'examiner le flux de données global au sein de Zabbix. Afin de créer un élément qui collecte des données, vous devez d'abord créer un hôte. À l'autre extrémité du spectre Zabbix, vous devez d'abord disposer d'un élément pour créer un déclencheur. Vous devez avoir un déclencheur pour créer une action. Ainsi, si vous souhaitez recevoir une alerte indiquant que la charge CPU de *Server X* est trop élevée, vous devez d'abord créer une entrée d'hôte pour *Server X*, puis un élément pour surveiller son CPU, ensuite un déclencheur qui s'active si le CPU est trop élevé, puis une action qui vous envoie un e-mail. Même si cela peut sembler faire beaucoup d'étapes, avec l'utilisation des modèles, ce n'est vraiment pas le cas. Cependant, grâce à cette conception, il est possible de créer une configuration très flexible.

Fonctionnalités Zabbix est une solution de supervision réseau hautement intégrée, offrant une multiplicité de fonctionnalités dans un seul package.

Collecte de données

- contrôles de disponibilité et de performance
- prise en charge de SNMP (à la fois trapping et polling), IPMI, JMX, supervision VMware
- contrôles personnalisés
- collecte des données souhaitées à des intervalles personnalisés
- effectuée par le serveur/proxy et par les agents

Définitions de seuils flexibles

- vous pouvez définir des seuils de problème très flexibles, appelés déclencheurs, en vous référant à des valeurs provenant de la base de données backend

Alerting hautement configurable

- l'envoi des notifications peut être personnalisé selon le calendrier d'escalade, le destinataire, le type de média
- les notifications peuvent être rendues pertinentes et utiles à l'aide de variables macro
- les actions automatiques incluent des commandes à distance

Génération de graphiques en temps réel

- les éléments supervisés sont immédiatement représentés sous forme de graphiques grâce à la fonctionnalité de graphes intégrée

Fonctionnalités de supervision web

- Zabbix peut suivre un parcours de clics de souris simulés sur un site web et vérifier le fonctionnement ainsi que le temps de réponse

Vastes options de visualisation

- possibilité de créer des graphiques personnalisés pouvant combiner plusieurs éléments dans une seule vue
- cartes réseau
- diaporamas dans une vue d'ensemble de type tableau de bord
- rapports
- vue de haut niveau (métier) des ressources supervisées

Stockage des données historiques

- données stockées dans une base de données
- historique configurable
- procédure de maintenance intégrée

Configuration facile

- ajouter des périphériques supervisés en tant qu'hôtes
- les hôtes sont pris en charge pour la supervision une fois dans la base de données
- appliquer des modèles aux périphériques supervisés

Utilisation des modèles

- regroupement des contrôles dans des modèles
- les modèles peuvent hériter d'autres modèles

Découverte réseau

- découverte automatique des périphériques réseau
- auto-enregistrement de l'agent
- découverte des systèmes de fichiers, des interfaces réseau et des OID SNMP

Interface web rapide

- une interface web basée sur PHP
- accessible depuis n'importe où
- vous pouvez naviguer en cliquant
- journal d'audit

API Zabbix

- l'API Zabbix fournit une interface programmable vers Zabbix pour les manipulations en masse, l'intégration de logiciels tiers et d'autres usages.

Système de permissions

- authentification sécurisée des utilisateurs
- certains utilisateurs peuvent être limités à certaines vues

agent complet et facilement extensible

- déployé sur les cibles de supervision
- peut être déployé à la fois sur Linux et Windows

Démons binaires

- écrits en C, pour des performances élevées et une faible empreinte mémoire
- facilement portables

Prêt pour les environnements complexes

- la supervision à distance est facilitée grâce à l'utilisation d'un proxy Zabbix

Glossaire Dans cette section, vous pouvez découvrir la signification de certains termes couramment utilisés dans Zabbix.

hôte

- tout périphérique physique ou virtuel, toute application, tout service ou tout autre ensemble logiquement associé de paramètres surveillés.

groupe d'hôtes

- un regroupement logique d'hôtes. Les groupes d'hôtes sont utilisés lors de l'attribution des droits d'accès aux hôtes pour différents groupes d'utilisateurs.

élément

- une donnée particulière que vous souhaitez recevoir d'un hôte, une métrique de données.

prétraitement des valeurs

- une transformation de la valeur de métrique reçue avant son enregistrement dans la base de données.

déclencheur

- une expression logique qui définit un seuil de problème et qui est utilisée pour « évaluer » les données reçues dans les éléments.

Lorsque les données reçues dépassent le seuil, les déclencheurs passent de l'état *OK* à l'état *Problème*. Lorsque les données reçues sont inférieures au seuil, les déclencheurs restent ou repassent à l'état *OK*.

modèle

- un ensemble d'entités (éléments, déclencheurs, graphiques, règles de découverte de bas niveau, scénarios web) prêtes à être appliquées à un ou plusieurs hôtes.

Les modèles ont pour objectif d'accélérer le déploiement des tâches de supervision sur un hôte et de faciliter l'application de modifications en masse aux tâches de supervision. Les modèles sont directement liés aux hôtes individuels.

groupe de modèles

- un regroupement logique de modèles. Les groupes de modèles sont utilisés lors de l'attribution des droits d'accès aux modèles pour différents groupes d'utilisateurs.

événement

- une occurrence unique d'un événement nécessitant une attention particulière, comme un changement d'état d'un déclencheur ou l'exécution d'une découverte ou de l'auto-enregistrement d'un agent.

étiquette d'événement

- un marqueur prédéfini pour l'événement. Elle peut être utilisée pour la corrélation d'événements, la granularité des permissions, etc.

corrélation d'événements

- une méthode permettant de corréler les problèmes avec leur résolution de manière flexible et précise.

Par exemple, vous pouvez définir qu'un problème signalé par un déclencheur peut être résolu par un autre déclencheur, qui peut même utiliser une méthode différente de collecte des données.

problème

- un déclencheur qui se trouve dans l'état « Problème ».

mise à jour d'un problème

- les options de gestion des problèmes fournies par Zabbix, telles que l'ajout d'un commentaire, la confirmation, la modification de la gravité ou la fermeture manuelle.

action

- un moyen prédéfini de réagir à un événement.

Une action se compose d'opérations (par exemple, l'envoi d'une notification) et de conditions (*quand* l'opération est exécutée).

escalade

- un scénario personnalisé d'exécution d'opérations dans le cadre d'une action ; une séquence d'envoi de notifications ou d'exécution de commandes à distance.

macro

- un espace réservé variable qui est résolu en une valeur spécifique selon le contexte.

média

- un moyen de transmettre des notifications ; un canal de transmission.

notification

- un message concernant un événement envoyé à un utilisateur via le canal média choisi.

commande à distance

- une commande prédéfinie qui est exécutée automatiquement sur un hôte surveillé lorsqu'une condition donnée est remplie.

scénario web

- une ou plusieurs requêtes HTTP permettant de vérifier la disponibilité d'un site web.

interface

- l'interface web fournie avec Zabbix.

tableau de bord

- une section personnalisable de l'interface web affichant des synthèses et des visualisations d'informations importantes dans des unités visuelles appelées widgets.

widget

- une unité visuelle affichant des informations d'un certain type et provenant d'une certaine source (une synthèse, une carte, un graphique, l'horloge, etc.), utilisée dans le tableau de bord.

API Zabbix

- l'API Zabbix vous permet d'utiliser le protocole JSON RPC pour créer, mettre à jour et récupérer des objets Zabbix (tels que des hôtes, des éléments, des graphiques et autres), ou d'effectuer toute autre tâche personnalisée.

serveur Zabbix

- un processus central du logiciel Zabbix qui effectue la supervision, interagit avec les proxy et les agents Zabbix, calcule les déclencheurs, envoie des notifications et constitue un référentiel central des données.

proxy Zabbix

- un processus qui peut collecter des données pour le compte du serveur Zabbix, réduisant ainsi la charge de traitement du serveur.

agent Zabbix

- un processus déployé sur les cibles de supervision afin de surveiller activement les ressources et les applications locales.

agent Zabbix 2

- une nouvelle génération d'agent Zabbix destinée à surveiller activement les ressources et les applications locales, permettant l'utilisation de plugins personnalisés pour la supervision.

> **Attention:**
> Comme l'agent Zabbix 2 partage de nombreuses fonctionnalités avec l'agent Zabbix, le terme « agent Zabbix » utilisé dans la documentation désigne à la fois l'agent Zabbix et l'agent Zabbix 2 lorsque leur comportement fonctionnel est identique. L'agent Zabbix 2 est nommé spécifiquement uniquement lorsque ses fonctionnalités diffèrent.

chiffrement

- la prise en charge des communications chiffrées entre les composants Zabbix (serveur, proxy, agent, utilitaires zabbix_sender et zabbix_get) à l'aide du protocole Transport Layer Security (TLS).

auto-enregistrement de l'agent

- un processus automatisé par lequel un agent Zabbix s'enregistre lui-même en tant qu'hôte et commence la supervision.

découverte réseau

- la découverte automatisée des périphériques réseau.

découverte de bas niveau

- la découverte automatisée d'entités de bas niveau sur un périphérique donné (par exemple, des systèmes de fichiers, des interfaces réseau, etc.).

règle de découverte de bas niveau

- un ensemble de définitions pour la découverte automatisée d'entités de bas niveau sur un périphérique.

prototype d'élément

- *une métrique dont certains paramètres sont définis comme variables, prête pour la découverte de bas niveau.* Après la découverte de bas niveau, les variables sont automatiquement remplacées par les paramètres réels découverts et la métrique commence automatiquement à collecter des données.

prototype de déclencheur

- *un déclencheur dont certains paramètres sont définis comme variables, prêt pour la découverte de bas niveau.* Après la découverte de bas niveau, les variables sont automatiquement remplacées par les paramètres réels découverts et le déclencheur commence automatiquement à évaluer les données.

Des *prototypes* d'autres entités Zabbix sont également utilisés dans la découverte de bas niveau : prototypes de graphiques, prototypes d'hôtes et prototypes de groupes d'hôtes.

Zabbix Cloud

Zabbix Cloud

Découvrez les options de configuration disponibles et les contrôles d'accès pour les organisations et les nœuds dans Zabbix Cloud.

Veuillez noter que les nœuds Zabbix Cloud exécutent actuellement **Zabbix 7.0**.
 Pour des informations détaillées sur Zabbix 7.0, consultez la documentation officielle.

Configurer et gérer Zabbix Cloud

Commencez à utiliser Zabbix Cloud Des instructions guidées pour mettre rapidement en service votre instance Zabbix Cloud, y compris la configuration initiale et la connexion de vos appareils surveillés.

Configuration du nœud Découvrez comment configurer votre nœud Zabbix Cloud, en vous concentrant sur l'accès, la gestion des données et les paramètres des ressources.

Ajout d'utilisateurs Découvrez comment ajouter des utilisateurs et leur attribuer des rôles au sein d'une organisation Zabbix Cloud et de ses nœuds.

Découvrez Zabbix Cloud

Zabbix Cloud vs. sur site Comparez les différences en matière de gestion, d'évolutivité et de maintenance entre les deux déploiements, et découvrez lequel peut prendre en charge votre flux de travail de la manière la plus fluide.

Découvrez Zabbix Cloud Obtenez un aperçu de Zabbix Cloud, y compris ses principales fonctionnalités, ses avantages et ce qui est inclus dans un abonnement. Pour les questions fréquentes sur la configuration et l'utilisation, consultez la FAQ.

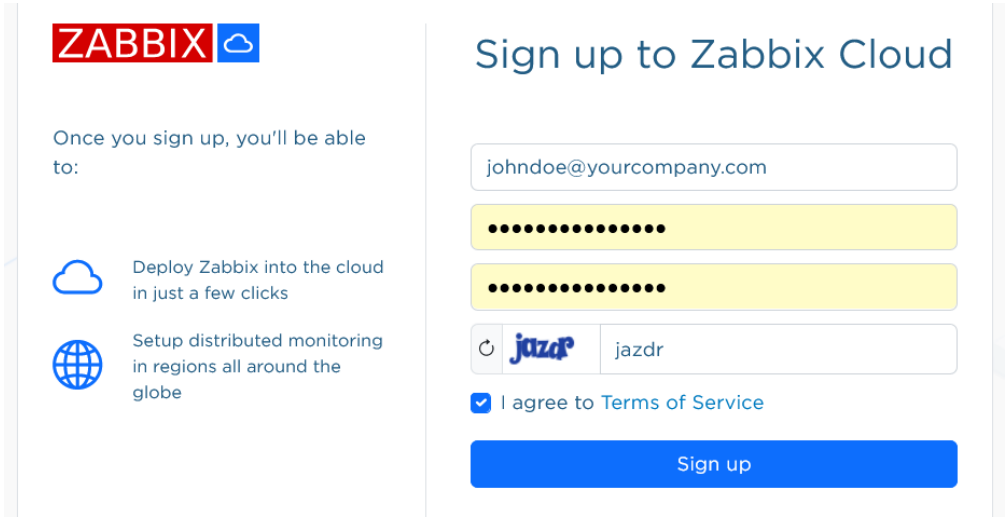
Déployer Zabbix dans le cloud

Pour démarrer Zabbix dans le cloud, suivez ce processus :

1. Connectez-vous ou inscrivez-vous sur cloud.zabbix.com.
2. Créez un nœud cloud.
3. Commencez à utiliser Zabbix dans le cloud.

Se connecter ou s'inscrire Vous pouvez vous connecter sur cloud.zabbix.com avec un compte Google, GitHub ou Microsoft existant.

Vous pouvez également accéder à la page d'inscription pour vous enregistrer.



Après votre inscription, vous recevrez un e-mail confirmant la création de votre nouveau compte.

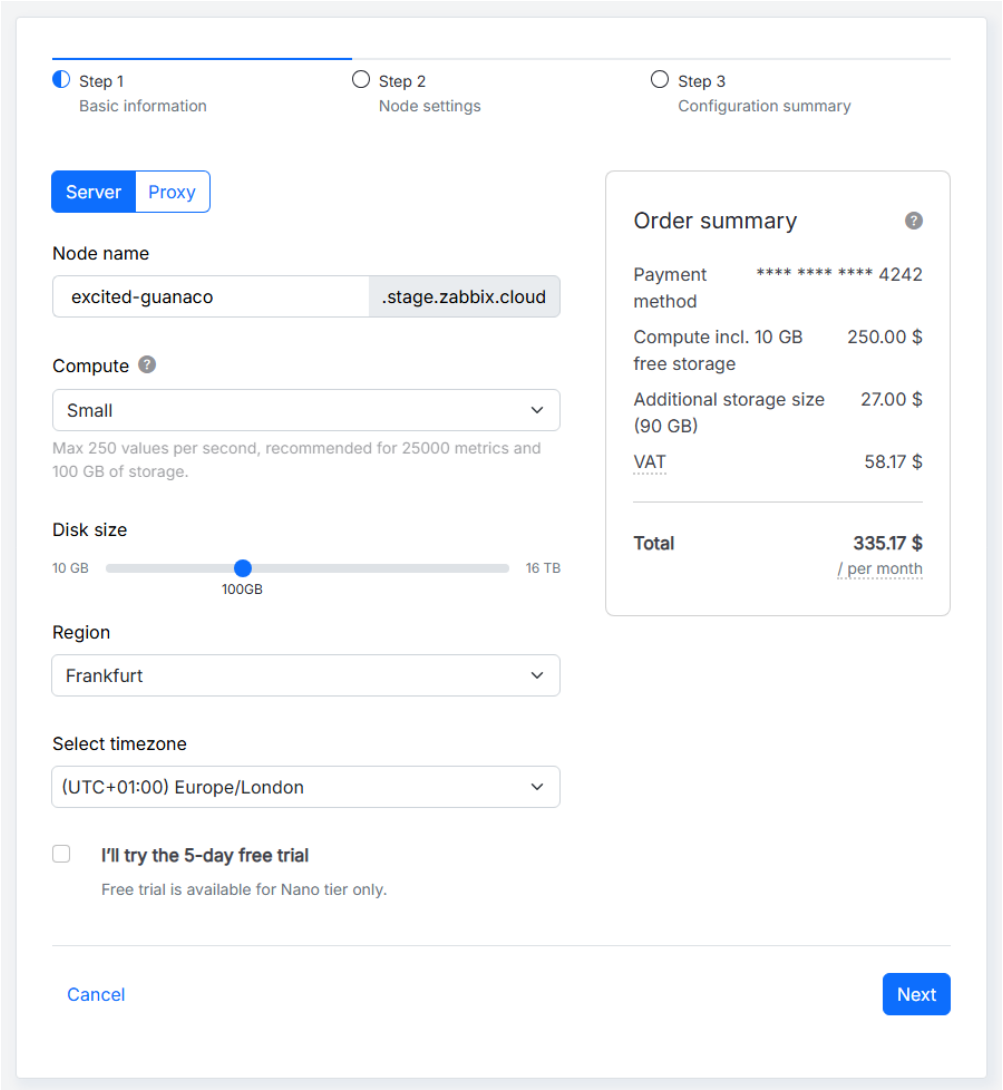
Accédez à cloud.zabbix.com pour vous connecter. Un mot de passe à usage unique vous sera demandé lors de la connexion. Ce mot de passe vous sera envoyé par e-mail.

Créer un nœud cloud Un nœud cloud est une instance Zabbix déployée dans Zabbix Cloud. Il peut être déployé en tant que nœud serveur ou nœud proxy.

Une fois connecté, vous constaterez qu'une organisation *Private* existe déjà par défaut pour votre compte. Vous pourrez renommer l'organisation par défaut ultérieurement.

Cliquez sur *Create New Node*. Un formulaire de configuration du nœud en trois étapes s'ouvre.

Lors de la première étape, vous sélectionnez les informations de base.



Parameter	Description
Server/Proxy	Sélectionnez si vous souhaitez déployer une instance de serveur Zabbix ou une instance de proxy Zabbix.
Node name	Le nom du nœud forme un sous-domaine unique au format `<your-name>.zabbix.cloud`, utilisé pour accéder aux services du nœud.
Compute	Sélectionnez la puissance de calcul du nœud. Pour déterminer le niveau de ressources dont vous avez besoin, consultez les détails des tarifs. Vous pourrez toujours effectuer une mise à niveau ultérieurement si vous avez besoin de davantage de puissance. Une méthode de facturation sera requise avant la création d'un nœud hors période d'essai. Ajoutez les informations de facturation via le menu *Billing*. Un **essai gratuit** de 5 jours est disponible pour tester un nœud de niveau *Nano*. Aucun frais ne vous sera facturé pendant la période d'essai, mais à son terme, votre nœud sera supprimé s'il n'est pas converti en nœud payant.
Disk size	Le curseur est défini par défaut sur un minimum recommandé en fonction du niveau. Augmentez cette valeur si vous stockez de gros volumes d'historique ou de tendances. Notez que les disques plus volumineux entraînent des coûts mensuels plus élevés et que vous ne pouvez pas réduire vous-même la taille d'un disque : vous devrez contacter le support pour la diminuer.
Region	Sélectionnez la région dans laquelle le nœud sera déployé.
Select timezone	Sélectionnez le fuseau horaire du serveur Zabbix. Ce fuseau horaire est utilisé par le serveur Zabbix et est indépendant des fuseaux horaires de la maintenance et du profil utilisateur.

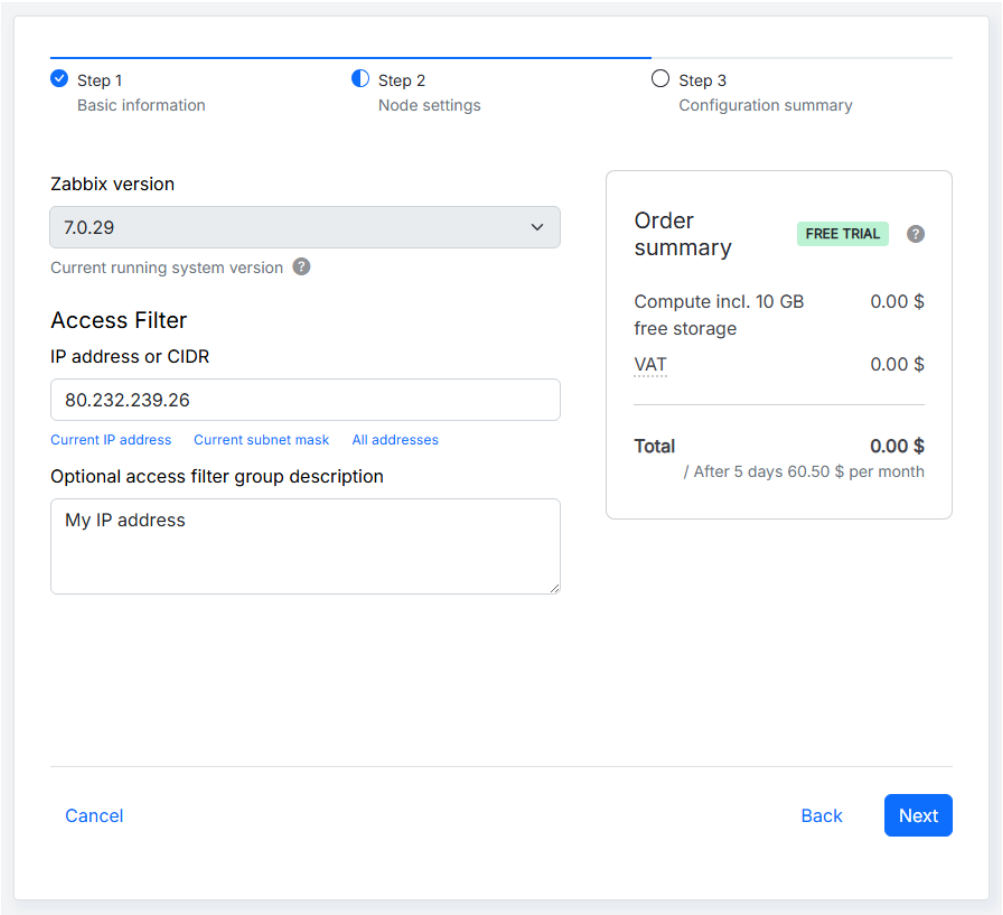
Après avoir terminé la configuration, cliquez sur *Next*.

Lors de la deuxième étape, vous sélectionnez les paramètres du nœud.

Les options de configuration de la deuxième étape dépendent de la création d'un nœud serveur ou proxy.

Configurer un nœud serveur

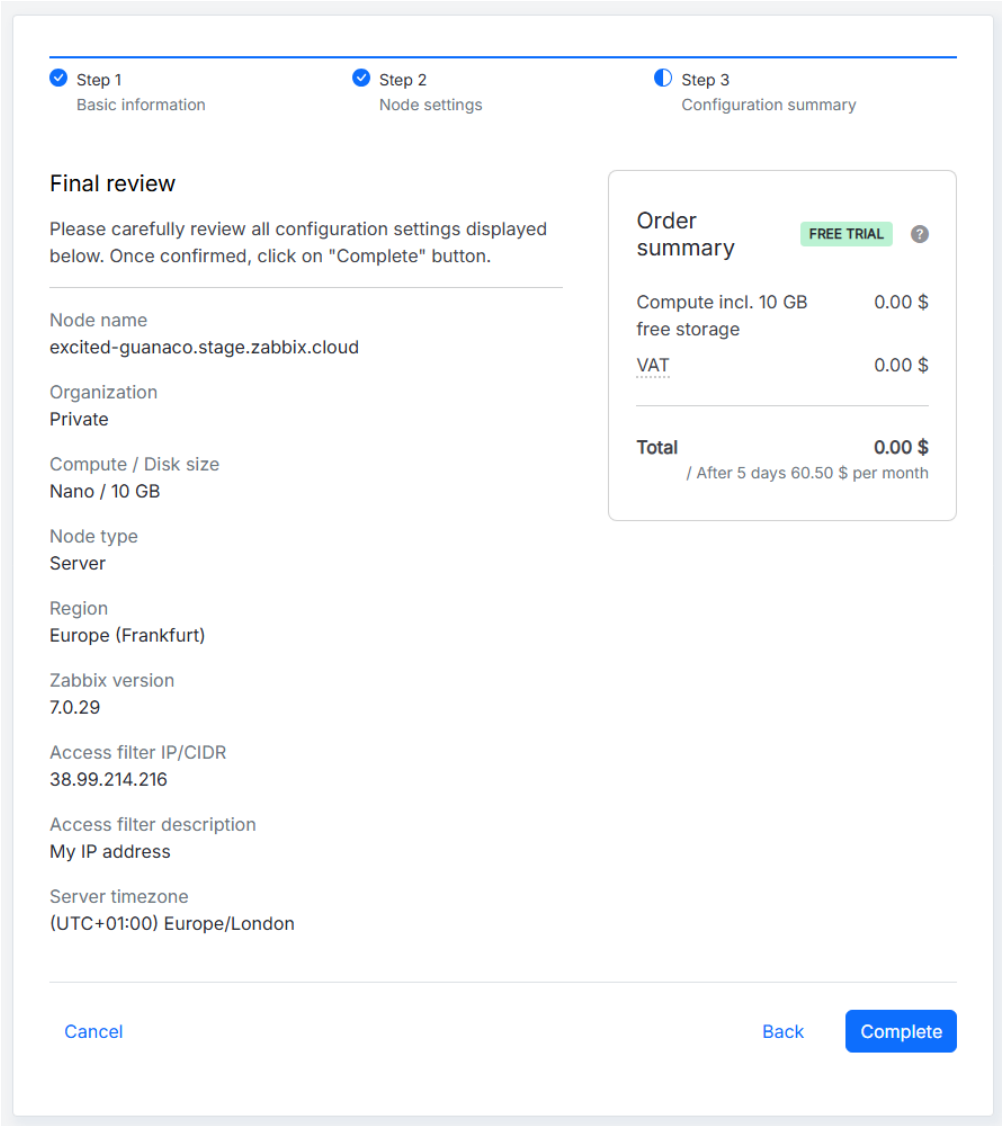
Lors de la création d'un nœud serveur, les paramètres suivants sont disponibles :



Parameter	Description
Zabbix version	Sélectionnez la version de Zabbix. Zabbix Cloud prend en charge la dernière version LTS et applique automatiquement les mises à jour de version mineure.
IP address or CIDR	Spécifiez le filtre d'accès initial du nœud. Saisissez manuellement l'adresse IP ou le CIDR, ou remplissez automatiquement le champ en sélectionnant l'option correspondante. Pour les nœuds serveur, ce filtre s'applique aux connexions au serveur, à l'API et à l'interface. Le filtre peut être modifié ultérieurement dans l'onglet Access filters.
Optional access filter group description	Ajoutez une description facultative pour le groupe de filtres d'accès, qui peut contenir plusieurs adresses IP ou CIDR.

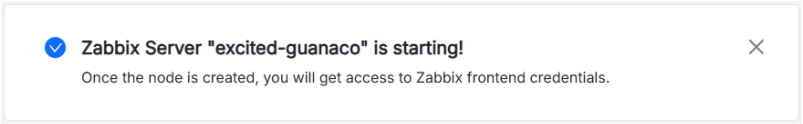
Warning:
Lors de la création du nœud, le champ *IP address or CIDR* est prérempli avec l'adresse IP utilisée pour accéder à Zabbix Cloud. Vous pouvez modifier cette valeur afin de spécifier un autre filtre d'accès initial. Une fois le nœud créé, vous pouvez ajouter, modifier ou supprimer des filtres d'accès depuis l'onglet *Access filters*.

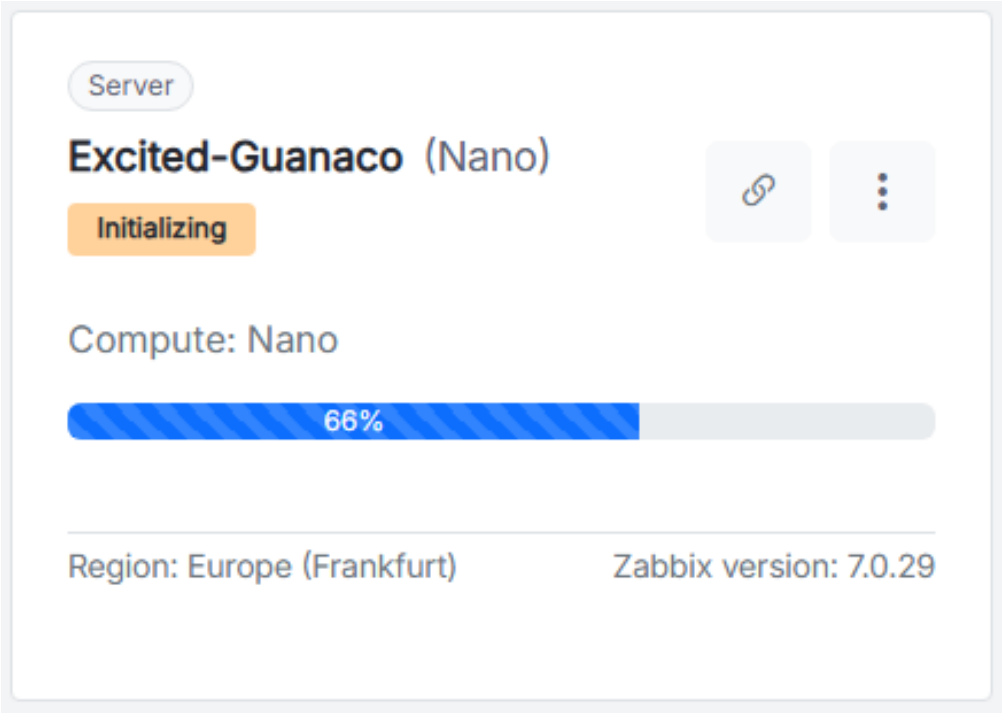
La troisième étape affiche un récapitulatif de la configuration du nœud.



Après avoir vérifié les paramètres de configuration et confirmé que tout est correct, cliquez sur *Complete* pour un nœud d'essai ou sur *Proceed to payment* pour un nœud payant. Sinon, revenez à une étape précédente en cliquant sur le bouton *Back*, ou cliquez sur *Cancel* pour recommencer depuis le début.

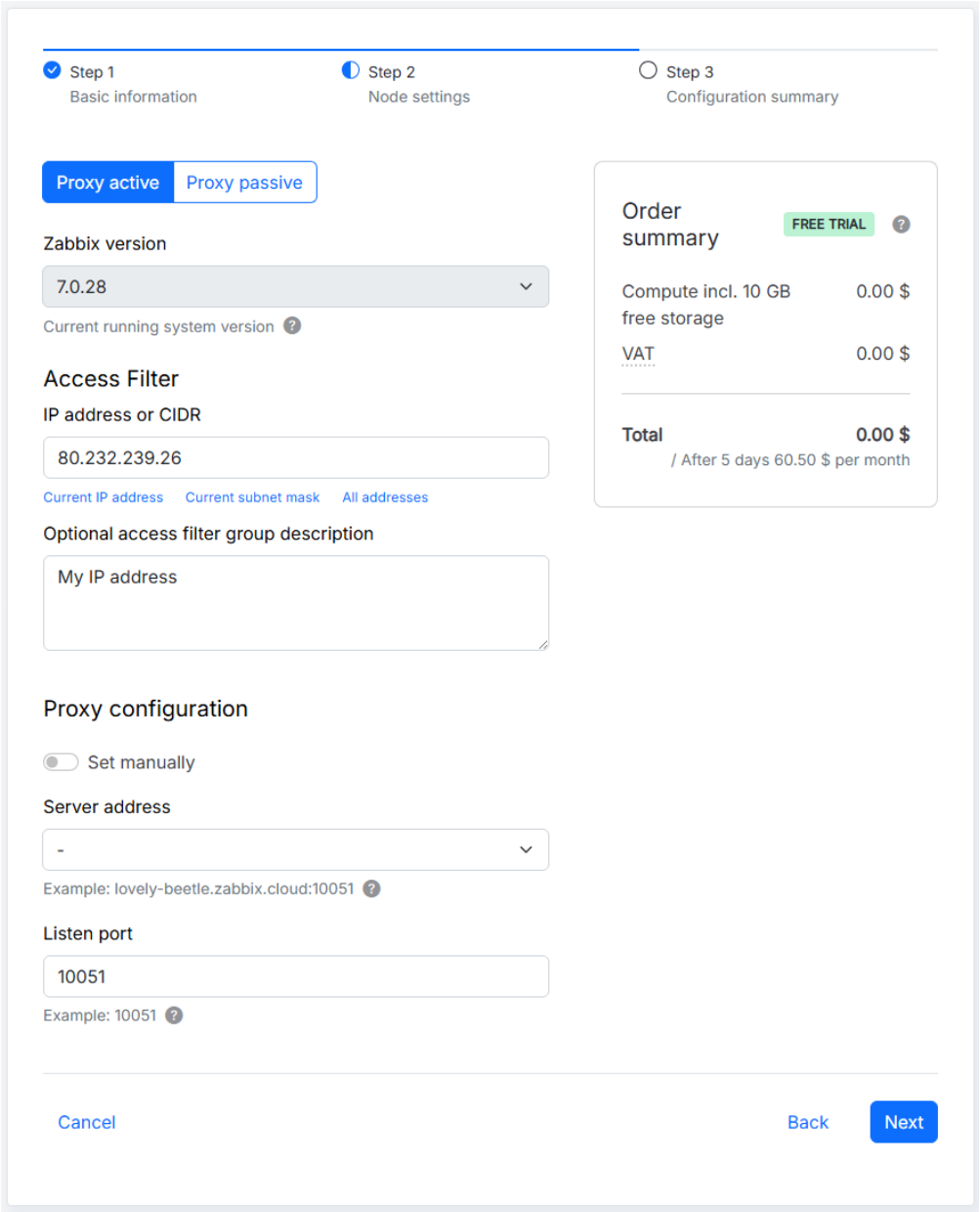
Pendant l'initialisation du nœud, un message correspondant s'affiche et le nœud apparaît déjà sous *Organizations*.





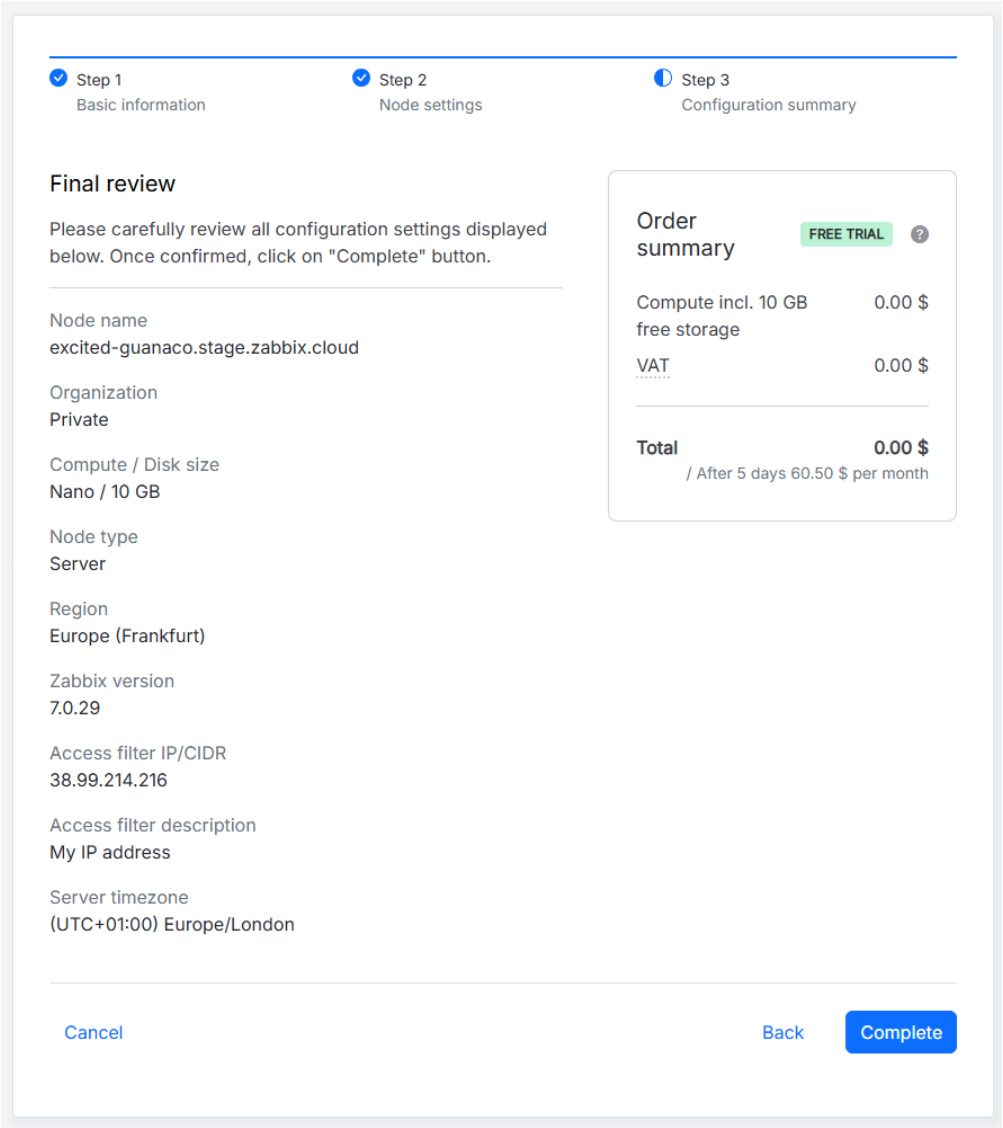
Configurer un nœud proxy

Lors de la création du nœud proxy, les paramètres suivants sont disponibles :



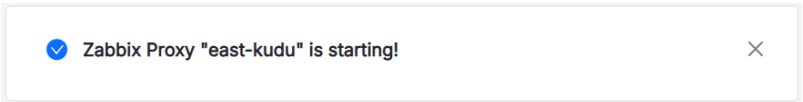
Parameter	Description
Proxy active/Proxy passive	Sélectionnez le mode du proxy. En mode actif, le proxy initie les connexions au serveur Zabbix ; en mode passif, le serveur Zabbix initie les connexions au proxy.
Zabbix version	Sélectionnez la version de Zabbix. Zabbix Cloud prend en charge la dernière version LTS et applique automatiquement les mises à jour de version mineure.
IP address or CIDR	Spécifiez le filtre d'accès IP entrant initial du proxy. Saisissez manuellement l'adresse IP ou le CIDR, ou remplissez automatiquement le champ en sélectionnant l'option correspondante. Le filtre définit les requêtes réseau autorisées à atteindre le proxy et peut être modifié ultérieurement dans l'onglet Access filters.
Optional access filter group description	Ajoutez une description facultative pour le groupe de filtres d'accès, qui peut contenir plusieurs adresses IP ou CIDR.
Set manually	Sélectionnez la manière de spécifier le serveur Zabbix. Lorsque cette option est désactivée, sélectionnez un nœud serveur dans la liste des nœuds serveur auxquels vous avez accès. Lorsqu'elle est activée, spécifiez manuellement les adresses des serveurs.
Server address	Spécifiez l'adresse du serveur Zabbix. Lorsque *Set manually* est désactivé, sélectionnez un nœud serveur dans la liste. Lorsqu'il est activé, spécifiez jusqu'à 10 adresses de serveur. Pour un proxy actif, spécifiez l'adresse IP ou le nom DNS du serveur Zabbix. Pour un proxy passif, spécifiez l'adresse IP, le nom DNS ou le CIDR du serveur Zabbix. Les filtres d'accès sont configurés séparément et ne sont pas mis à jour automatiquement lorsqu'une adresse de serveur est spécifiée.
Listen port	Spécifiez le port sur lequel le proxy écoute les connexions entrantes. Le port par défaut est 10051. Le port d'écoute ne peut pas être modifié après la création du nœud.

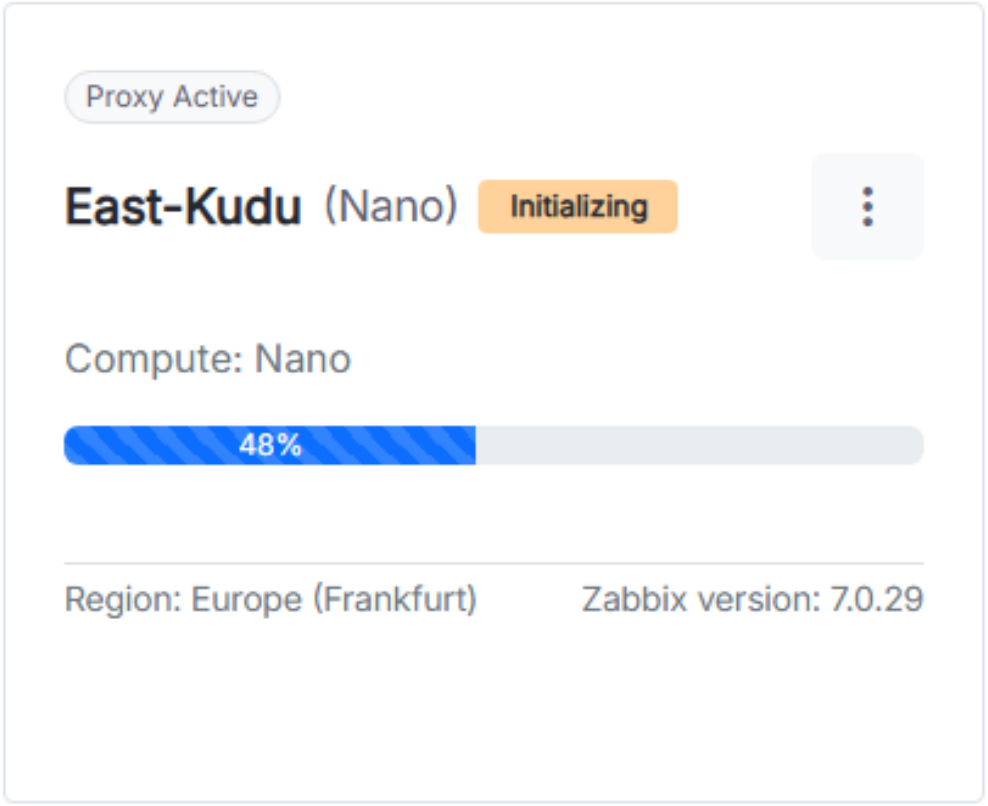
Lors de la troisième étape, vous verrez le récapitulatif de la configuration du nœud.



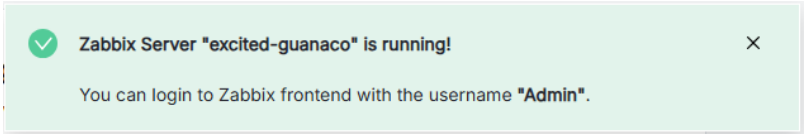
Après avoir vérifié les paramètres de configuration et confirmé que tout est correct, cliquez sur *Complete* pour un nœud d'essai ou sur *Proceed to payment* pour un nœud payant. Sinon, revenez aux étapes précédentes en cliquant sur le bouton *Back*, ou cliquez sur *Cancel* pour recommencer depuis le début.

Pendant l'initialisation du nœud, un message correspondant s'affiche et le nœud apparaît déjà sous *Organizations*.





Pour configurer un proxy Zabbix Cloud, consultez Configurer un proxy Zabbix Cloud.

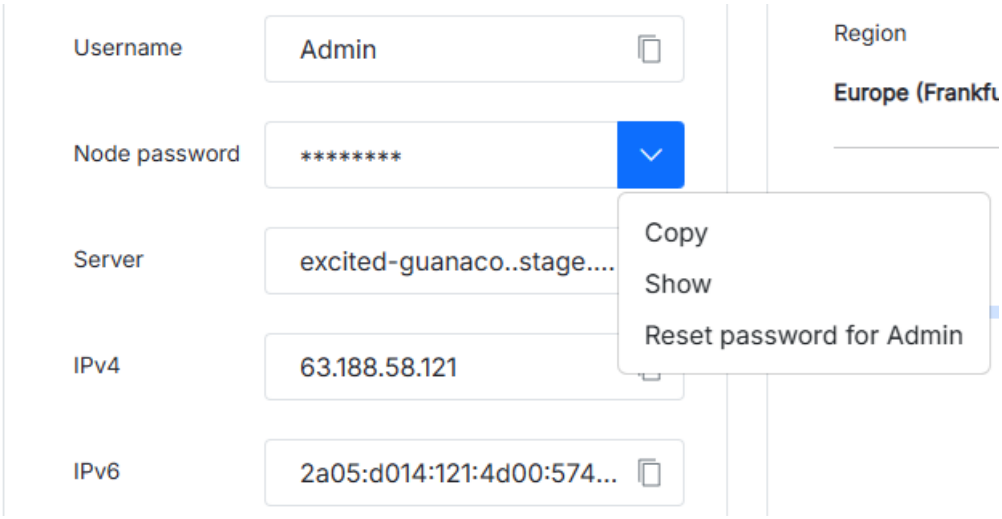


Commencer à utiliser Zabbix

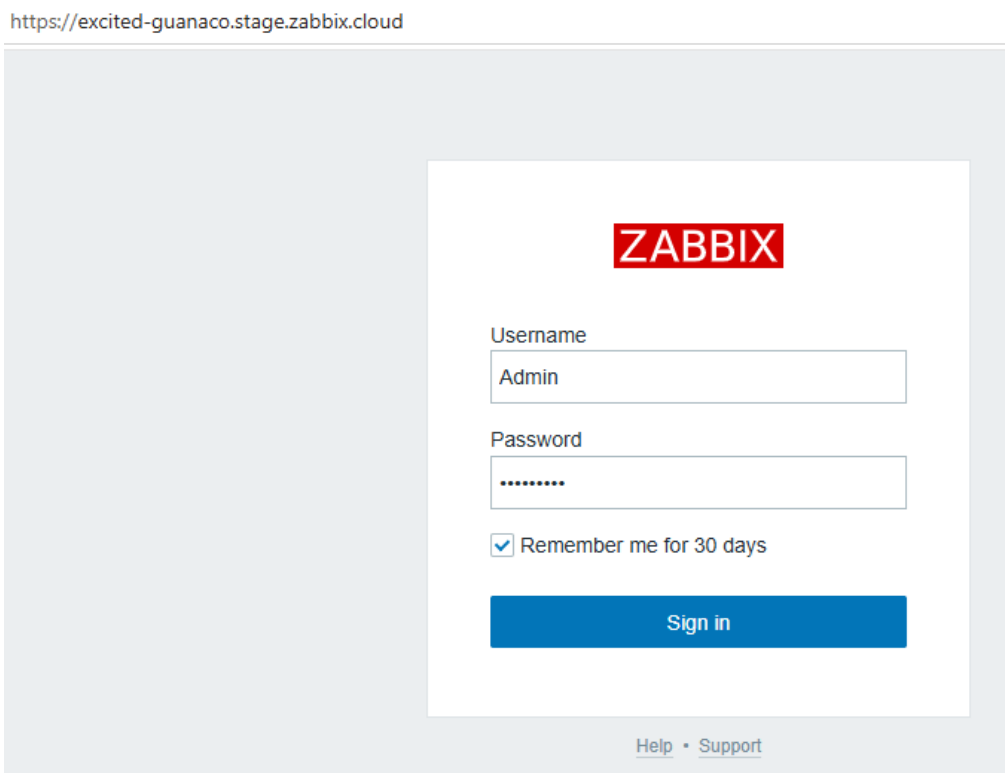
Lorsque l'état du nœud du serveur passe de *Initializing* à *Running*, vous pouvez vous connecter à l'interface Zabbix.

Pour accéder au mot de passe initial de l'interface, cliquez sur le nœud créé.

Vous serez redirigé vers les paramètres du nœud (vue d'ensemble, filtres d'accès, chiffrement, sauvegardes, etc.). Dans **Connection info**, repérez le champ *Node password* et sélectionnez *Reset password for Admin* dans la liste déroulante. Une fois le mot de passe réinitialisé, cliquez sur la même liste déroulante et sélectionnez *Copy*.



Cliquez sur l'URL de l'interface pour ouvrir l'interface Zabbix (`excited-guanaco.zabbix.cloud`).



Connectez-vous avec les identifiants suivants :

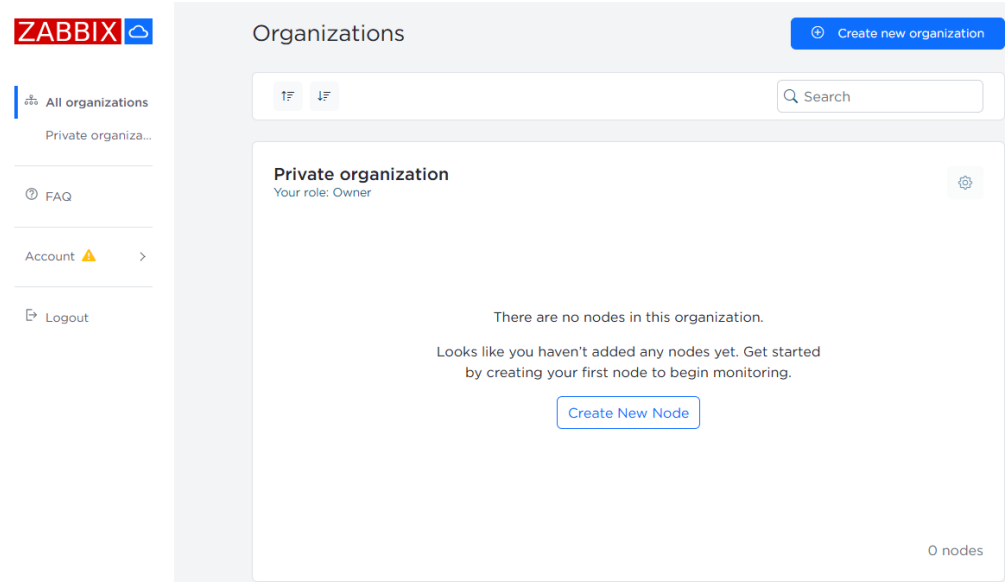
- *Username* - `Admin`
- *Password* - indiquez le mot de passe copié

Pour commencer la supervision, vous devez définir des filtres d'accès - une ou plusieurs adresses IP autorisées à se connecter au serveur Zabbix Cloud. Au moins une adresse doit être définie.

Gestion des organisations Une organisation est un conteneur pour un ou plusieurs nœuds. Les organisations vous aident également à structurer vos ressources et à déléguer des responsabilités, telles que :

- Gestion des nœuds
- Facturation

Lorsque vous vous connectez à Zabbix Cloud, la page *Toutes les organisations* s'affiche. Par défaut, une organisation privée est créée pour votre compte.

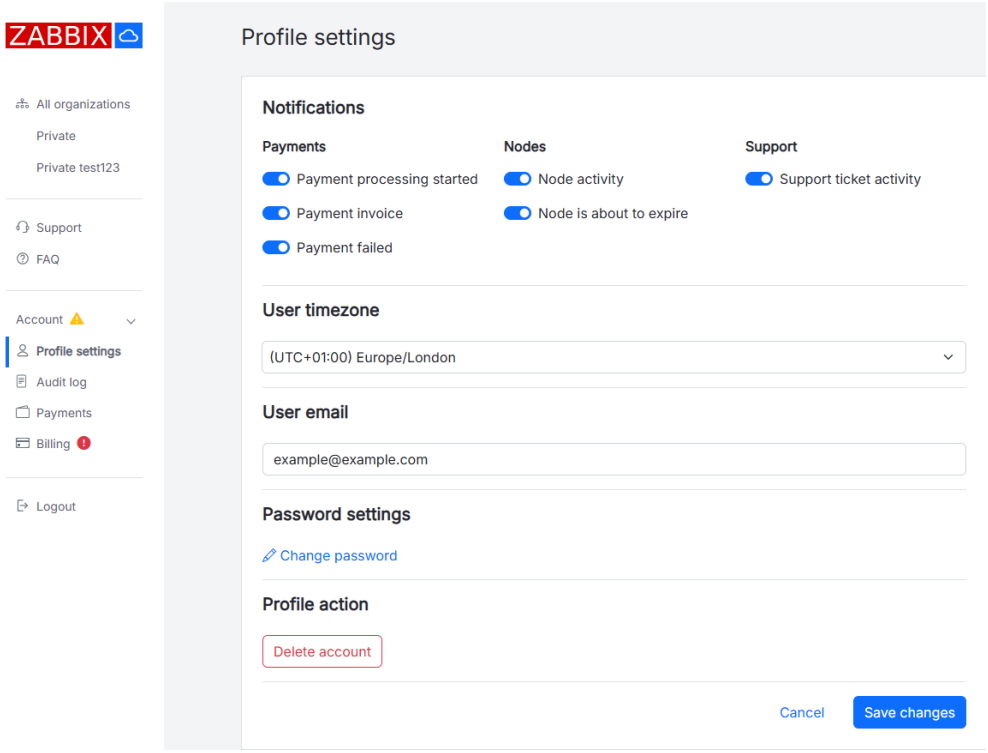


En tant que propriétaire de cette organisation, vous pouvez gérer les utilisateurs, les administrateurs, les nœuds et le nom de l'organisation. Cliquez sur le bouton de paramètres  pour accéder aux options de gestion.

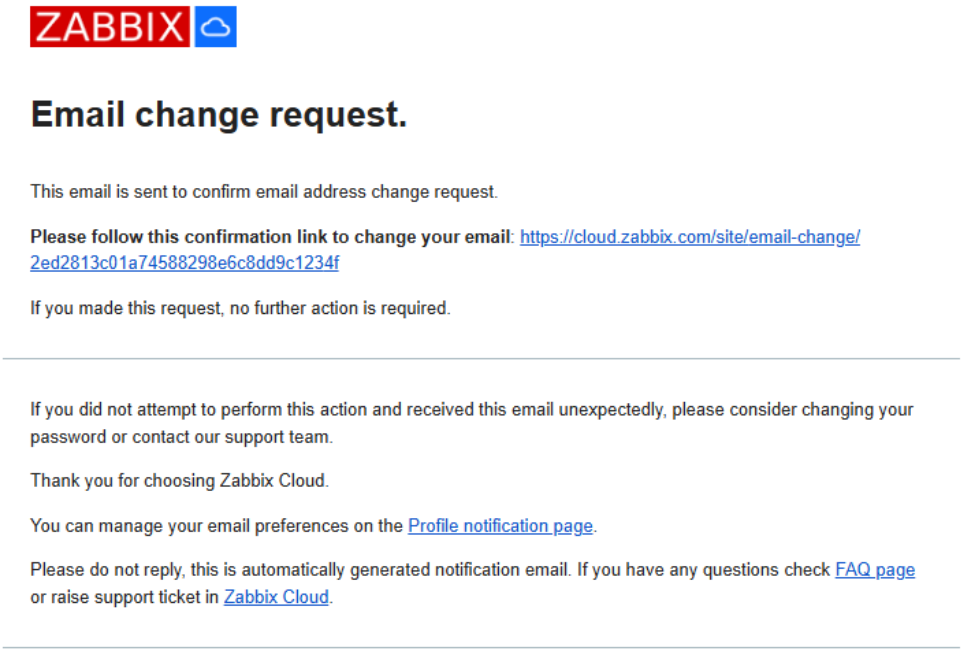
Gestion des comptes Modification de l'adresse e-mail du compte

Pour modifier l'adresse e-mail du compte, accédez à *Compte* > *Paramètres du profil*.

Dans le champ *E-mail de l'utilisateur*, saisissez la nouvelle adresse e-mail, puis cliquez sur *Enregistrer les modifications*.



Un message de confirmation est envoyé à la nouvelle adresse e-mail. Suivez le lien contenu dans le message pour confirmer la modification de l'adresse e-mail.



Le lien de confirmation ouvre la page *Paramètres du profil*, qui affiche un message de confirmation ainsi que la nouvelle adresse e-mail dans le champ E-mail de l'utilisateur.

Une fois la modification confirmée, un e-mail confirmant la modification est envoyé aux deux adresses e-mail, la nouvelle et l'ancienne. Le message envoyé à l'ancienne adresse inclut les anciennes et nouvelles adresses e-mail.

Configuration du nœud

Présentation

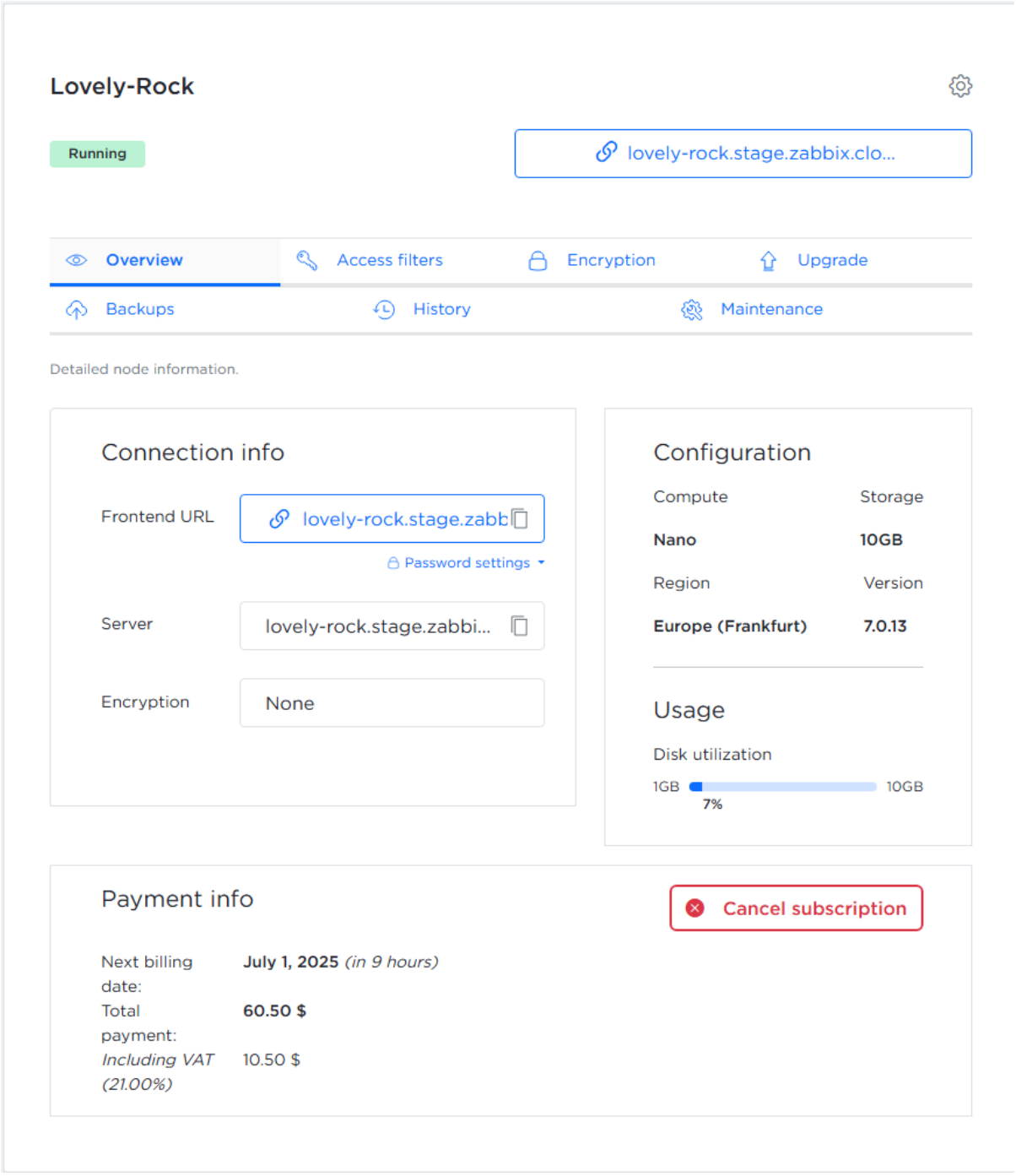
Un **nœud cloud** est un composant Zabbix déployé dans le cloud. Les nœuds serveur incluent le serveur Zabbix, la base de données et l'interface, tandis que les nœuds proxy incluent le proxy Zabbix et la base de données. Un nœud cloud dispose de ressources informatiques dédiées (processeur, mémoire) et d'un espace de stockage sur disque.

Les utilisateurs peuvent créer un ou plusieurs nœuds cloud. Les nœuds cloud peuvent être regroupés en **organisations** (structurelles, régionales, etc.), créant ainsi une configuration de supervision distribuée.

Les nœuds cloud et les organisations sont créés à l'aide de l'interface Zabbix Cloud. Pour accéder à la page de configuration d'un nœud, cliquez sur le nœud dans *Toutes les organisations*, dans votre menu d'organisation, ou sur le lien *Page de configuration du nœud* affiché après le lancement du nœud.

Dans la vue de configuration du nœud, chaque onglet vous permet de gérer un aspect spécifique de votre nœud :

- Présentation - afficher les détails du nœud, ouvrir son interface et gérer les utilisateurs du nœud.
- Filtres d'accès - contrôler l'accès par adresse IP ou sous-réseau.
- Chiffrement - configurer TLS en téléversant des certificats, des clés et des CRL facultatives.
- Sauvegardes - créer, restaurer et supprimer des instantanés du nœud ; afficher la prochaine sauvegarde planifiée.
- Historique - définir les périodes de conservation des journaux d'audit, de l'historique et des tendances, avec des unités prédéfinies et une limite de mise à jour de 3 fois par jour.
- Maintenance - définir des fenêtres de maintenance hebdomadaires selon le fuseau horaire, le jour et l'heure.
- Mise à niveau - augmenter les ressources de calcul ou de stockage, ou gérer l'état de votre période d'essai ou de votre paiement.

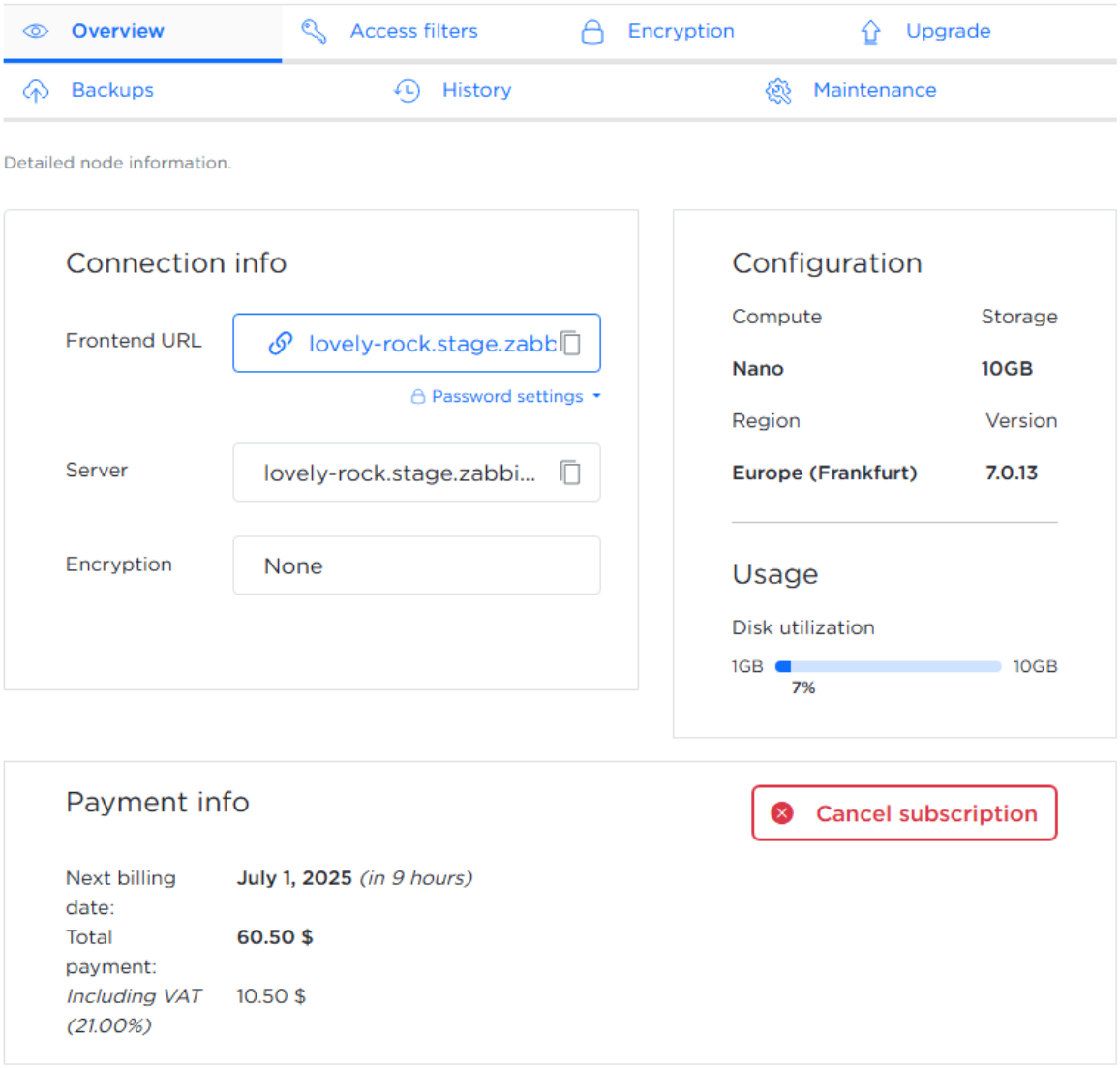


Note:
Certains onglets peuvent être indisponibles selon le type de nœud (serveur ou proxy), le type d'utilisateur qui consulte la page et l'état du nœud (version d'essai ou non).

Onglet Vue d'ensemble

L'onglet *Vue d'ensemble* fournit des informations détaillées sur le nœud. Les informations affichées dépendent du fait que le nœud soit un serveur ou un proxy.

Nœud serveur L'onglet *Overview* du nœud serveur contient les sections suivantes :



Informations de connexion :

- URL de l'interface cliquable avec une option de copie dans le presse-papiers
- Nom d'utilisateur
- Mot de passe du nœud et ses paramètres
- Nom d'hôte du serveur avec une option de copie dans le presse-papiers
- Adresse IPv4 avec une option de copie dans le presse-papiers
- Adresse IPv6 avec une option de copie dans le presse-papiers
- Méthode de chiffrement actuelle

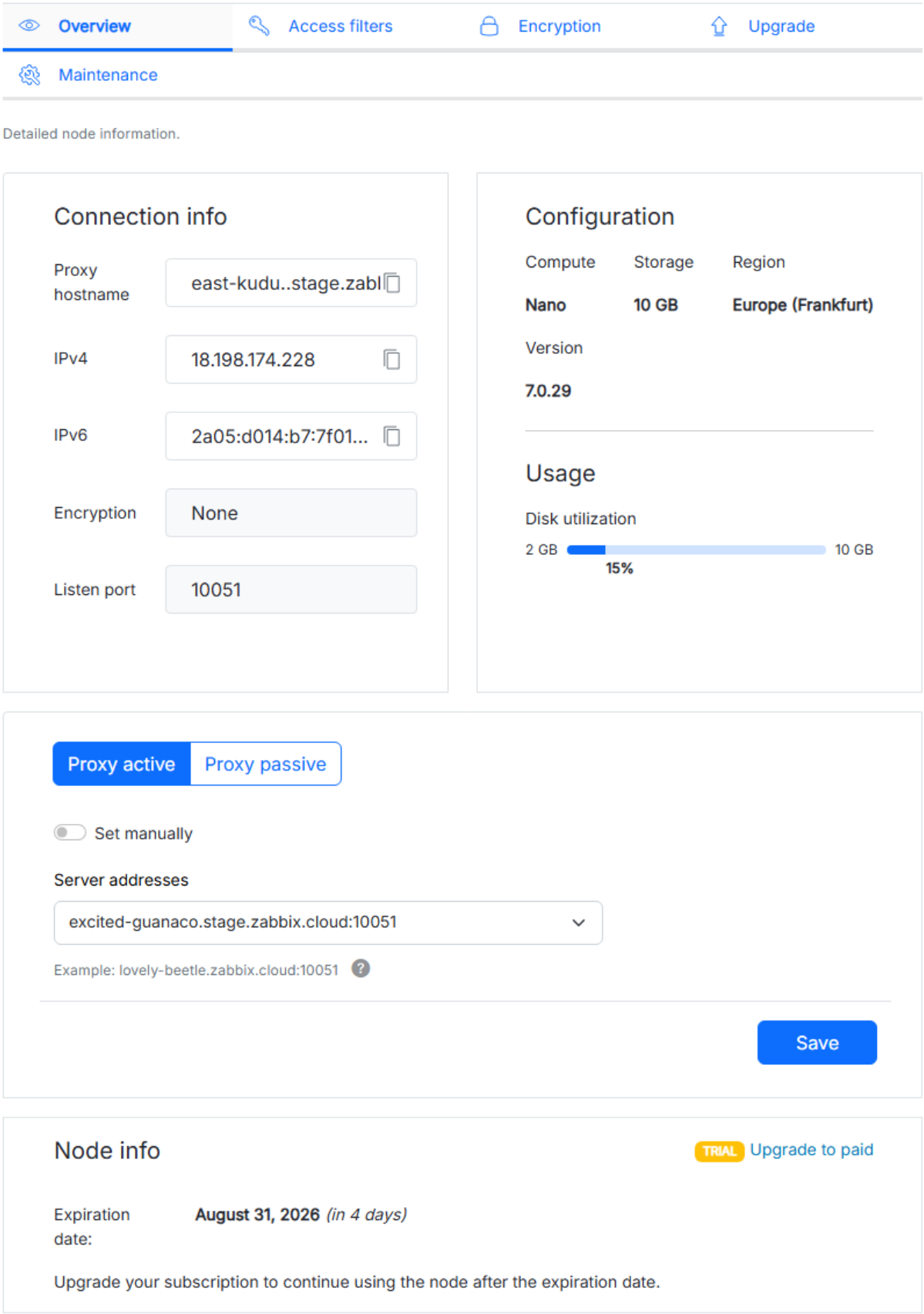
Configuration et utilisation :

- Niveau de calcul, stockage maximal, région
- Version des composants Zabbix
- Utilisation du disque

Informations sur le nœud/Informations de paiement :

- Date d'expiration (pour les nœuds d'essai et prépayés)
- Un bouton contextuel est affiché selon le type de nœud une fois que le nœud a terminé son initialisation : - *Passer à une offre payante* - disponible pour les nœuds en période d'essai gratuite - *Annuler l'abonnement* - disponible pour les nœuds payants avec un abonnement actif - *Se réabonner à l'offre payante* - affiché si un abonnement a été annulé - Aucun bouton d'action n'est affiché pour les nœuds prépayés, qui ont une longue période de validité et ne peuvent pas être annulés

Nœud proxy L'onglet *Overview* du nœud proxy contient les sections suivantes :



Informations de connexion :

- Nom d'hôte du proxy avec une option de copie dans le presse-papiers
- Adresse IPv4 avec une option de copie dans le presse-papiers
- Adresse IPv6 avec une option de copie dans le presse-papiers
- Méthode de chiffrement actuelle
- Port d'écoute actuel

Configuration et utilisation :

- Niveau de calcul, stockage maximal, région
- Version des composants Zabbix
- Utilisation du disque

Mode du proxy :

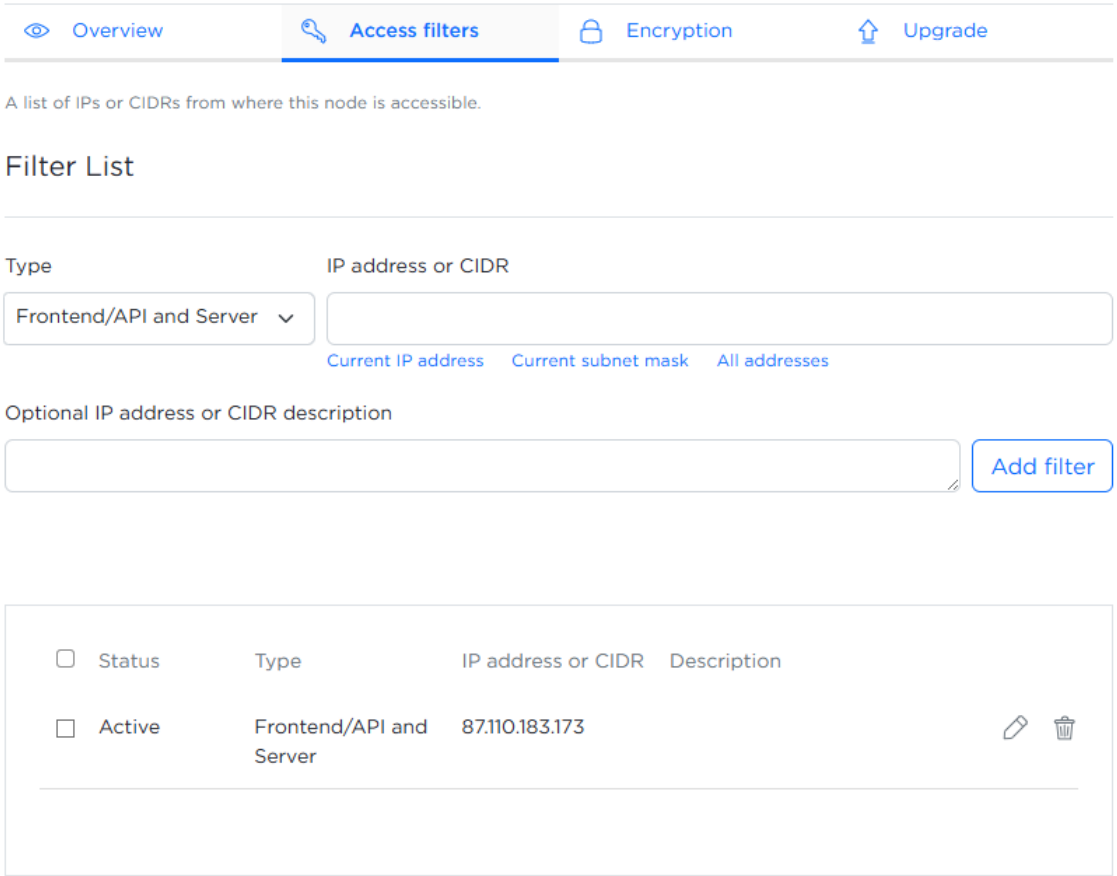
- Basculer entre les modes proxy actif et passif
- Sélectionner l'adresse du serveur dans la liste déroulante *Server addresses*, ou activer *Set manually* pour saisir manuellement l'adresse du serveur

Informations sur le nœud/Informations de paiement :

- Date d'expiration (pour les nœuds d'essai et prépayés)
- Un bouton contextuel s'affiche selon le type de nœud une fois l'initialisation du nœud terminée : - *Upgrade to paid* - disponible pour les nœuds bénéficiant d'un essai gratuit - *Cancel subscription* - disponible pour les nœuds payants avec un abonnement actif - *Resubscribe to paid* - affiché si un abonnement a été annulé - Aucun bouton d'action n'est affiché pour les nœuds prépayés, qui ont une longue période de validité et ne peuvent pas être annulés

Onglet des filtres d'accès

L'onglet *Filtres d'accès* vous permet de restreindre l'accès à votre nœud Zabbix par adresse IP ou sous-réseau.



Pour configurer un filtre d'accès, renseignez les champs suivants :

- *Type* - pour les nœuds serveur, utilisez la liste déroulante pour choisir le composant Zabbix à limiter : interface/API uniquement, serveur uniquement ou les deux. Pour les nœuds proxy, seul *IP entrante* est disponible. Cette option définit quelles requêtes réseau entrantes sont autorisées à atteindre le proxy.
- *Adresse IP ou CIDR* - saisissez manuellement l'adresse IP ou le CIDR, ou renseignez automatiquement le champ en sélectionnant l'option correspondante.
- *Description facultative de l'adresse IP ou du CIDR* - ajoutez une description facultative pour chaque adresse IP ou sous-réseau.

Après avoir renseigné les champs, cliquez sur *Ajouter un filtre*. Les filtres ajoutés apparaissent dans le tableau ci-dessous, avec leur état, leur type, leur adresse IP ou sous-réseau et leur description.

Warning:
Lors de l'initialisation du nœud, un filtre d'accès est automatiquement créé pour l'adresse IP que vous avez utilisée. Si vous vous connectez ultérieurement depuis une autre adresse IP ou un autre réseau, vous devez ajouter cette adresse ici, sinon votre accès sera bloqué.

Vous pouvez :

-  - modifier un filtre.
-  - supprimer un filtre ou plusieurs filtres à l'aide des cases à cocher.

Onglet Chiffrement

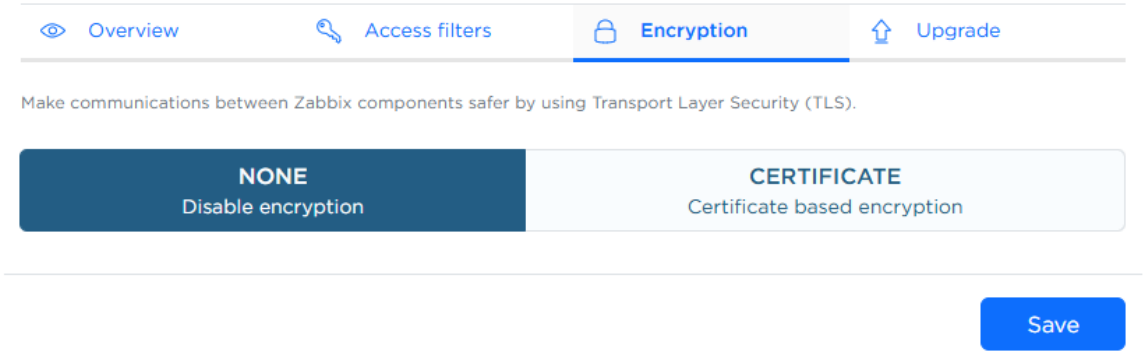
L'onglet *Chiffrement* vous permet de configurer le chiffrement TLS pour les communications entre les composants Zabbix en téléversant les fichiers de certificats requis.

Pour activer les communications sécurisées à l'aide de certificats, sélectionnez *CERTIFICATE* et téléversez les fichiers suivants depuis votre ordinateur :

- *Certificats d'autorité de certification racine* - certificats d'autorité de certification approuvés utilisés pour vérifier les certificats des pairs. Cliquez sur *Choose file* et sélectionnez le fichier contenant les certificats d'autorité de certification racine approuvés (généralement un fichier `.crt` ou `.pem`, encodé au format PEM).
- *Chaîne de certificats* - représente la chaîne de confiance de votre propre certificat. Cliquez sur *Choose file* et sélectionnez le fichier encodé au format PEM (par exemple, `.crt` ou `.pem`) qui contient votre certificat ainsi que les éventuels certificats d'autorité de certification intermédiaires.

- *Clé privée* - clé privée correspondant à votre certificat. Cliquez sur *Choose file* et sélectionnez le fichier `.key` encodé au format PEM.
- *Certificats révoqués* (facultatif) - si vous gérez une liste de révocation de certificats (CRL), cliquez sur *Choose file* et sélectionnez le fichier CRL afin de spécifier les certificats qui ne doivent plus être approuvés.

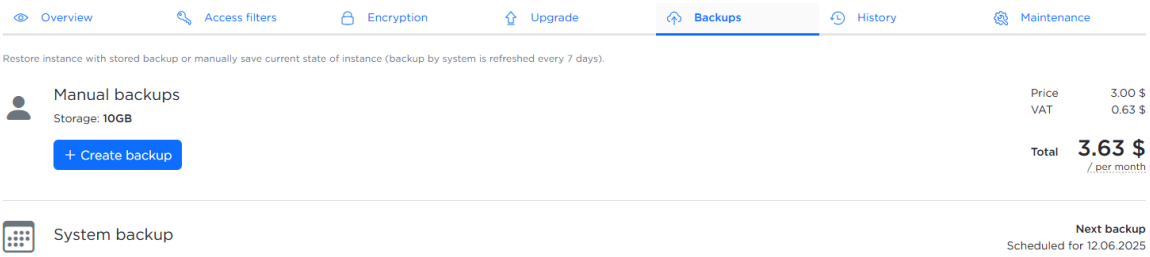
Après avoir sélectionné les fichiers, cliquez sur *Save*. Les fichiers seront téléversés, validés et appliqués pour configurer le chiffrement TLS entre les composants Zabbix.



Onglet Backups

L'onglet *Backups* vous permet d'afficher, de créer, de restaurer et de supprimer les sauvegardes de votre nœud Zabbix.

Vous pouvez restaurer votre instance à partir de n'importe quelle sauvegarde enregistrée ou créer une nouvelle sauvegarde manuelle à tout moment.
L'onglet affiche également la date de la prochaine sauvegarde planifiée.



Pour créer une sauvegarde :

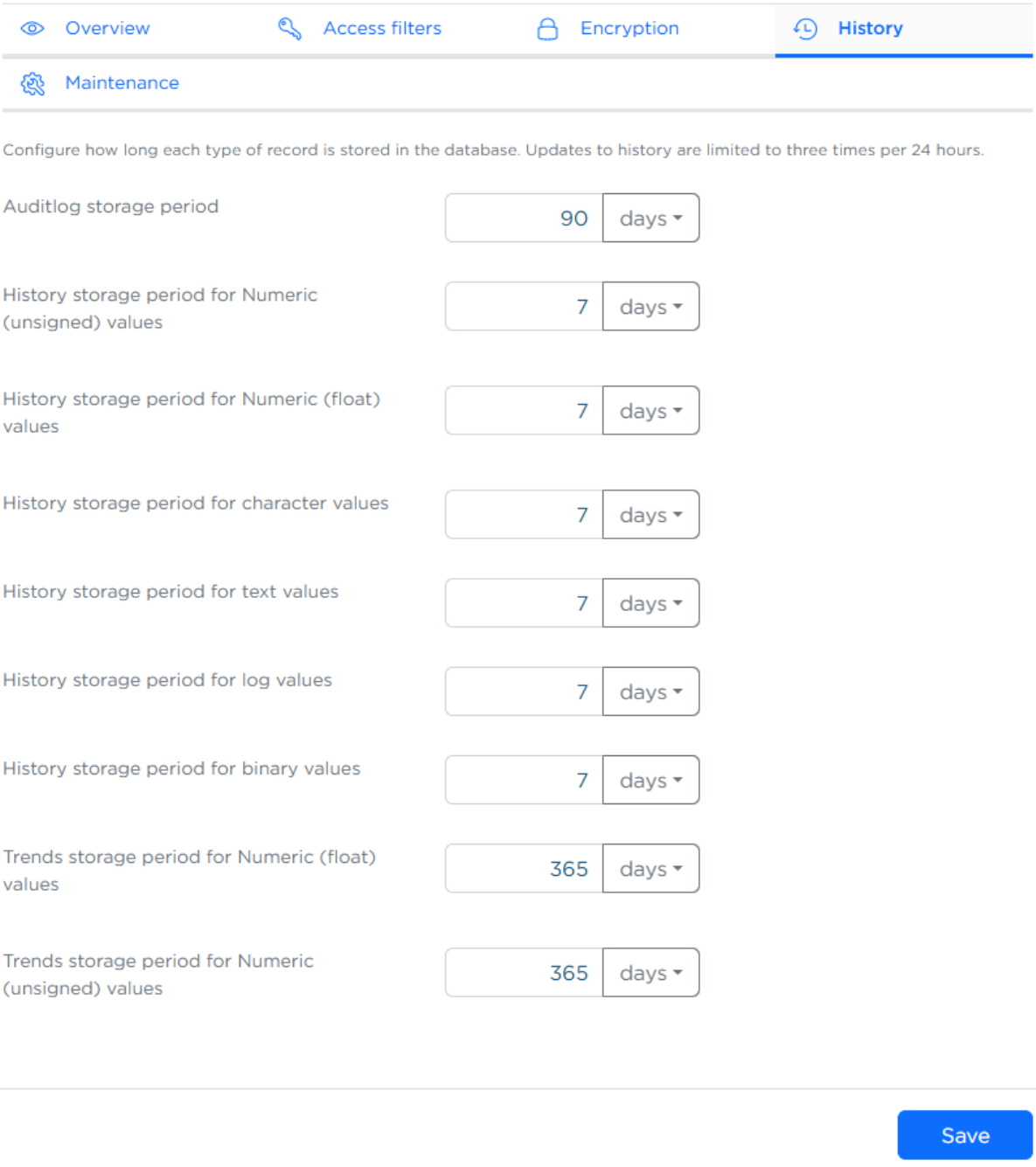
1. Cliquez sur le bouton *Create backup*.

2. Saisissez une description facultative dans la fenêtre contextuelle.
3. Cliquez sur *Create* pour confirmer.

> **Warning:**
> Les sauvegardes supplémentaires sont facturées mensuellement en fonction de l'espace de stockage utilisé.

Onglet Historique

L'onglet *Historique* vous permet de configurer la durée de conservation des différents types de données historiques.



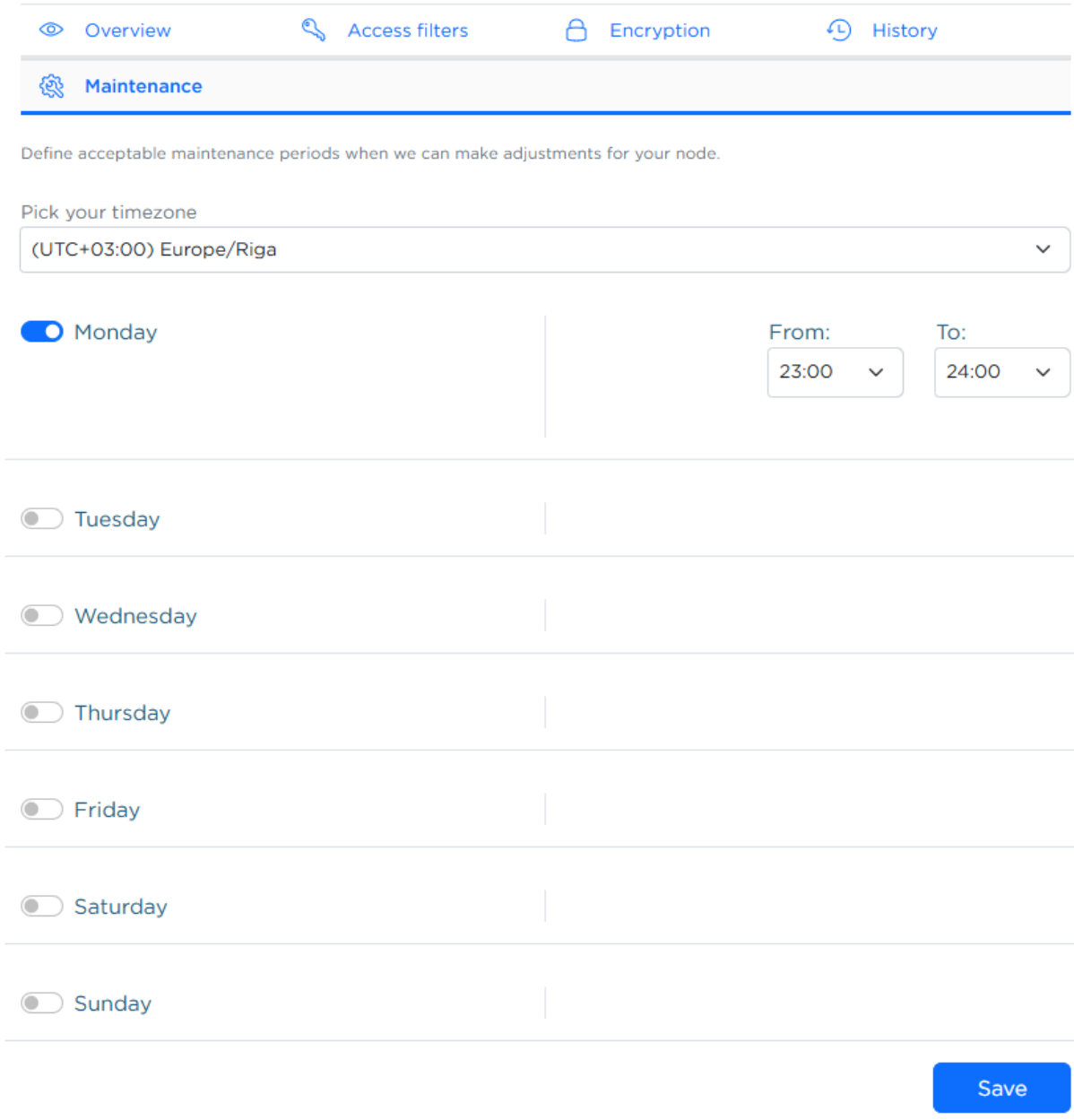
Vous pouvez définir la période de conservation pour chaque type de données (par exemple, les données numériques, le texte et les journaux), et choisir l'unité (jours, mois, années). Les modifications peuvent être effectuées jusqu'à trois fois toutes les 24 heures.

Après avoir mis à jour les valeurs, cliquez sur *Enregistrer* pour appliquer les modifications.

Warning:
Des périodes de conservation plus longues peuvent augmenter l'utilisation du disque et affecter les performances. Il est recommandé de configurer les périodes de conservation en fonction de vos données réelles et de vos besoins d'analyse.
 Zabbix Cloud utilise le partitionnement de la base de données pour gérer le stockage des données, plutôt que le processus de housekeeping traditionnel de Zabbix. Cela permet d'obtenir de meilleures performances et une gestion de la conservation plus prévisible.

Onglet Maintenance

L'onglet *Maintenance* vous permet de définir les fenêtres de maintenance acceptables pendant lesquelles des mises à jour ou des ajustements peuvent être effectués sur votre nœud.

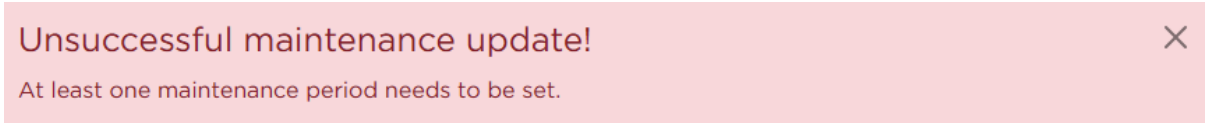


Vous devez :

- Sélectionner votre fuseau horaire dans la liste déroulante.
- Choisir au moins un jour et une plage horaire pour la maintenance.

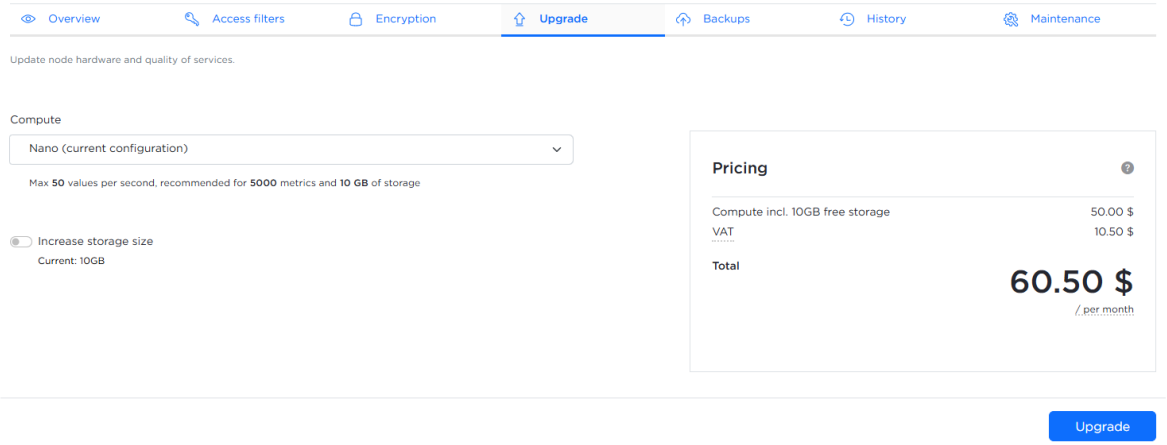
Cliquez sur *Enregistrer* pour appliquer les modifications.

Si aucun jour ni aucune plage horaire n'est sélectionné, une erreur s'affiche.



Onglet Upgrade

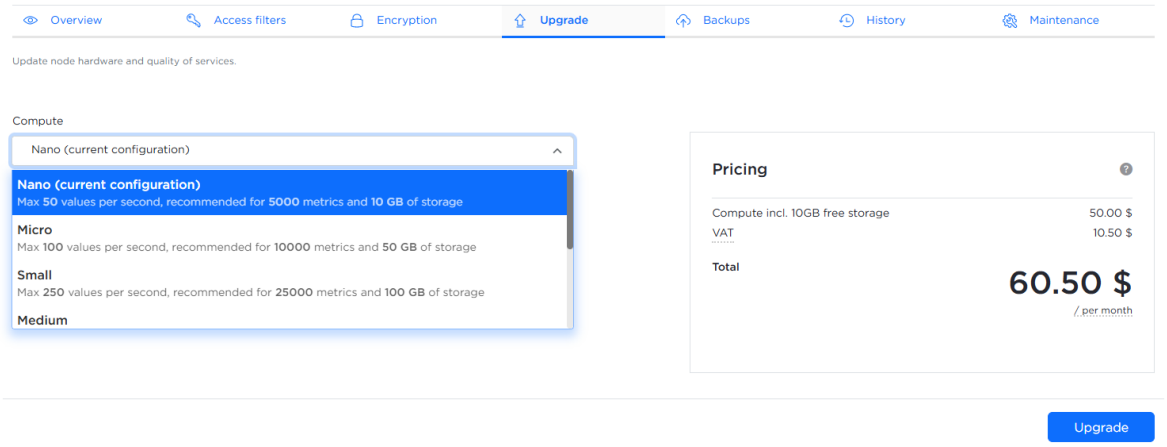
L'onglet *Upgrade* vous permet d'augmenter les ressources de votre nœud ou de gérer votre abonnement.



Vous pouvez augmenter le niveau de calcul ou la taille du disque. Notez que la taille du disque ne peut être augmentée qu'une fois toutes les 24 heures et ne peut pas être réduite via l'interface (contactez le support Zabbix pour toute demande de réduction). Les nœuds en période d'essai gratuit peuvent être convertis en abonnements payants.

Pour augmenter le niveau de calcul :

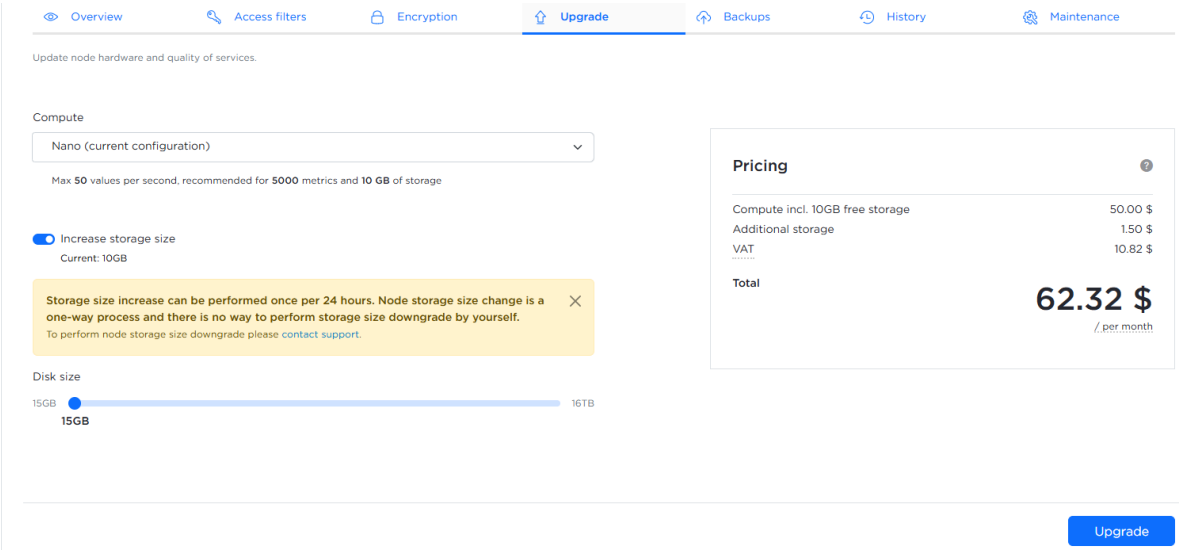
1. Cliquez sur la liste déroulante *Compute*.



2. Sélectionnez le niveau de configuration souhaité.
3. Cliquez sur le bouton *Upgrade*.

Pour augmenter la taille du disque :

1. Cliquez sur le bouton radio *Increase storage size*.
2. Ajustez le curseur *Disk size* à la valeur souhaitée.



3. Cliquez sur *Upgrade*.

Note:
Le nouveau prix mensuel total est affiché dans le tableau *Pricing* avant la confirmation des modifications.

Ajout d’utilisateurs

Introduction

Les exemples suivants décrivent comment ajouter des utilisateurs à votre organisation et à vos nœuds.

Note:
La copie, l’effacement et la réinitialisation du mot de passe du frontend ne peuvent être effectués que par le propriétaire.

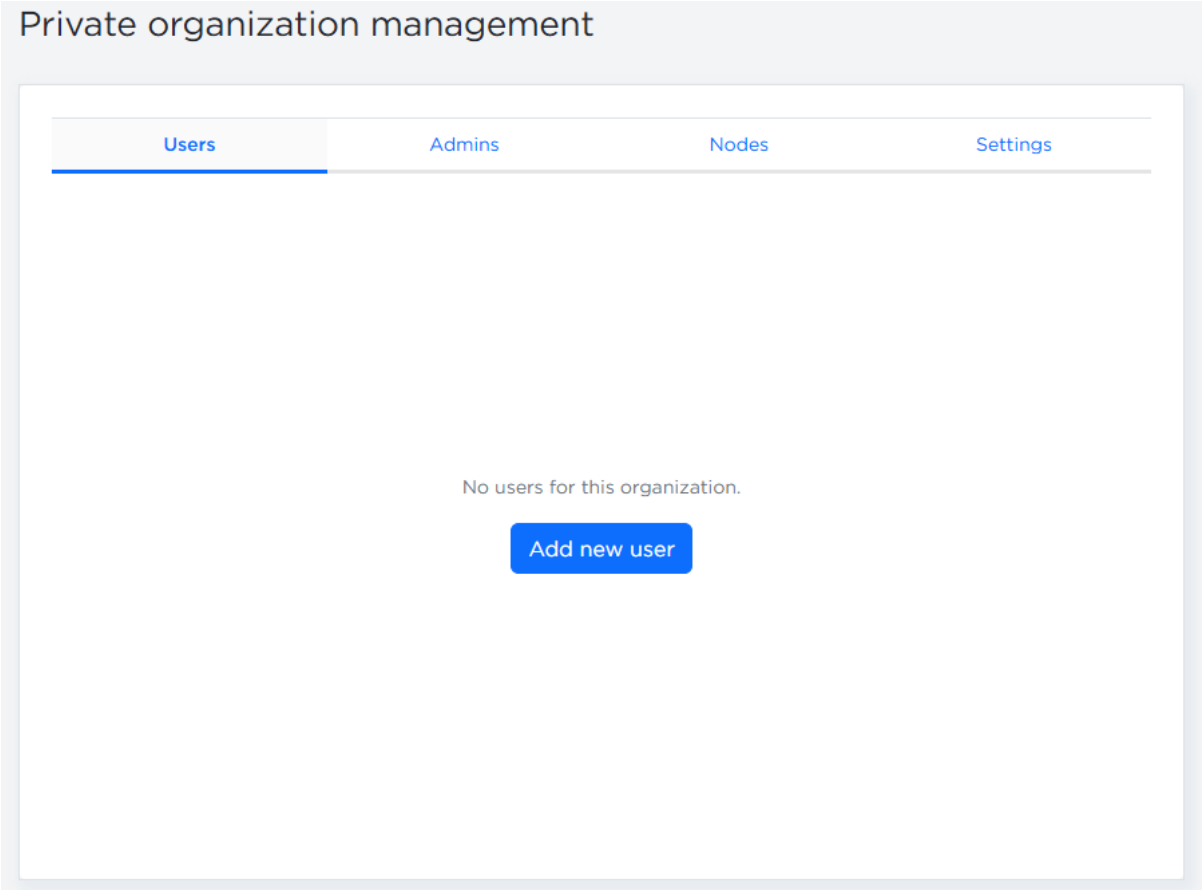
Ajout d’utilisateurs aux organisations

Comme pour les utilisateurs dans Zabbix, vous pouvez créer des utilisateurs dans Zabbix Cloud et leur attribuer des rôles afin de contrôler ce qu’ils peuvent faire au sein d’une organisation et de ses nœuds.

Pour ajouter des utilisateurs à votre organisation :

1. Ouvrez la page *Toutes les organisations*.

2. Cliquez sur le bouton des paramètres de l’organisation concernée pour accéder à son menu de gestion.



3. Dans l'onglet *Users* ou *Admins*, cliquez sur *Add new user*.

4. Saisissez l'adresse e-mail de l'utilisateur et sélectionnez un rôle : *Administrator* ou *User*.

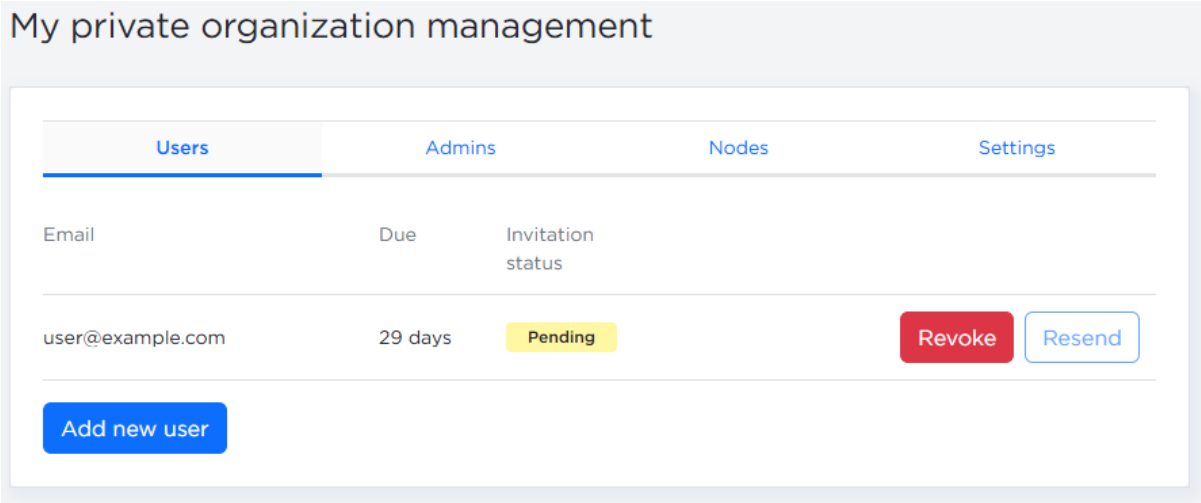
Notez que la sélection du rôle *User* ajoute l'utilisateur à l'organisation, mais ne lui accorde pas l'accès à des nœuds.

Chaque rôle accorde un niveau d'accès différent :

- **User** - peut accéder aux nœuds qui lui sont attribués et effectuer des tâches de configuration qui ne génèrent pas de frais, par exemple :
  - Gérer les filtres d'accès
  - Configurer le chiffrement
  - Définir les périodes de conservation de l'historique/des tendances
  - Planifier la maintenance
- **Administrator** - dispose de toutes les autorisations du rôle User et peut également effectuer des actions susceptibles de générer des frais, par exemple :
  - Créer et restaurer des sauvegardes
  - Augmenter ou réduire les ressources du nœud (niveau de calcul et taille du disque)
  - Annuler des abonnements
- **Owner** - dispose d'un contrôle total sur l'organisation. En plus de toutes les autorisations du rôle Administrator, le propriétaire peut :
  - Créer et détruire des nœuds
  - Ajouter ou supprimer des Users et des Administrators
  - Renommer ou supprimer l'organisation
  - Gérer les informations de facturation

Attention:
Le mode de paiement du propriétaire est débité pour tous les services. Si une confirmation bancaire est requise, le propriétaire doit autoriser la transaction, même si elle a été initiée par un Administrator.

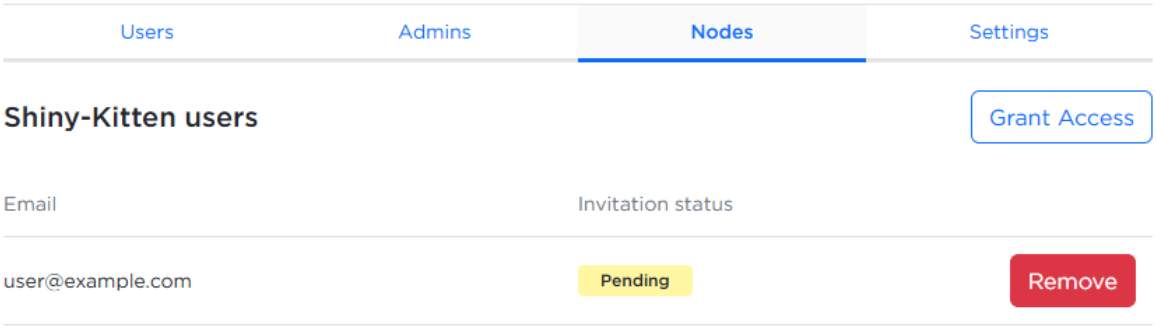
L'utilisateur invité sera ajouté avec le statut *Pending*. Vous pouvez révoquer l'invitation ou la renvoyer (le renvoi est possible une fois toutes les 10 minutes — le bouton sera désactivé si cette limite est atteinte).



Ajouter des utilisateurs aux nœuds

Pour ajouter l'utilisateur au nœud :

1. Accédez à l'onglet *Nodes*.
2. Cliquez sur *Grant Access*.
3. Sélectionnez dans la liste déroulante un utilisateur ajouté à l'organisation.
4. Cliquez sur *Grant access*.



Vous pouvez également accorder l'accès depuis la fenêtre *Node configuration* :

1. Cliquez sur le bouton de paramètres .
2. Sélectionnez *Add user*.
3. Choisissez un utilisateur et cliquez sur *Grant access*.

Les utilisateurs invités doivent accepter l'invitation en créant un compte Zabbix Cloud avec la même adresse e-mail que celle à laquelle l'invitation a été envoyée. Si l'utilisateur possède déjà un compte Zabbix Cloud associé à cette adresse e-mail, il doit se connecter et confirmer l'invitation. Les invitations sont liées à une adresse e-mail spécifique et ne peuvent pas être acceptées depuis un autre compte.

Principales différences de Zabbix Cloud

Introduction

Zabbix Cloud est une plateforme basée sur le cloud permettant à ses utilisateurs d'exécuter des instances Zabbix autonomes dans le cloud (nœuds cloud).

Par rapport à un déploiement Zabbix classique sur site :

- Aucune installation ni maintenance n'est requise.
- Les nœuds cloud sont entièrement gérés par Zabbix.

- Les nœuds cloud peuvent être démarrés en quelques minutes.

Ce service est proposé selon un modèle de paiement à l'usage (un **essai gratuit** est disponible).

Tableau comparatif Bien que la plupart des fonctionnalités se recoupent entre les installations Zabbix sur site et les nœuds cloud, il existe également des différences. Un tableau comparatif récapitulatif est présenté ci-dessous. Les sections suivantes décrivent certains éléments en détail.

Fonctionnalité / Aspect	Zabbix Cloud	Zabbix sur site
Cycle de publication	Synchronisé avec les versions LTS	Vous choisissez votre version de Zabbix (stable, bêta, compilations personnalisées)
Facilité d'utilisation	Plateforme prête à l'emploi, entièrement gérée par Zabbix	Nécessite une installation, une configuration et une maintenance
Application des correctifs de vulnérabilité	Application automatique des correctifs du système d'exploitation et de Zabbix	Application manuelle des correctifs par l'utilisateur
Certificat HTTPS	Certificat valide prêt à l'emploi (sans Let's Encrypt)	Certificat autosigné par défaut (Let's Encrypt ou une autre autorité de certification peut être configuré)
Performances de la base de données (INSERT/UPDATE/SELECT)	Partitionnement automatique pour une vitesse maximale ; aucune opération DELETE de longue durée	Partitionnement et nettoyage manuels (transactions DELETE de longue durée possibles)
Optimisation des performances	Préconfigurée par l'équipe Zabbix Cloud (sans journaux des modifications)	Visibilité et contrôle complets des paramètres d'optimisation
Contrôle du pare-feu	Géré par l'équipe Zabbix Cloud ; les ports de l'interface et du trapper ne sont pas exposés publiquement	Contrôle complet des règles du pare-feu et de l'exposition des ports
Trafic sortant sur le port 25	Bloqué par défaut pour empêcher le spam et les abus ; utilisez plutôt le port SMTP 587 ou 465	Non restreint par Zabbix ; dépend de votre réseau et de votre fournisseur d'hébergement
Isolation et protection de la base de données	Isolation élevée, aucun accès direct à la base de données ou via SSH	Dépend de votre réseau et de la sécurité de votre hôte
Haute disponibilité (HA)	S'exécute sur une plateforme cloud hautement disponible, les proxys Zabbix peuvent être utilisés pour une meilleure résilience de la collecte des données	HA gérée par l'utilisateur
Traps SNMP	Uniquement via un proxy dédié (aucun récepteur de traps en HA ou à équilibrage de charge)	Directement ou via un proxy, HA et équilibrage de charge possibles
Interrogation SNMP	Difficile sans proxys (chaque périphérique nécessite un NAT ou un port personnalisé)	Interrogation SNMP native ; proxy facultatif
Paramètres de conservation	Historique/Tendances/Audit uniquement via l'interface (aucune API ni fichier de configuration)	Configurable dans le fichier de configuration du serveur Zabbix ou via l'API
Scripts personnalisés	`AlertScriptsPath`, `ExternalScripts`, modules de l'interface et modules communautaires non pris en charge	Entièrement pris en charge (chemin des scripts, modules, intégrations)
Supervision ODBC	Uniquement PostgreSQL (pilote `{postgresql}`) ; le plugin MariaDB est présent comme espace réservé	ODBC pour PostgreSQL, MySQL, Oracle, etc. - entièrement configurable
Limitation des appels ODBC	Impossible (`StartODBCPollers=1` uniquement) ; les requêtes de synchronisation lourdes bloquent les autres tâches	Nombre de pollers et planification entièrement configurables
Téléversement du certificat SAML	Non pris en charge	Pris en charge via l'interface ou l'API
Rapports planifiés	Vous devez créer votre propre média Email (les médias de type script ne sont pas pris en charge)	Prend en charge les médias de type script et Email prêts à l'emploi
Interface de l'hôte pour les vérifications actives	Adresse IP attribuée par la plateforme (créée automatiquement)	Interfaces gérées par vos soins ; adresses IP sous votre contrôle

Différences entre les fonctionnalités Traps SNMP

Les traps SNMP ne sont pris en charge que par l'intermédiaire d'un proxy Zabbix dédié. Si la surveillance des traps SNMP est requise, l'équilibrage de charge automatisé ou la haute disponibilité pour le proxy ne peuvent pas être utilisés, car les traps SNMP doivent être envoyés à une seule adresse IP.

Interrogation SNMP

Sans proxys, l'interrogation SNMP nécessite que chaque appareil soit exposé via NAT et des ports personnalisés.

Paramètres de rétention

Les périodes de rétention de l'historique, des tendances et du journal d'audit ne peuvent être configurées que via l'interface web dans le cloud. Ces paramètres ne peuvent pas être configurés via `zabbix_server.conf` ni via l'API.

Les remplacements manuels de l'historique par élément ne sont pas pris en charge (le partitionnement est contrôlé globalement). Les URL des nœuds cloud n'acceptent pas les mêmes paramètres de requête que les installations sur site.

Scripts personnalisés

Les types de scripts personnalisés suivants ne sont pas pris en charge par Zabbix Cloud :

- Scripts d'alerte (`AlertScriptsPath`)
- Scripts externes (`ExternalScripts`)
- Scripts frontend sur Zabbix Server

Les modules frontend développés par la communauté ne peuvent pas être installés.

Supervision ODBC

Zabbix Cloud prend en charge la supervision ODBC uniquement pour PostgreSQL. Utilisez le modèle ODBC officiel de Zabbix et définissez la chaîne de connexion dans le modèle en utilisant :

```
Driver={postgresql}
```

Un plugin MariaDB est également installé, mais il n'est actuellement pas fonctionnel. S'il est utilisé, définissez :

```
Driver={mysql}
```

> **Attention:**
> Il n'existe aucune configuration connue fonctionnelle pour Oracle MySQL 8.0 dans Zabbix Cloud. Bien que des requêtes simples (telles que `SELECT 1`) puissent réussir, des requêtes plus complexes renvoient `SQL_ERROR.<br><br>` Il n'est pas possible de limiter les appels ODBC (`StartODBCPollers=1` uniquement). Des rapports synchrones lourds peuvent affecter les performances, et une seule instruction `SELECT` peut être exécutée à la fois.

Accès à l'infrastructure

Zabbix Cloud ne fournit pas d'accès SSH aux nœuds sous-jacents et n'autorise pas non plus les connexions directes à la base de données (par exemple, vers le port 3306). Toutes les activités de configuration, de surveillance et de dépannage doivent être effectuées via l'interface utilisateur du cloud ou l'API, afin de garantir que les couches du système d'exploitation et de la base de données restent isolées et sécurisées.

Trafic sortant sur le port 25

Pour prévenir le spam et les abus, les connexions sortantes des nœuds Zabbix Cloud vers le port TCP 25 sont bloquées par défaut. Cela concerne :

- Les types de média Email qui envoient des messages via un serveur SMTP sur le port 25. Comme 25 est le *port du serveur SMTP* par défaut pour un nouveau type de média Email, remplacez-le par 587 (avec la *sécurité de la connexion* définie sur STARTTLS) ou 465 (SSL/TLS), selon ce que votre serveur SMTP prend en charge. Le fournisseur d'e-mails *Office365 relay* ne peut pas être utilisé, car il se connecte au point de terminaison de relais de Microsoft sur le port 25.
- Les rapports planifiés, qui nécessitent un type de média Email.
- Les vérifications effectuées par le nœud cloud lui-même sur le port 25, telles que `net.tcp.service[smtp]`, des vérifications simples ou des vérifications SMTP dans les règles de découverte réseau. Ces vérifications indiquent que le service SMTP est arrêté même lorsqu'il fonctionne ; effectuez plutôt ces vérifications via un proxy Zabbix ou un agent Zabbix dans votre propre réseau.

Le type de média *Cloud Email* par défaut n'est pas concerné.

Interface de vérification active

Les vérifications actives dans le cloud créent automatiquement une interface d'hôte avec une adresse IP sans lien avec votre réseau sur site. Par défaut, cette adresse IP peut être externe. Pour conserver la cohérence, vous pouvez supprimer ou ajuster manuellement l'interface d'hôte après sa création.

Téléversement du certificat SAML

L'authentification SAML n'est pas prise en charge, car peu de fournisseurs acceptent les messages SAML non chiffrés ou non signés.

Rapports planifiés

Le type de média *Cloud Email* par défaut utilise un transport par script et ne peut pas être utilisé pour les rapports planifiés. Pour envoyer des rapports par e-mail :

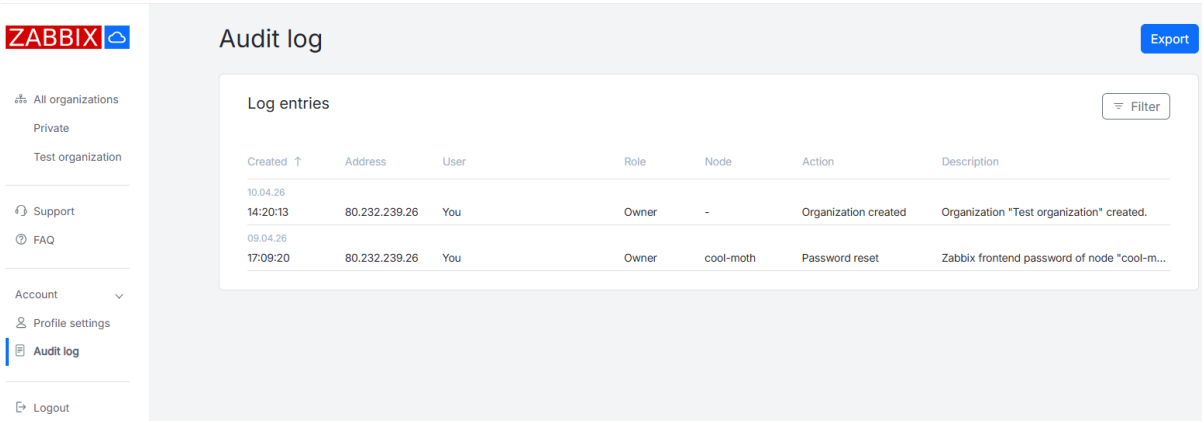
1. Créez un nouveau type de média E-mail.
2. Attribuez le média e-mail à votre utilisateur dans *Paramètres utilisateur* > *Média*.
3. Sélectionnez ce type de média lors de la configuration des rapports planifiés.

Attention:
N'utilisez pas le port 25 comme *port du serveur SMTP* pour ce type de média ; consultez Trafic sortant sur le port 25.

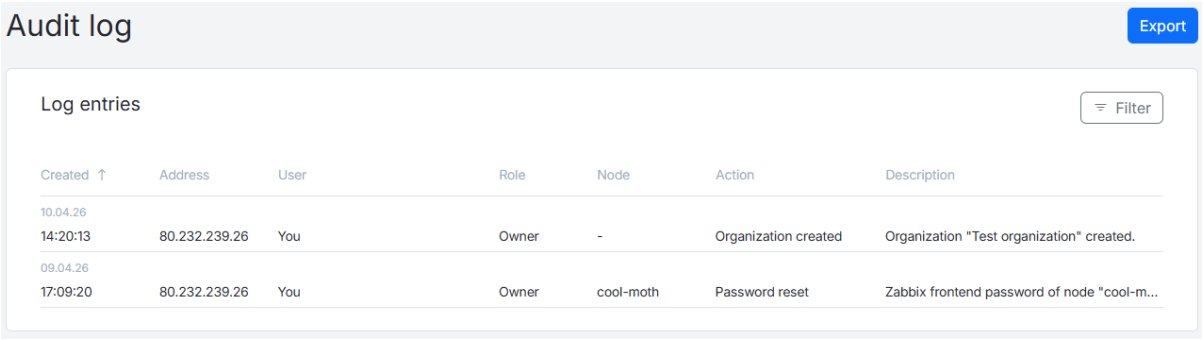
Journal d'audit

Vue d'ensemble Dans Zabbix Cloud, le journal d'audit enregistre l'activité des utilisateurs et du système.

Après vous être connecté à un compte Zabbix Cloud, accédez à *Account* > *Audit log* pour afficher les enregistrements du journal d'audit. Les entrées du journal d'audit disponibles pour un utilisateur dépendent du rôle de son organisation.



Affichage des enregistrements du journal d'audit Les enregistrements du journal d'audit sont affichés dans un tableau.



Le tableau contient les informations suivantes :

Column	Description
Created	Date et heure de création de l'entrée du journal. L'heure affichée correspond au fuseau horaire de l'utilisateur sélectionné dans la liste déroulante *User timezone* dans *Account* > *Profile settings*.
User	Nom de l'utilisateur de l'organisation qui a effectué l'action.
Role	Le rôle de l'utilisateur qui a effectué l'action.
Node	Nom du nœud lié à l'action.
Action	Nom de l'action effectuée.
Description	Description de l'action effectuée.

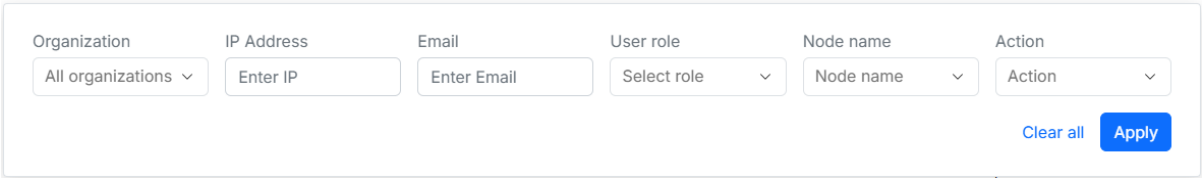
Les noms des colonnes peuvent être cliqués pour trier les enregistrements par ordre croissant ou décroissant.

Les enregistrements du journal d'audit filtrés peuvent être exportés au format `.csv` à l'aide du bouton *Export*.

Utilisation des filtres

Utilisez le bouton *Filtre* pour ouvrir le filtre.

Pour afficher les résultats filtrés, cliquez sur *Appliquer* après avoir sélectionné un ou plusieurs paramètres de filtrage. Pour effacer le filtre, cliquez sur *Tout effacer*.



Il est possible de filtrer les enregistrements selon un ou plusieurs paramètres. Les paramètres de filtrage suivants sont disponibles :

Paramètre	Description
Organisation	Sélectionnez toutes les organisations ou une organisation spécifique.
Adresse IP	Indiquez une adresse IP.
E-mail	Indiquez une adresse e-mail associée à l'utilisateur.
Rôle utilisateur	Sélectionnez tous les rôles ou un rôle spécifique : Propriétaire, Administrateur ou Utilisateur.
Nom du nœud	Sélectionnez tous les nœuds ou un nœud spécifique.
Action	Sélectionnez un type d'action spécifique.

Le journal d'audit suit les actions suivantes :

- *Nœud créé*
- *Nœud supprimé*
- *Nœud mis à niveau*
- *Version du nœud mise à jour*
- *Mot de passe réinitialisé*
- *Maintenance mise à jour*
- *Paramètre de l'historique mis à jour*
- *Date d'expiration du nœud prolongée*
- *Configuration du proxy mise à jour*
- *Sauvegarde créée*
- *Sauvegarde supprimée*
- *Sauvegarde restaurée*
- *Sauvegarde mise à jour*
- *Filtre d'accès ajouté*
- *Filtre d'accès mis à jour*
- *Filtre d'accès supprimé*
- *Chiffrement activé*
- *Chiffrement désactivé*
- *Abonnement annulé*
- *Abonnement renouvelé*
- *Échec du paiement de l'abonnement*
- *Paiement nécessitant une confirmation*
- *Organisation créée*
- *Organisation supprimée*
- *Administrateur invité*
- *Utilisateur invité*
- *Administrateur rejoint*
- *Utilisateur rejoint*
- *Administrateur retiré de l'organisation*
- *Utilisateur retiré de l'organisation*
- *Nœud ajouté à l'organisation*
- *Nœud retiré de l'organisation*
- *Accès au nœud accordé*
- *Accès au nœud révoqué*

- *Paramètres utilisateur mis à jour*
- *Paiement créé*
- *Paiement effectué*
- *Échec du paiement*
- *Frais de TVA appliqués*
- *Échec de la validation de la TVA*
- *Informations de facturation mises à jour*
- *Mode de paiement mis à jour*
- *Mode de paiement supprimé*
- *Nœud prépayé créé*
- *Nœud prépayé mis à jour*
- *Incident commencé*
- *Incident terminé*

Configurer un proxy Zabbix Cloud

Vue d'ensemble Cette page décrit comment configurer un proxy Zabbix déployé sur Zabbix Cloud afin de collecter les données de supervision des hôtes dans un environnement distant.

La procédure est principalement destinée à un serveur Zabbix et à un proxy créés sur Zabbix Cloud.

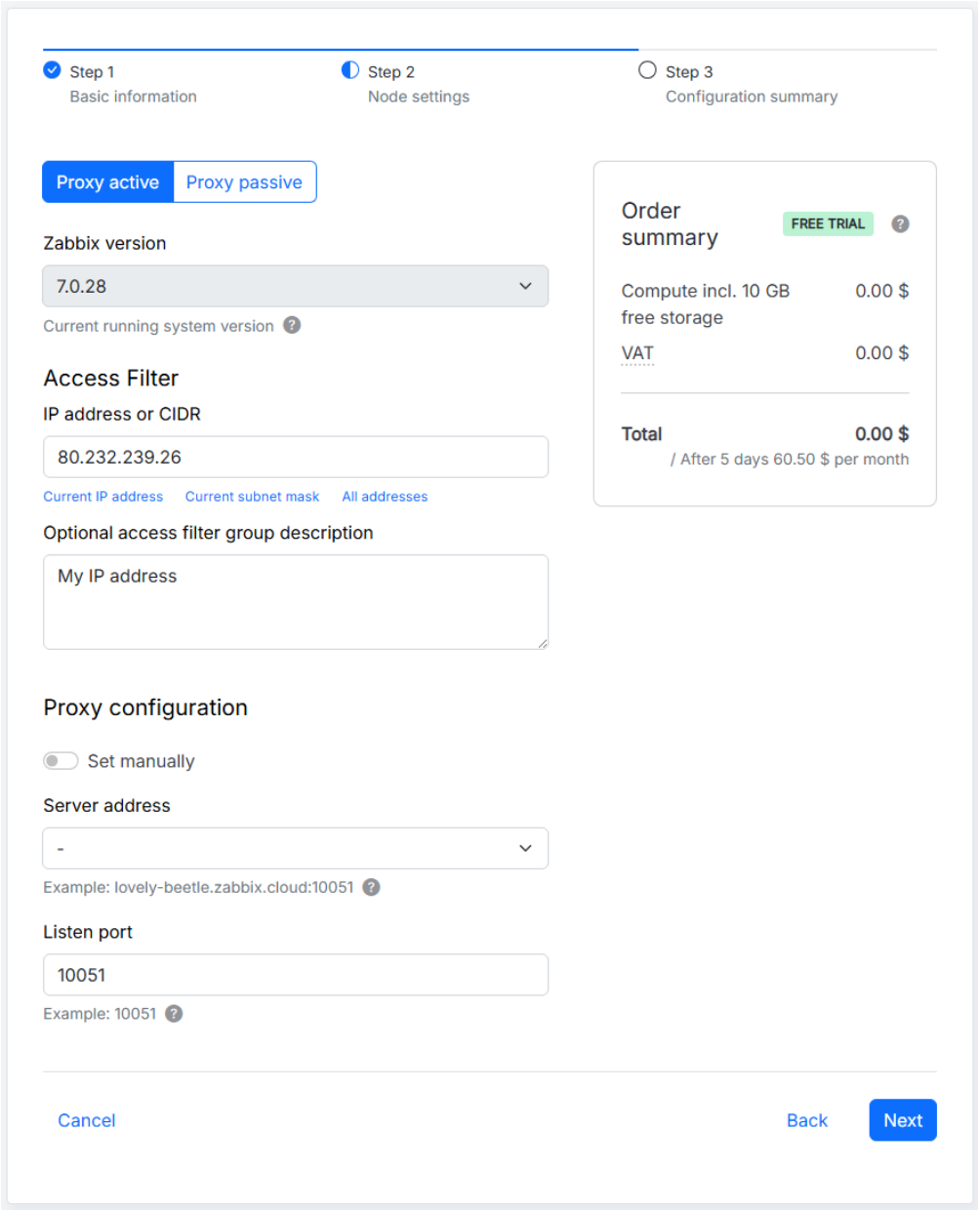
La procédure décrit également la configuration requise lorsque le serveur Zabbix n'est pas hébergé sur Zabbix Cloud.

Prérequis Un serveur Zabbix est installé, configuré et en cours d'exécution, ou un nœud de serveur Zabbix est en cours d'exécution sur Zabbix Cloud.

Créer un nœud proxy Créez un nœud proxy Zabbix Cloud.

Lors de la création du nœud :

1. Sélectionnez *Proxy*.
2. Sélectionnez *Proxy actif* ou *Proxy passif* selon la manière dont le proxy doit communiquer avec le serveur Zabbix.
3. Configurez les filtres d'accès par adresse IP ou CIDR du nœud (cette opération peut être effectuée après la création du nœud).
4. Indiquez le serveur Zabbix auquel le proxy se connectera.



5. Terminez le processus de création du nœud.

Pour plus d'informations sur la création d'un nœud proxy, consultez Déployer Zabbix dans le cloud.

Configurer les filtres d'accès Zabbix Cloud utilise des filtres d'accès pour contrôler les adresses IP ou les sous-réseaux qui peuvent se connecter à un nœud cloud.

Lors de l'initialisation du nœud, un filtre d'accès est automatiquement créé pour l'adresse IP utilisée pour accéder au nœud. Cela n'inclut pas nécessairement les adresses des hôtes surveillés ou d'autres composants Zabbix. Ajoutez manuellement les adresses requises.

Consultez Configuration du nœud — Filtres d'accès pour plus d'informations.

Nœud proxy

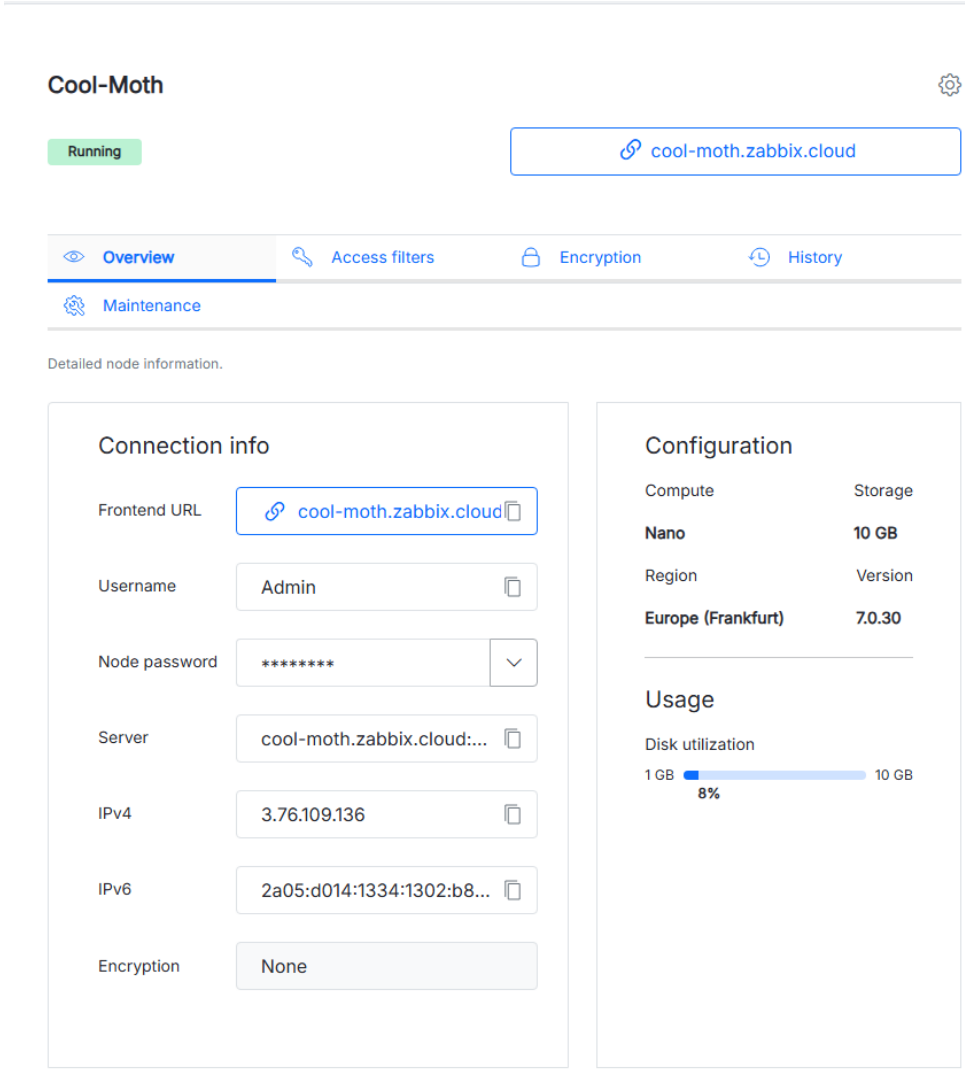
Ajoutez les adresses IP ou les sous-réseaux à partir desquels les connexions au proxy sont attendues.

Si les hôtes surveillés utilisent des vérifications actives de l'agent, les agents initient les connexions au proxy. L'adresse IP ou le sous-réseau des hôtes surveillés doit donc être autorisé par les filtres d'accès du nœud proxy.

Si *Proxy Passive* a été défini comme mode du proxy, le serveur Zabbix se connecte au proxy. L'adresse IP ou le sous-réseau du serveur Zabbix doit donc être autorisé par les filtres d'accès du nœud proxy.

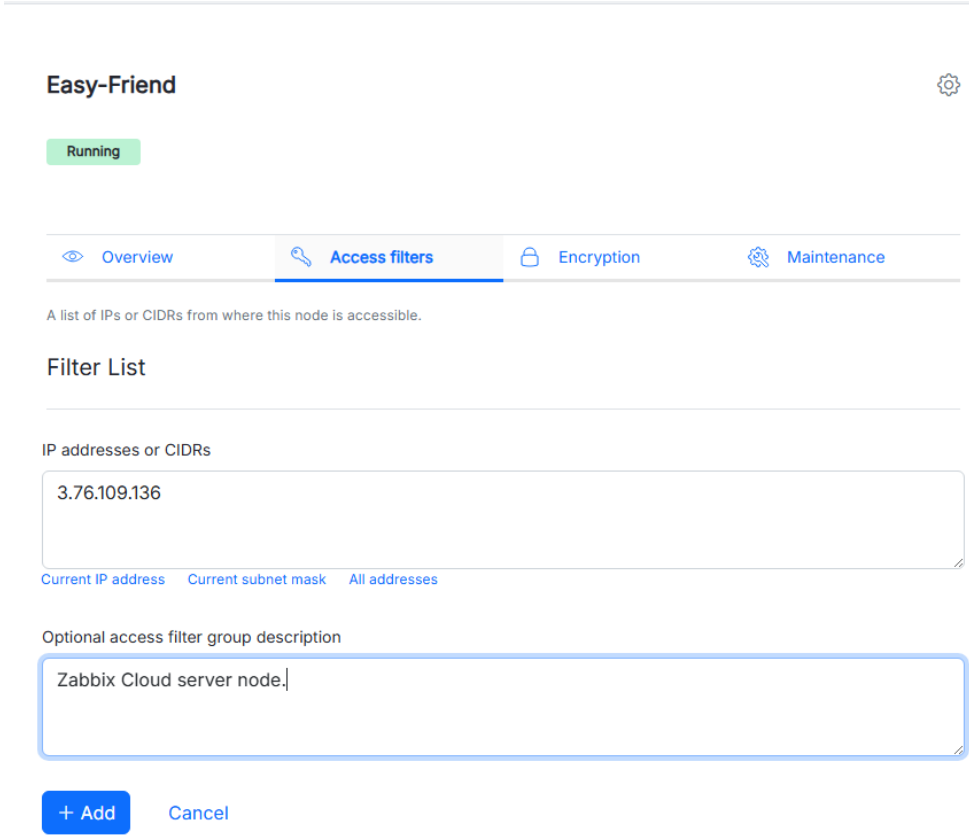
Si *Proxy passive* a été sélectionné et qu'un nœud serveur est utilisé :

1. Accédez à l'onglet *Overview* de votre nœud serveur et copiez l'adresse IPv4 ou IPv6 du nœud.

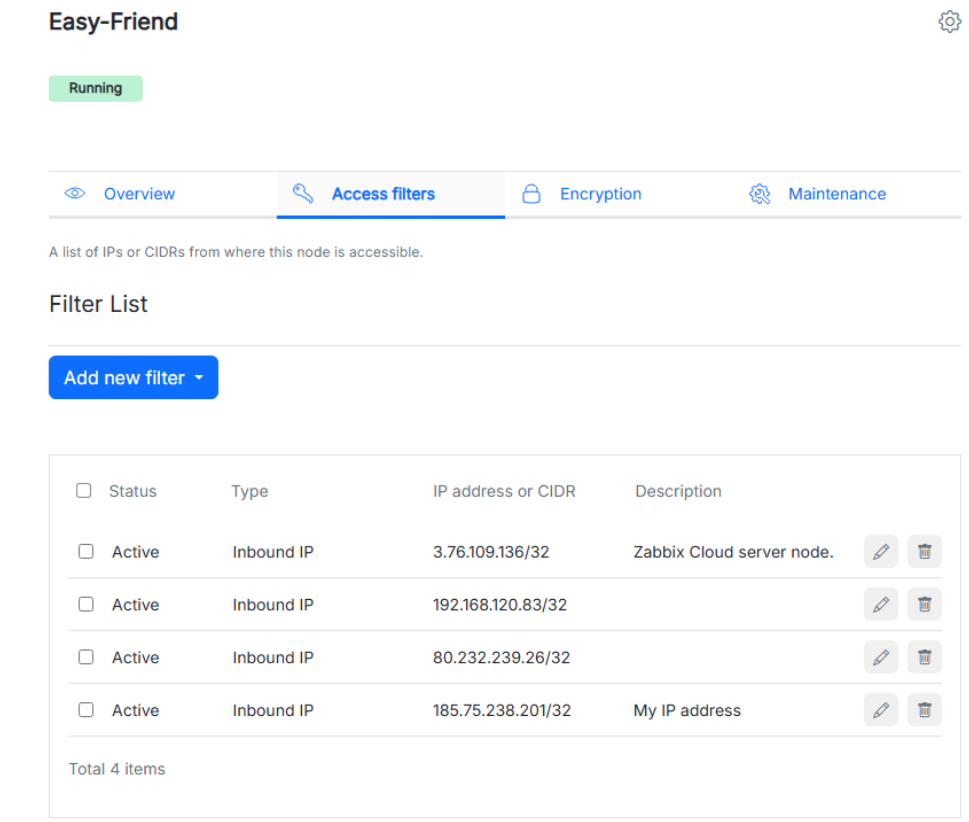


2. Accédez à l'onglet *Access filters* de votre nœud proxy, puis cliquez sur *Add new filter* > *Add*.

3. Collez l'adresse copiée dans le champ *IP addresses or CIDRs* et ajoutez une description dans le champ *Optional access filter group description*.



4. Cliquez sur *Add*.



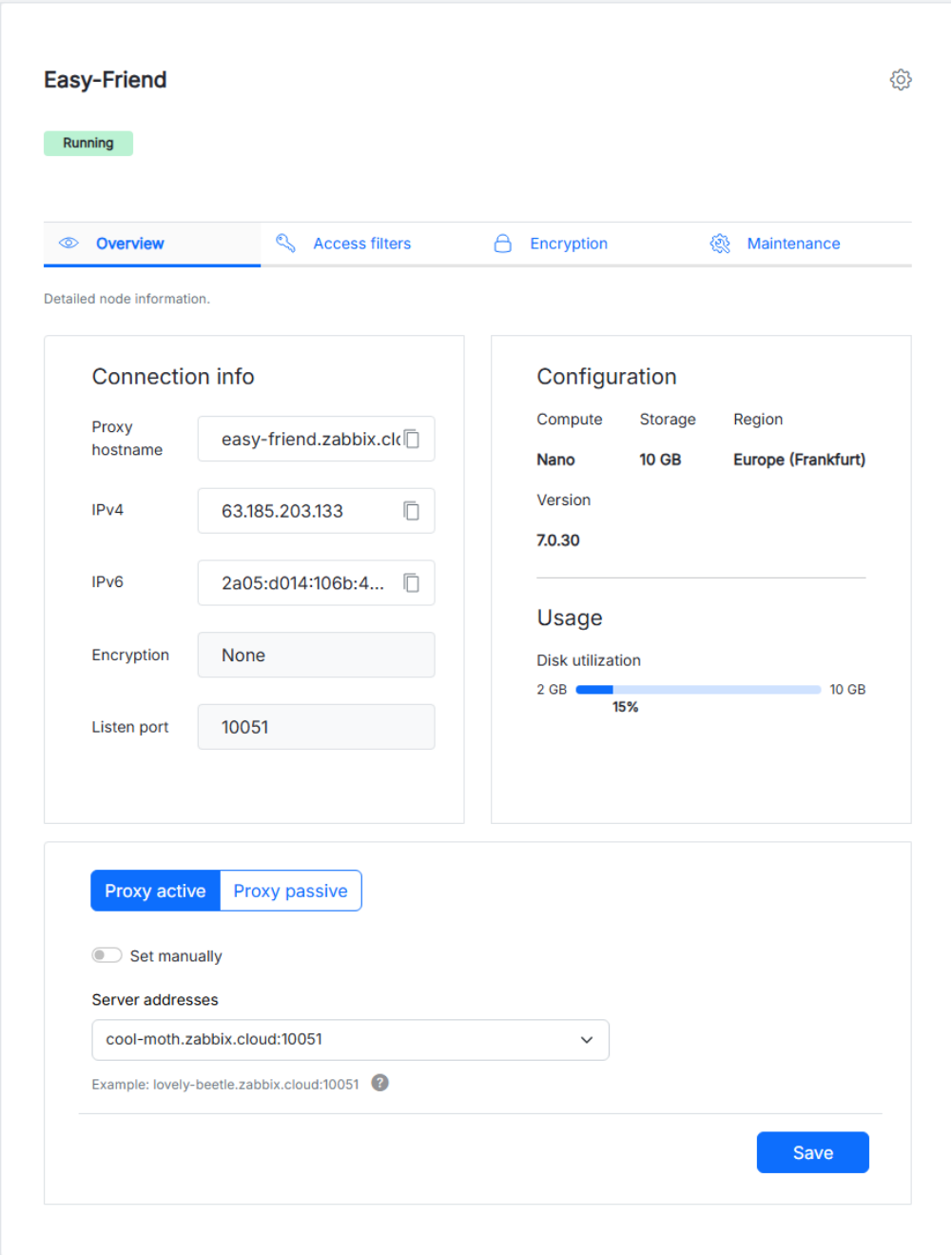
Effectuez les étapes 2 à 4 pour l'adresse IP d'un agent Zabbix utilisant des vérifications actives.

Nœud du serveur

Si le serveur Zabbix est hébergé sur Zabbix Cloud, configurez également les filtres d'accès sur le nœud du serveur.

Pour un proxy en mode actif, ajoutez l'adresse IP ou le CIDR du nœud proxy aux filtres d'accès du nœud du serveur, car le proxy initie la connexion au serveur.

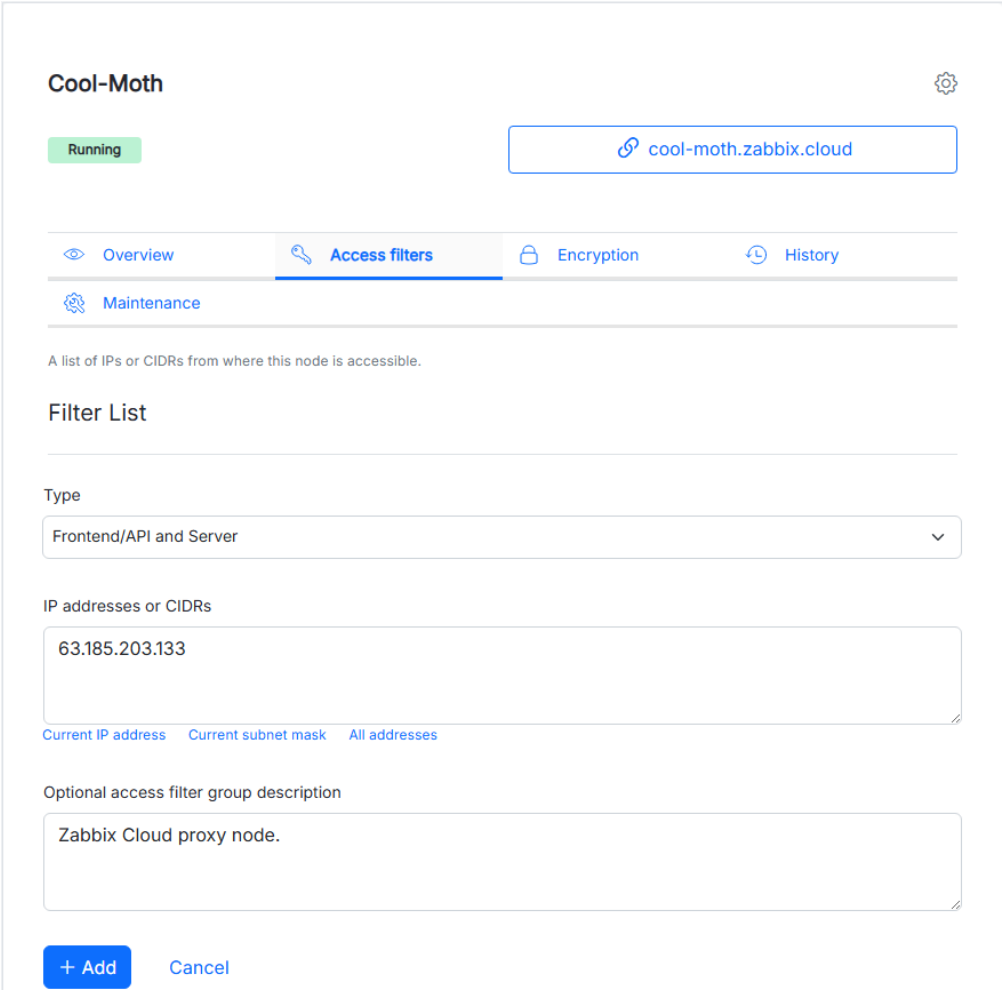
1. Accédez à l'onglet *Overview* de votre nœud proxy et copiez l'adresse IPv4 ou IPv6 du nœud.



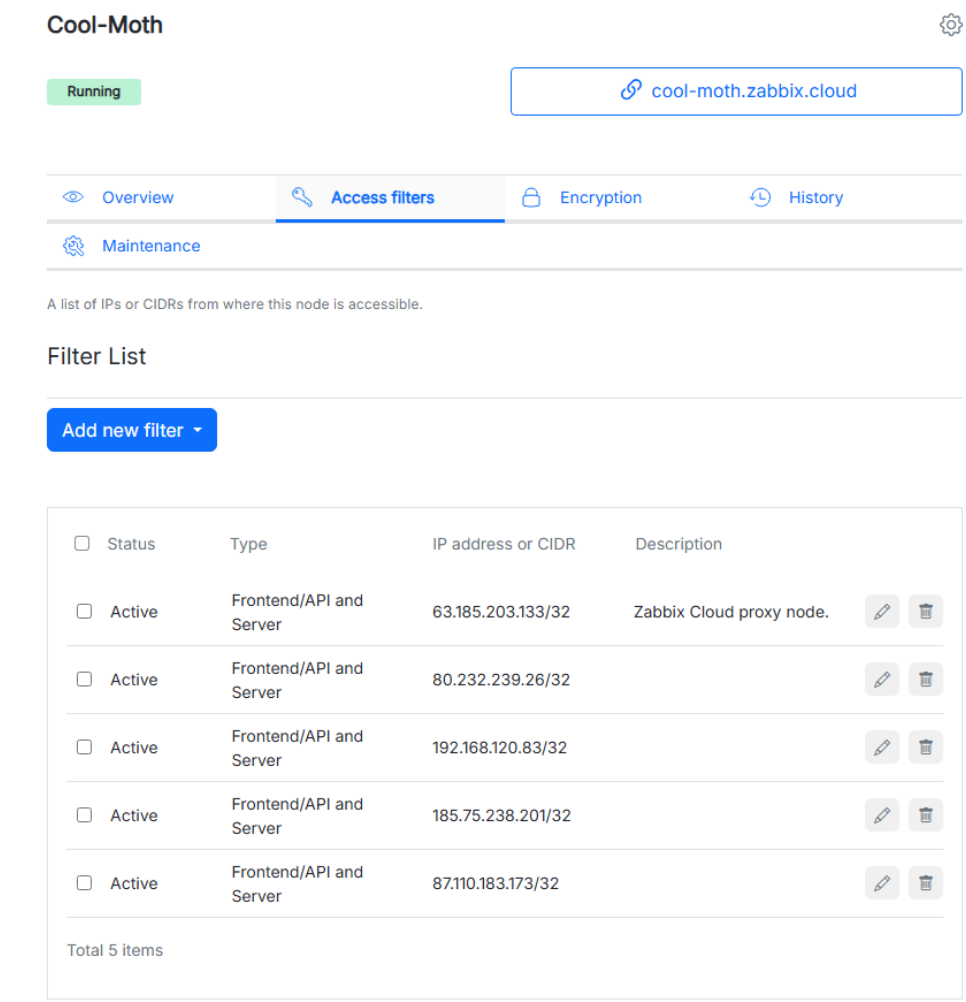
2. Accédez à l'onglet *Access filters* de votre nœud du serveur, puis cliquez sur *Add new filter* > *Add*.

3. Collez l'adresse copiée dans le champ *IP addresses or CIDRs* et ajoutez une description dans le champ *Optional access filter group description*.

4. Sélectionnez le type de filtre approprié.



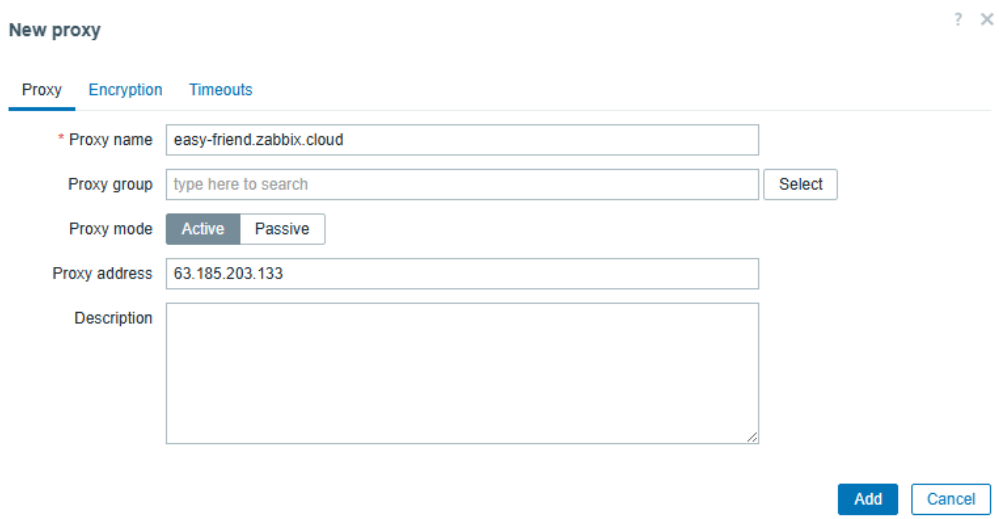
5. Cliquez sur *Add*.



Configurer le proxy Le proxy doit être configuré dans l'interface Zabbix pour communiquer avec le serveur Zabbix.

Ouvrez *Administration* > *Proxies* dans l'interface Zabbix, cliquez sur *Create proxy* et configurez le proxy selon vos besoins :

1. Ajoutez le nom du proxy dans le champ *Proxy name*. Il doit correspondre au nom d'hôte du proxy configuré pour le proxy. Saisissez-le ou copiez-le depuis l'onglet *Overview* de votre proxy.
2. Sélectionnez le mode du proxy correspondant à celui sélectionné lors de la création du proxy.
3. Pour un proxy actif, saisissez l'adresse IP ou le nom DNS du proxy.
4. Cliquez sur *Add*.



Pour plus d'informations, consultez Surveillance distribuée à l'aide de proxies.

Configurer les hôtes surveillés Ajoutez les hôtes qui doivent être surveillés via le proxy.

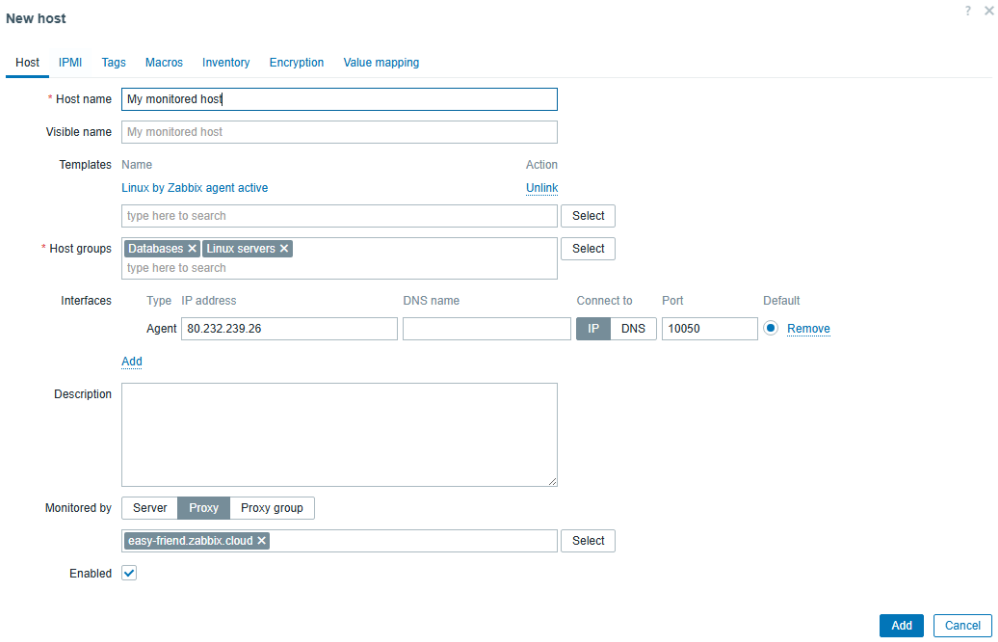
1. Accédez à *Collecte de données* > *Hôtes*.

2. Créez un hôte ou ouvrez un hôte existant.

3. Dans *Surveillé par*, sélectionnez le *Proxy* et saisissez le nom du proxy pour le rechercher et le sélectionner, ou cliquez sur le bouton *Sélectionner* pour ouvrir une liste de proxies parmi lesquels effectuer votre choix.

4. Configurez l'interface de l'hôte et les autres paramètres.

5. Cliquez sur *Ajouter* ou *Mettre à jour*.



Un hôte configuré pour être surveillé par un proxy envoie ou reçoit les données de surveillance via ce proxy, selon le type d'élément et le mode de l'agent.

Configurer l'agent Zabbix Lorsqu'un hôte surveillé utilise l'agent Zabbix ou Zabbix agent 2, configurez l'agent pour communiquer avec le composant Zabbix approprié.

La configuration dépend du fait que l'agent utilise des vérifications *passives* ou *actives*.

Vérifications passives de l'agent

Avec les vérifications passives, le proxy Zabbix initie les connexions vers l'agent et demande les données de supervision.

Ouvrez le fichier de configuration de l'agent Zabbix et mettez à jour le paramètre `Server` afin d'autoriser les connexions depuis l'adresse IP du proxy Zabbix Cloud.

Par exemple :

```
Server=<proxy_IP>
```

Assurez-vous que l'hôte supervisé autorise les connexions TCP entrantes depuis le proxy sur le port de l'agent, qui est 10050 par défaut.

Le proxy doit pouvoir atteindre l'hôte supervisé via le réseau.

Vérifications actives de l'agent

Avec les vérifications actives, l'agent établit des connexions avec le proxy Zabbix et envoie les données de supervision collectées.

Ouvrez le fichier de configuration de l'agent Zabbix et mettez à jour le paramètre `ServerActive` avec l'adresse du proxy Zabbix Cloud.

Par exemple :

`ServerActive=<proxy_IP>:<proxy_listen_port>`

Remplacez <proxy_IP> par l'adresse IP du proxy Zabbix Cloud et <proxy_listen_port> par le port d'écoute configuré lors de la création du nœud proxy. Le port d'écoute par défaut est 10051.

Si un proxy est spécifié dans ServerActive, l'agent obtient sa configuration de vérification active auprès du proxy plutôt que directement auprès du serveur Zabbix.

Assurez-vous que l'hôte supervisé peut établir une connexion TCP sortante vers le proxy sur le port d'écoute configuré.

Pour plus d'informations sur les vérifications passives et actives de l'agent, consultez Vérifications passives et actives de l'agent.

Serveur Zabbix externe Un proxy Zabbix Cloud peut également communiquer avec un serveur Zabbix déployé en dehors de Zabbix Cloud.

Dans ce cas, le serveur Zabbix externe doit pouvoir communiquer avec le proxy cloud selon le mode de proxy sélectionné.

Pour un proxy en mode actif, le proxy initie la connexion au serveur Zabbix externe. Configurez le serveur Zabbix externe pour accepter les connexions provenant du proxy et assurez-vous que le proxy peut atteindre le serveur sur le réseau.

Pour un proxy passif, le serveur Zabbix initie la connexion au proxy. Ajoutez l'adresse IP ou le sous-réseau du serveur Zabbix externe aux filtres d'accès du nœud de proxy et assurez-vous que le serveur peut atteindre le proxy sur le réseau.

Le serveur Zabbix externe n'utilise pas les filtres d'accès de Zabbix Cloud. Configurez plutôt les règles de pare-feu et de réseau requises dans l'environnement externe.

Les hôtes surveillés peuvent se trouver dans le même environnement que le serveur Zabbix externe, sur un autre réseau ou sur un réseau accessible par le proxy.

Lorsque les agents utilisent des vérifications actives, configurez leur paramètre ServerActive pour pointer vers le proxy Zabbix Cloud. Lorsque les agents utilisent des vérifications passives, configurez leur paramètre Server pour autoriser les connexions provenant du proxy.

Vérifier la configuration Après avoir configuré le proxy, les agents, les filtres d'accès et la connectivité réseau, vérifiez que le proxy communique avec le serveur Zabbix.

1. Accédez à *Administration* > *Proxies*.
2. Vérifiez l'*État* du proxy.
3. Accédez à *Monitoring* > *Latest data* et vérifiez que des valeurs récentes des éléments sont disponibles.

Si le proxy ne communique pas avec le serveur, vérifiez :

- le mode du proxy ;
- l'adresse et le port du serveur ;
- les filtres d'accès du proxy ;
- les filtres d'accès du serveur, lorsque le serveur est hébergé sur Zabbix Cloud ;
- la configuration de l'agent ;
- les règles de pare-feu et de routage ;
- l'adresse configurée dans Server ou ServerActive.

Application mobile

Application mobile

L'application mobile Zabbix vous permet de surveiller votre environnement Zabbix depuis votre appareil mobile. Consultez les problèmes, vérifiez leurs détails, mettez-les à jour et recevez des notifications push lorsqu'une intervention est nécessaire.

Obtenir l'application (bientôt disponible)

Configurer Zabbix pour l'application

Configurer l'adaptateur bridge Zabbix Installez l'adaptateur bridge Zabbix pour permettre aux appareils mobiles de se connecter à Zabbix.

Ajouter des appareils Ajoutez votre appareil mobile dans l'interface web de Zabbix pour le connecter à votre serveur Zabbix.

Configurer les notifications push Recevez des alertes sur votre appareil mobile lorsqu'un problème survient.

Découvrir l'application

Présentation de l'application Découvrez les bases de l'utilisation de l'application mobile Zabbix.

Accès mobile sécurisé Sécurisez l'application, vos appareils mobiles et l'adaptateur Zabbix bridge, ou révoquez l'accès des appareils perdus ou compromis.

Configurer Zabbix pour l'application

Pour utiliser l'application mobile Zabbix (bientôt disponible), vous devez d'abord :

1. Configurer l'adaptateur bridge Zabbix.
2. Configurer les notifications push.
3. Ajouter votre appareil mobile dans l'interface web Zabbix.

Les sections ci-dessous couvrent chaque étape.

Après la configuration, consultez Sécuriser l'accès mobile pour sécuriser l'application et l'adaptateur bridge Zabbix.

Configurer l'adaptateur bridge Zabbix Pour permettre à votre serveur Zabbix de communiquer avec l'application mobile, vous devez configurer l'adaptateur bridge Zabbix.

L'adaptateur bridge peut être installé sur le même système que votre serveur Zabbix ou sur un autre système, à l'aide de paquets :

> **Attention:**
> Les étapes ci-dessous concernent Ubuntu 24.04. Pour les autres systèmes d'exploitation, utilisez les commandes d'installation du dépôt et des paquets (par exemple, `wget`, `dpkg`, `apt`) indiquées sur la page Obtenir Zabbix correspondant à votre système d'exploitation, puis utilisez-les pour installer le paquet `zabbix-bridge-adapter`.

1. Démarrez une nouvelle session shell avec les privilèges root :

```
sudo -s
```

2. Installez le dépôt Zabbix :

```
wget https://repo.zabbix.com/zabbix/8.0/release/ubuntu/pool/main/z/zabbix-release/zabbix-release_latest_8.
dpkg -i zabbix-release_latest_8.0+ubuntu24.04_all.deb
apt update
```

3. Installez l'adaptateur bridge Zabbix :

```
apt install zabbix-bridge-adapter
```

4. Modifiez le fichier de configuration du serveur Zabbix :

```
EnableMobileDevices=1
BridgeAdapterURL=https://bridge-adapter.example.com:10054/rpc
```

5. Redémarrez le serveur Zabbix, par exemple (voir les autres options) :

```
systemctl restart zabbix-server
```

6. Copiez l'identifiant du serveur Zabbix depuis *Rapports* > *Informations système* ; vous l'utiliserez à l'étape suivante.

7. Modifiez le fichier de configuration de l'adaptateur bridge Zabbix :

```
ZabbixAPIURL=https://example.com/zabbix/api_jsonrpc.php
ServerId=019fb1f9-22bd-74ea-bc03-51efb8b76aba
DataDir=/var/lib/zabbix-bridge-adapter
```

> **Attention:**
> Le répertoire spécifié dans `DataDir` doit disposer des permissions 0700, et tous les fichiers qui y sont stockés doivent disposer des permissions 0600.

8. Consultez et acceptez le contrat de licence utilisateur final (EULA) de l'adaptateur bridge Zabbix :

```
zabbix_bridge_adapter license show
sudo -u zabbix-bridge-adapter zabbix_bridge_adapter -c /etc/zabbix/zabbix_bridge_adapter.conf license acce
```

9. Démarrez l'adaptateur bridge Zabbix :

```
systemctl start zabbix-bridge-adapter
```

> **Attention:**
> Lors de son démarrage, l'adaptateur bridge Zabbix utilise l'identifiant du serveur pour inscrire votre serveur Zabbix auprès du Zabbix Bridge hébergé de manière centralisée. L'état de l'inscription est stocké dans un fichier d'état situé dans `DataDir`. Si vous réinstallez l'adaptateur bridge et que le fichier d'état est perdu, l'inscription du même serveur Zabbix peut échouer. Pour éviter cela, sauvegardez le fichier d'état avant de réinstaller l'adaptateur bridge, puis restaurez-le ensuite.

Configurer les notifications push Pour recevoir des alertes sur votre appareil mobile lorsqu'une intervention est nécessaire dans votre environnement surveillé :

1. Créez un type de média de notification push (ou activez celui par défaut).
2. Créez une action (ou activez l'action de déclencheur par défaut *Signaler les problèmes aux administrateurs Zabbix*).
3. Ajoutez un appareil pour ajouter automatiquement le type de média à votre utilisateur (ou ajoutez manuellement un média utilisateur).

Ajouter des appareils Pour utiliser l'application mobile Zabbix sur votre appareil, vous devez ajouter cet appareil dans l'interface web Zabbix. Pour obtenir des instructions, consultez :

- *Utilisateurs* > *Appareils* (en tant que *Super administrateur*, ajoutez vos propres appareils mobiles ou ceux d'un autre utilisateur)
- *Paramètres utilisateur* > *Appareils* (en tant qu'utilisateur, ajoutez vos propres appareils mobiles)

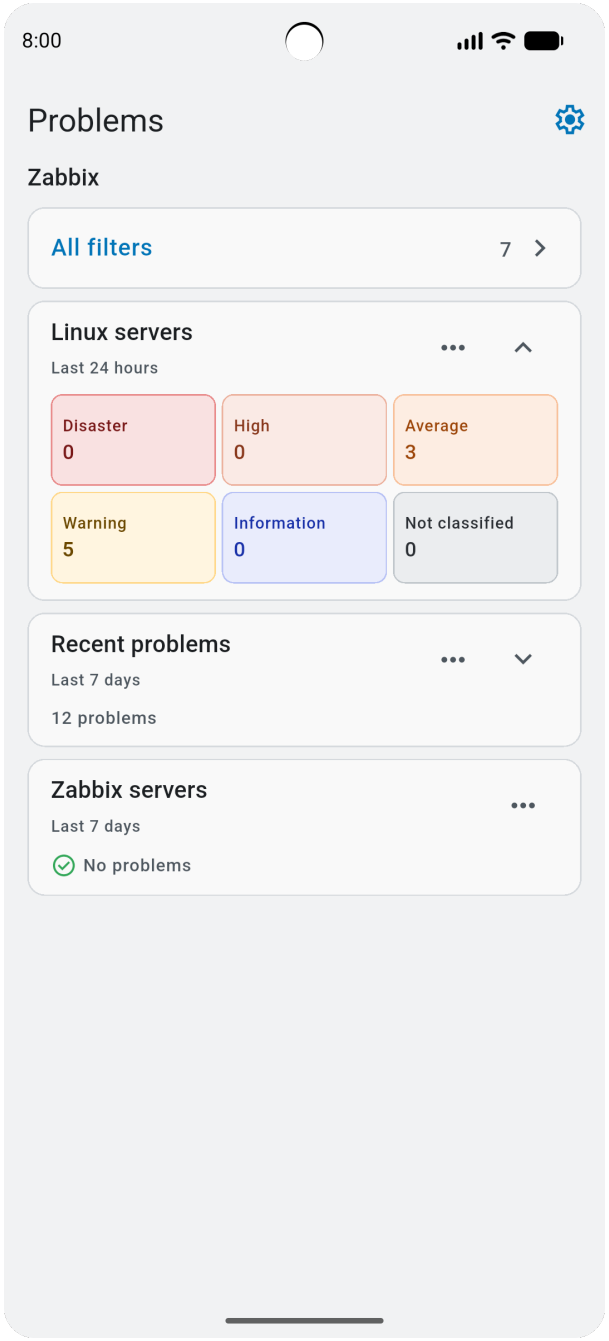
Présentation de l'application

L'application mobile Zabbix (bientôt disponible) vous permet de surveiller votre environnement Zabbix depuis votre appareil mobile. Consultez les problèmes, vérifiez leurs détails, mettez-les à jour et recevez des notifications push lorsqu'une intervention est nécessaire.

Cette page vous présente l'application mobile et explique comment l'utiliser.

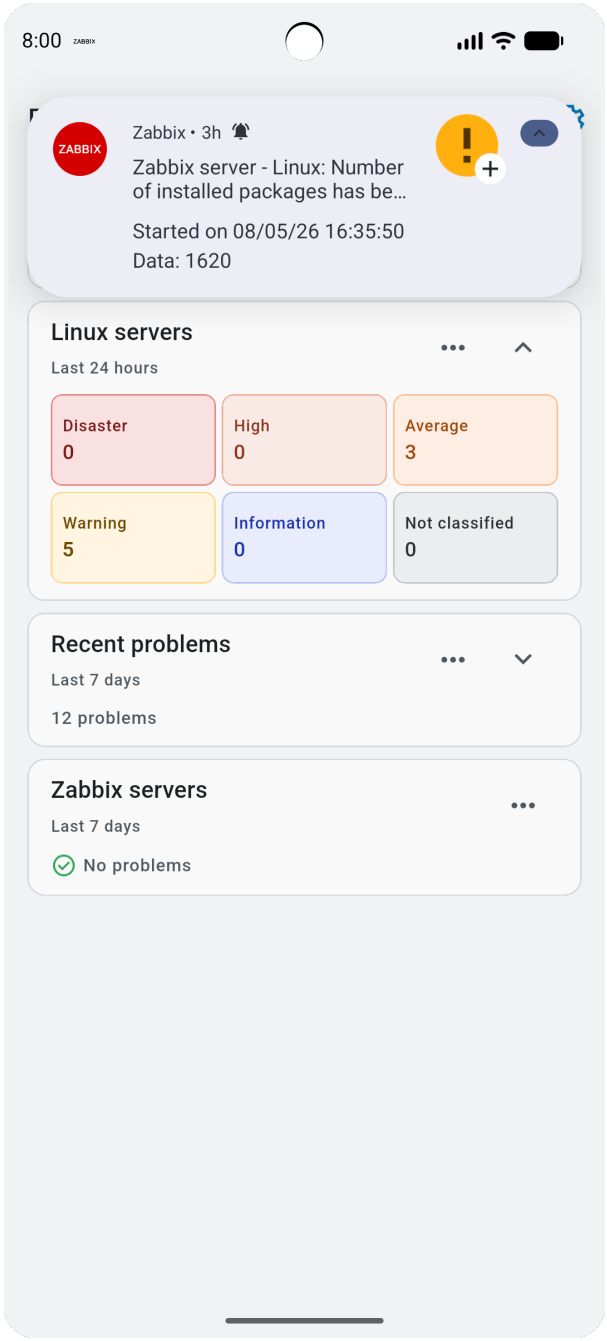
Problèmes L'écran *Problèmes* affiche une vue d'ensemble des problèmes en fonction de vos filtres épinglés. Il contient :

- Icône *Paramètres* - appuyez dessus pour afficher les paramètres de l'application.
- *Tous les filtres* - appuyez dessus pour afficher et gérer les filtres.
- Nom du serveur Zabbix (ou *Zabbix*, s'il n'a pas été configuré lors de l'installation de l'interface web Zabbix).
- Filtres épinglés - appuyez sur leur nom ou sur une gravité pour ouvrir l'écran *Liste des problèmes*. Appuyez sur le menu à trois points pour gérer le filtre (le modifier, le dupliquer, le désépingler ou le supprimer), ou appuyez sur la flèche pour le développer ou le réduire. Notez que les filtres épinglés sont classés par ordre alphabétique et que le filtre *Problèmes récents* ne peut pas être modifié ni supprimé.



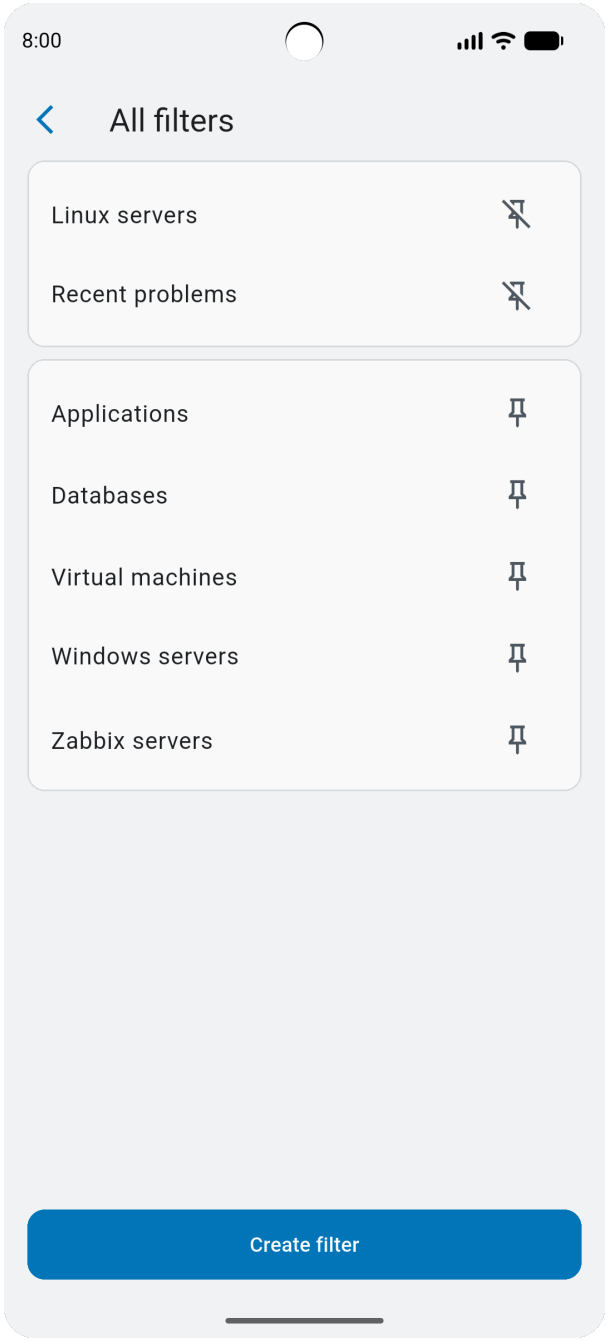
Notifications Lorsque le serveur Zabbix détecte un problème, ou lorsqu'un problème est mis à jour ou résolu, il peut envoyer une notification à votre appareil mobile. La notification contient le hôte concerné par le problème, son nom et un message généré à partir des modèles de messages configurés pour le type de média notification push.

Appuyez sur la notification pour ouvrir l'écran *Détails du problème*.



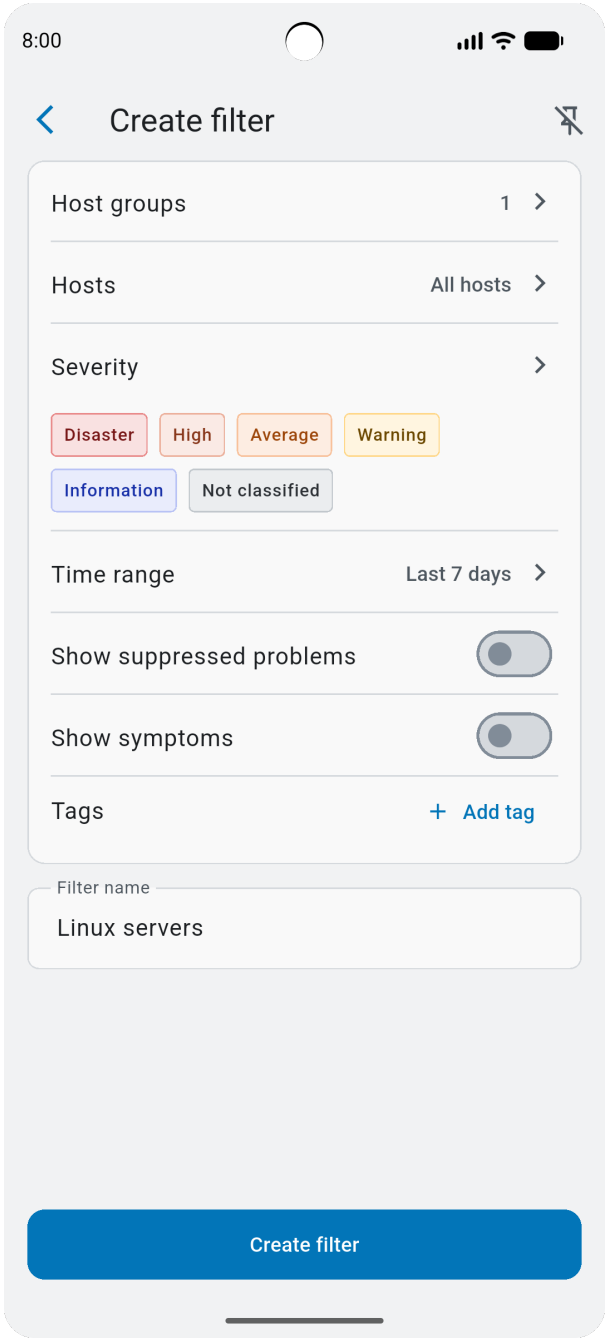
Tous les filtres L'écran *Tous les filtres* vous permet de gérer les filtres. Il contient :

- Noms des filtres : appuyez dessus pour modifier le filtre.
- Icônes d'épingle : appuyez dessus pour épingler ou désépingler le filtre dans l'écran *Problèmes*.
- *Créer un filtre* : appuyez dessus pour créer un filtre.



Créer un filtre L'écran *Créer un filtre* vous permet de créer un filtre, avec des options similaires à celles de *Supervision* > *Problèmes* (voir les détails). Vous pouvez également définir une *Plage horaire* pour indiquer pendant combien de temps afficher les problèmes non résolus.

Appuyez sur *Créer un filtre* pour enregistrer le filtre. Les nouveaux filtres sont épinglés à l'écran *Problèmes* ; appuyez sur l'icône représentant une épingle pour modifier ce comportement.



Liste des problèmes L'écran *Liste des problèmes* affiche les problèmes correspondant au filtre (par exemple, *Problèmes récents*).

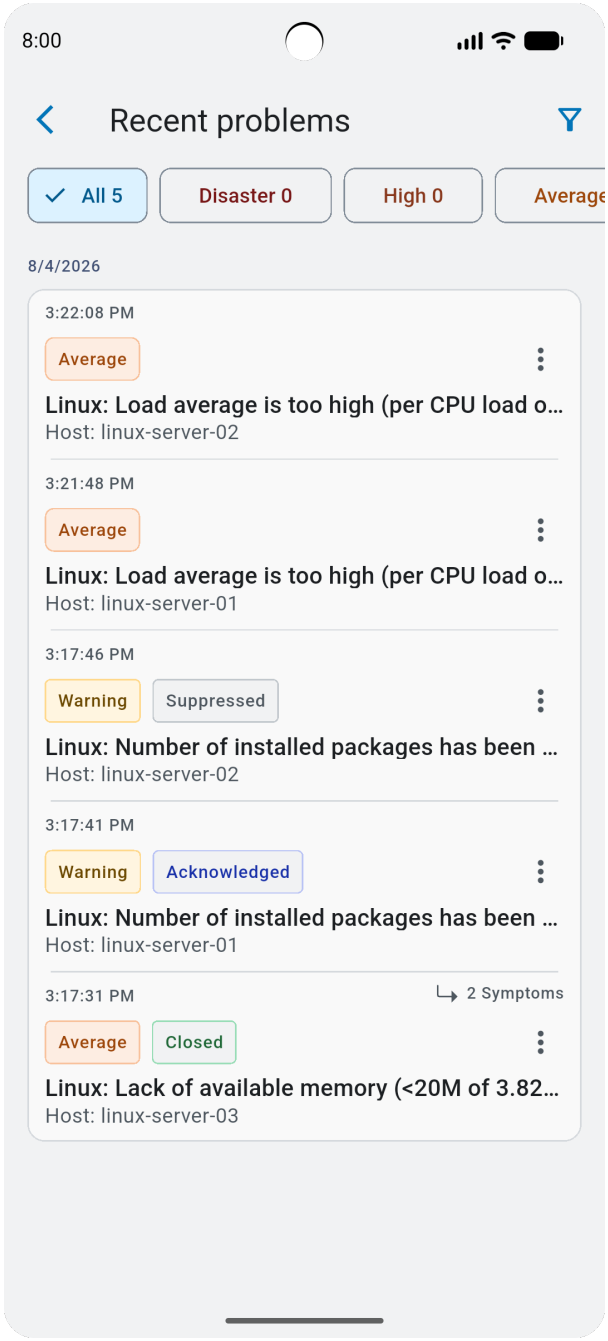
Appuyez sur l'icône de filtre ou sur une gravité de problème pour affiner davantage la liste.

Pour chaque problème, vous pouvez voir :

- L'heure et la date de début.
- La gravité, le nom et l'hôte.
- S'il a été supprimé (affiché si *Afficher les problèmes supprimés* est activé).
- S'il a été reconnu ou récemment fermé (résolu automatiquement ou fermé manuellement).
- S'il comporte des symptômes - appuyez dessus pour afficher la liste des problèmes symptomatiques.

Appuyez sur un problème pour ouvrir l'écran *Détails du problème*.

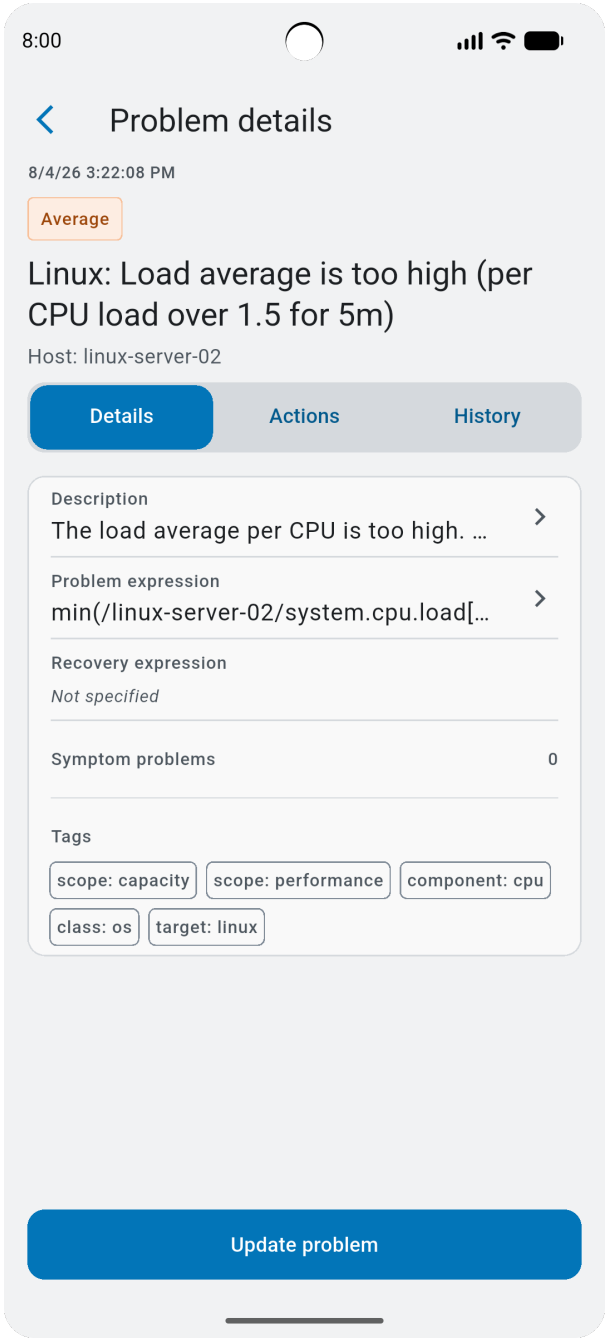
Appuyez sur le menu à trois points pour reconnaître rapidement le problème ou le supprimer pendant la durée de votre choix.



Détails du problème L'écran *Détails du problème* affiche les détails, les actions et l'historique du problème.

- Onglet *Détails* - affiche la description du problème, les expressions, les problèmes symptômes (appuyez dessus pour afficher la liste des problèmes) et les étiquettes.
- Onglet *Actions* - affiche la chronologie du problème : quand il a commencé ou a été reconnu, si une notification a été envoyée et les autres actions.
- Onglet *Historique* - affiche l'ancienneté, la durée et l'heure de récupération du problème (ou *Non disponible* si le problème est toujours en cours).

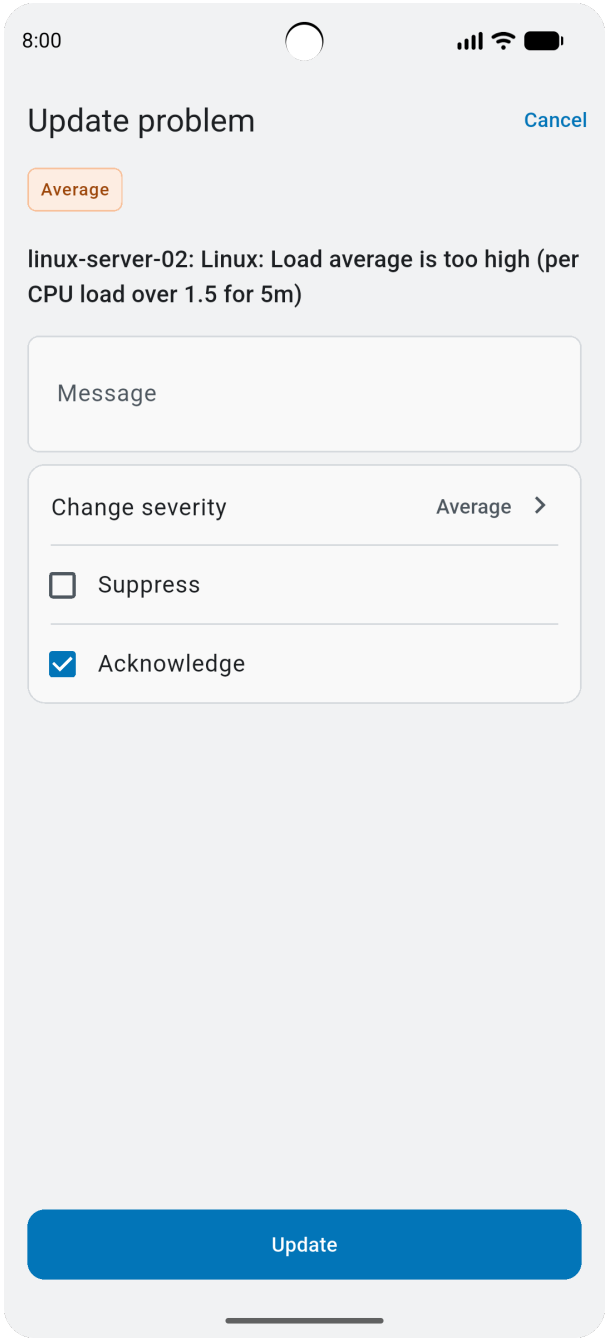
Appuyez sur *Mettre à jour le problème* pour mettre à jour le problème.



Mettre à jour le problème L'écran *Mettre à jour le problème* vous permet de mettre à jour un problème :

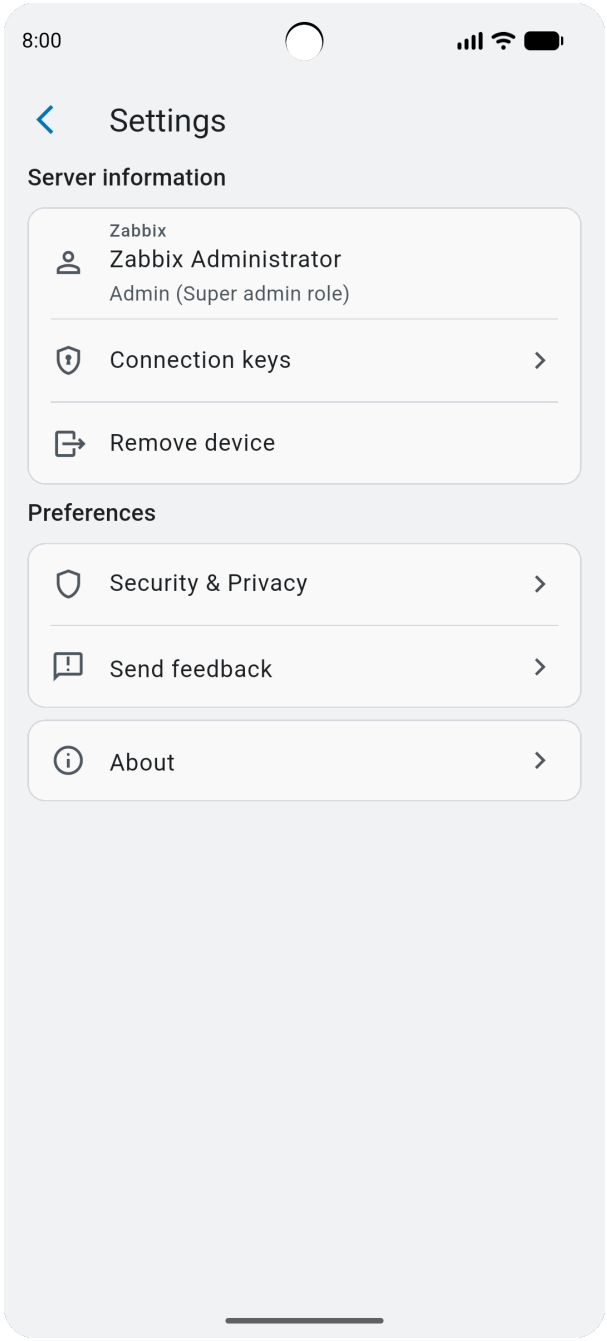
- Ajouter un message.
- Modifier la gravité du problème.
- Supprimer temporairement le problème pendant la durée de votre choix.
- Acquitter ou annuler l'acquittement du problème.
- Fermer le problème.

Appuyez sur *Mettre à jour* pour enregistrer vos modifications.



Paramètres L'écran *Paramètres* vous permet d'accéder aux paramètres de l'application. Il contient :

- Le nom du serveur Zabbix (ou *Zabbix*, s'il n'a pas été configuré lors de l'installation de l'interface web Zabbix).
- Le nom, le type et le rôle de l'utilisateur auquel l'appareil mobile est ajouté.
- *Clés de connexion* - ouvrez cette section pour afficher et copier l'ID du serveur, l'ID de l'appareil et d'autres informations.
- *Supprimer l'appareil* - appuyez sur cette option pour supprimer l'appareil mobile de Zabbix.
- *Sécurité et confidentialité* - ouvrez cette section pour activer le verrouillage de l'application et gérer le partage des données de diagnostic.
- *Envoyer des commentaires* - ouvrez cette section pour envoyer des commentaires à Zabbix.
- *À propos* - ouvrez cette section pour afficher la version de l'application et d'autres informations la concernant.



Accès mobile sécurisé

Cette page présente les bonnes pratiques pour sécuriser l'application mobile Zabbix et l'adaptateur Zabbix bridge.

Protéger les appareils mobiles Pour empêcher d'autres personnes d'ouvrir l'application sur votre appareil mobile, activez le verrouillage de l'application dans *Paramètres*.

Si un appareil a été perdu ou compromis, supprimez-le dans l'interface web de Zabbix. Cela révoque et supprime ses identifiants d'accès, afin que l'appareil ne puisse plus accéder à Zabbix. Selon les permissions associées à votre rôle utilisateur, vous pouvez supprimer des appareils dans les sections suivantes :

- *Paramètres utilisateur > Appareils* - tout utilisateur peut supprimer ses propres appareils (la permission *Gérer ses propres appareils* est requise).
- *Utilisateurs > Appareils* - un *Super administrateur* peut supprimer les appareils de n'importe quel utilisateur (la permission *Gérer les appareils des utilisateurs* est requise).

Vous pouvez également limiter les parties de Zabbix accessibles depuis un appareil mobile. Pour ce faire, attribuez aux utilisateurs un rôle utilisateur qui accorde uniquement les permissions requises par l'application. Les permissions minimales sont les suivantes :

Section	Configuration
Accès aux éléments de l'interface	Décochez toutes les cases.
Accès aux services	Définissez *Accès en lecture-écriture aux services* et *Accès en lecture seule aux services* sur `None`.
Accès aux modules	Décochez toutes les cases.
Accès à l'API	Décochez la case *Activé*.
Accès aux actions	Décochez toutes les cases. Pour permettre aux utilisateurs de mettre à jour les problèmes dans l'application, cochez également les actions dont ils ont besoin : - *Ajouter des commentaires aux problèmes* - *Modifier la sévérité* - *Acquitter les problèmes* - *Supprimer les problèmes* - *Fermer les problèmes* - *Créer et modifier ses propres médias* Notez que les problèmes que les utilisateurs peuvent voir dans l'application dépendent également de leur accès aux hôtes.
Appareils	Cochez la case *Activé*. Pour permettre aux utilisateurs d'ajouter et de supprimer leurs propres appareils mobiles, cochez également la case *Gérer ses propres appareils* ; sinon, un *Super administrateur* doit gérer leurs appareils dans *Utilisateurs > Appareils*.

Sécuriser l'adaptateur bridge Zabbix L'adaptateur bridge Zabbix stocke ses données (fichier d'état et clés de chiffrement) dans le répertoire spécifié par le paramètre `DataDir` du fichier de configuration de l'adaptateur bridge Zabbix. Assurez-vous que le répertoire dispose des permissions 0700 et que tous les fichiers qui y sont stockés disposent des permissions 0600.

L'adaptateur bridge Zabbix transmet les requêtes des appareils mobiles à l'API Zabbix. Pour limiter les méthodes d'API pouvant être appelées via l'adaptateur bridge Zabbix, utilisez son paramètre `AllowedAPIMethods`. Par exemple, autorisez uniquement l'ensemble minimal de méthodes nécessaires à l'application mobile, voire un ensemble encore plus restreint si vous ne souhaitez pas que certaines fonctionnalités de l'application mobile soient utilisées (par exemple, n'autorisez pas `event.acknowledge` si les utilisateurs de l'application mobile ne doivent pas mettre à jour les problèmes).

Centre de développement

Centre de développement

Commencez à développer des extensions personnalisées pour Zabbix grâce à des conseils sur les plugins, les modules et les widgets.

Développer sur Zabbix

Plugins Un aperçu de la manière de développer et de gérer des plugins Zabbix afin d'étendre les fonctionnalités ou de s'intégrer à des systèmes externes.

Bibliothèque Python pour Zabbix Ressources pour utiliser la bibliothèque officielle `zabbix_utils` afin d'interagir avec l'API Zabbix et de créer des intégrations.

Modules frontend Guides et références pour créer des modules personnalisés qui étendent ou modifient le frontend Zabbix afin de répondre à des cas d'utilisation spécifiques.

Widgets Une présentation de la structure et de la logique des widgets, avec des instructions pour créer des éléments de tableau de bord personnalisés adaptés à vos besoins.

Création d'un plugin

Cette page s'adresse aux personnes qui souhaitent développer leur propre plugin (collecteur de données indépendant) pour l'agent Zabbix 2.

Un plugin personnalisé étend les fonctionnalités de l'agent Zabbix 2 au-delà des plugins intégrés et des plugins chargeables fournis.

Chaque plugin est un package *Go* qui définit la structure et implémente une ou plusieurs interfaces (*Exporter*, *Configurator*, *Runner*). Pour plus d'informations, consultez Interfaces des plugins et Diagramme de connexion.

Ce guide vous aidera à créer un plugin chargeable personnalisé.

Pour obtenir des conseils supplémentaires, consultez également le dépôt des éléments suivants :

- Plugin d'exemple
- Plugins chargeables existants

Ce que vous allez créer Il s'agit d'un tutoriel étape par étape pour créer un plugin chargeable simple nommé **MyIP**. Ce plugin implémentera une seule clé d'élément (`myip`) renvoyant l'adresse IP externe de l'hôte de l'agent Zabbix.

Étape 1 : Configuration

1. Un plugin est un module Go standard. Commencez par initialiser le fichier `go.mod` dans le répertoire du plugin afin de suivre les dépendances du plugin :

```
cd path/to/plugins/myip # Basculez vers votre répertoire de plugin
go mod init myip
```

2. Installez la dépendance obligatoire Zabbix Go SDK (`golang.zabbix.com/sdk`) :

```
go get golang.zabbix.com/sdk@$LATEST_COMMIT_HASH
```

Remplacez `$LATEST_COMMIT_HASH` par le hachage du commit `HEAD` le plus récent du dépôt `golang.zabbix.com/sdk` dans la branche de version appropriée. Par exemple :

```
go get golang.zabbix.com/sdk@af85407
```

Notez que le versionnement de `golang.zabbix.com/sdk` n'est actuellement pas pris en charge, mais cela pourrait changer à l'avenir.

Des dépendances supplémentaires peuvent être installées selon les besoins à l'aide de `go get`.

3. Créez un fichier `main.go` vide pour le code source du plugin :

```
touch main.go
```

La configuration initiale est maintenant terminée, et le plugin est prêt pour le développement.

Étape 2 : Structure du plugin Le module `golang.zabbix.com/sdk`, installé à l'étape précédente, fournit le framework nécessaire au développement des plugins et garantit que tous les plugins ont une structure cohérente.

1. Configurez le flux d'exécution de base.

Commencez par définir le flux d'exécution principal du plugin. Ajoutez le code suivant à `main.go` :

```
package main

func main() {
    err := run()
    if err != nil {
        panic(err)
    }
}

func run() error {
    return nil
}
```

Cela établit le flux d'exécution de base du plugin. La fonction run contiendra ultérieurement la logique principale du plugin.

2. Explorez les interfaces du plugin.

Un plugin Zabbix agent 2 doit être représenté par une structure qui implémente les interfaces du package `golang.zabbix.com/sdk/plugin` :

- *Accessor* - définit les méthodes essentielles que tous les plugins doivent implémenter, telles que la définition du nom du plugin et la gestion des délais d'expiration des clés d'élément.
- Une ou plusieurs des interfaces fonctionnelles suivantes :
  - *Exporter* - effectue une interrogation et renvoie une valeur (ou plusieurs valeurs), rien ou une erreur ; souvent utilisé avec l'interface *Collector*.
  - *Collector* - gère la collecte périodique des données.
  - *Runner* - définit les procédures de démarrage et d'arrêt du plugin.
  - *Watcher* - permet l'interrogation indépendante des métriques, en contournant le planificateur interne de l'agent ; utile pour la supervision basée sur des traps ou pilotée par les événements.
  - *Configurator* - définit la manière dont le plugin lit et applique ses paramètres de configuration.

Vous pouvez implémenter vous-même ces interfaces ou utiliser celles fournies par défaut par le SDK Go de Zabbix, en les modifiant selon vos besoins. Ce tutoriel utilise les implémentations par défaut.

3. Créez la structure du plugin.

Importez maintenant le package *plugin* et créez une structure `myIP` qui intègre la structure `plugin.Base` :

```
import "golang.zabbix.com/sdk/plugin"

type myIP struct {
    plugin.Base
}
```

La structure myIP satisfait actuellement l'interface Accessor. Une méthode permettant d'implémenter l'une des interfaces fonctionnelles du plugin, *Exporter*, sera ajoutée ultérieurement dans ce tutoriel.

Étape 3 : Définir les clés d'élément Votre plugin a besoin des clés d'élément pour collecter les données et les fournir au serveur Zabbix ou au proxy.

1. Importez le package *errs* pour la gestion des erreurs :

```
import "golang.zabbix.com/sdk/errs"
```

2. Enregistrez les clés d'élément à l'aide de la fonction `plugin.RegisterMetrics()` dans la fonction `run()` :

```
func run() error {
    p := &myIP{}

    // Register the `myip` item key.
    err := plugin.RegisterMetrics(
        p,
        "MyIP",                          // Plugin name
        "myip",                          // Item key name
        "Returns the host's IP address.", // Item key description
    )
    if err != nil {
        return errs.Wrap(err, "failed to register metrics")
    }

    return nil
}
```

Pour enregistrer plusieurs clés d'élément, répétez les paramètres *nom de métrique* et *description* pour chaque métrique.
Par exemple :

```
plugin.RegisterMetrics(&impl, "Myip", "metric.one", "Metric one description.", "metric.two", "Metric two d
```

Étape 4 : Configurer le handler Le handler facilite la communication entre l'agent et le plugin.

1. Importez le package *container* :

```
import "golang.zabbix.com/sdk/plugin/container"
```

2. Dans la fonction `run()`, ajoutez le code pour créer et configurer un handler :

```
func run() error {
    p := &myIP{}
```

```
    // Register the `myip` item key.
    err := plugin.RegisterMetrics(
        p,
        "MyIP",                          // Plugin name
        "myip",                          // Item key name
        "Returns the host's IP address.", // Item key description
    )
    if err != nil {
        return errs.Wrap(err, "failed to register metrics")
    }

    // Create a new handler.
    h, err := container.NewHandler("MyIP") // Plugin name
    if err != nil {
        return errs.Wrap(err, "failed to create new handler")
    }

    // Setup logging to forward logs from the plugin to the agent.
    // Available via p.Logger.Infof, p.Logger.Debugf, etc.
    p.Logger = h

    // Start plugin execution.
    // Blocks until a termination request is received from the agent.
    err = h.Execute()
    if err != nil {
        return errs.Wrap(err, "failed to execute plugin handler")
    }

    return nil
}
```

Étape 5 : Implémenter la collecte de données La collecte de données s'effectue via l'interface Exporter, qui décrit la méthode Export :

```
func Export(
  key string,             // La clé de l'élément à collecter.
  params []string,        // Arguments transmis à la clé de l'élément (`myip[arg1, arg2]`).
  context ContextProvider // Métadonnées sur la collecte de données de la clé de l'élément.
) (any, error)
```

1. Importez les packages requis pour les requêtes HTTP et la lecture des réponses :

```
import (
    "io"
    "net/http"
)
```

2. Implémentez la méthode `Export` pour la structure `myIP` :

```
func (p *myIP) Export(
    key string, params []string, context plugin.ContextProvider,
) (any, error) {
    // Le plugin peut utiliser une logique de collecte de données différente selon le paramètre `key`.
    // Cette implémentation vérifie uniquement que la valeur `key` fournie est prise en charge.
    if key != "myip" {
        return nil, errs.Errorf("unknown item key %q", key)
    }

    // Le journal sera transféré vers le journal de l'agent 2.
    p.Infof(
        "received request to handle %q key with %d parameters",
        key,
        len(params),
    )
```

```
    // Collectez les données et renvoyez-les.

    resp, err := http.Get("https://api.ipify.org")
    if err != nil {
        return nil, errs.Wrap(err, "failed to get IP address")
    }

    defer resp.Body.Close()

    body, err := io.ReadAll(resp.Body)
    if err != nil {
        return nil, errs.Wrap(err, "failed to read response body")
    }

    return string(body), nil
}
```

Étape 6: Compiler et configurer le plugin

1. Pour compiler le plugin, exécutez :

```
go mod tidy
go build
```

Cela devrait créer un exécutable `myip` dans le répertoire courant.

2. Configurez agent Zabbix 2 pour utiliser le plugin :

```
echo "Plugins.MyIP.System.Path=$PATH_TO_THE_MYIP_PLUGIN_EXECUTABLE" > /etc/zabbix/zabbix_agent2.d/plugins.
```

Remplacez `$PATH_TO_THE_MYIP_PLUGIN_EXECUTABLE` par le chemin vers le `myip` créé à l'étape 5.

Le nom du plugin dans le nom du paramètre de configuration (*MyIP* dans ce tutoriel) doit correspondre au nom du plugin défini dans la fonction *plugin.RegisterMetrics()*.

3. Pour tester le plugin et son élément `myip`, exécutez :

```
zabbix_agent2 -c /etc/zabbix/zabbix_agent2.conf -t myip
```

La sortie doit contenir une adresse IP externe de votre hôte et ressembler à ceci :

```
myip                                          [s|192.0.2.1]
```

Ainsi, vous avez créé un plugin chargeable simple pour agent Zabbix 2. Félicitations !

```
package main

import (
    "io"
    "net/http"

    "golang.zabbix.com/sdk/errs"
    "golang.zabbix.com/sdk/plugin"
    "golang.zabbix.com/sdk/plugin/container"
)

var _ plugin.Exporter = (*myIP)(nil)

type myIP struct {
    plugin.Base
}

func main() {
    err := run()
    if err != nil {
        panic(err)
```

```
    }
}

func run() error {
    p := &myIP{}

    // Enregistrer la clé d'élément `myip`.
    err := plugin.RegisterMetrics(
        p,
        "MyIP",                          // Nom du plugin
        "myip",                          // Nom de la clé d'élément
        "Returns the host's IP address.", // Description de la clé d'élément
    )
    if err != nil {
        return errs.Wrap(err, "failed to register metrics")
    }

    // Créer un nouveau gestionnaire.
    h, err := container.NewHandler("MyIP") // Nom du plugin
    if err != nil {
        return errs.Wrap(err, "failed to create new handler")
    }

    // Configurer la journalisation pour transférer les journaux du plugin vers l'agent.
    // Disponible via p.Logger.Infof, p.Logger.Debugf, etc.
    p.Logger = h

    // Démarrer l'exécution du plugin.
    // Bloque jusqu'à la réception d'une demande d'arrêt de l'agent.
    err = h.Execute()
    if err != nil {
        return errs.Wrap(err, "failed to execute plugin handler")
    }

    return nil
}

func (p *myIP) Export(
    key string, params []string, context plugin.ContextProvider,
) (any, error) {
    // Le plugin peut utiliser une logique de collecte de données différente selon le paramètre `key`.
    // Cette implémentation vérifie uniquement que la valeur `key` fournie est prise en charge.
    if key != "myip" {
        return nil, errs.Errorf("unknown item key %q", key)
    }

    // Le journal sera transféré vers le journal 2 de l'agent.
    p.Infof(
        "received request to handle %q key with %d parameters",
        key,
        len(params),
    )

    // Collecter les données et les renvoyer.

    resp, err := http.Get("https://api.ipify.org")
    if err != nil {
        return nil, errs.Wrap(err, "failed to get IP address")
    }

    defer resp.Body.Close()
```

```
    body, err := io.ReadAll(resp.Body)
    if err != nil {
        return nil, errs.Wrap(err, "failed to read response body")
    }

    return string(body), nil
}
```

Code source complet

Interfaces de plugins

Cette section décrit les interfaces de plugin disponibles.

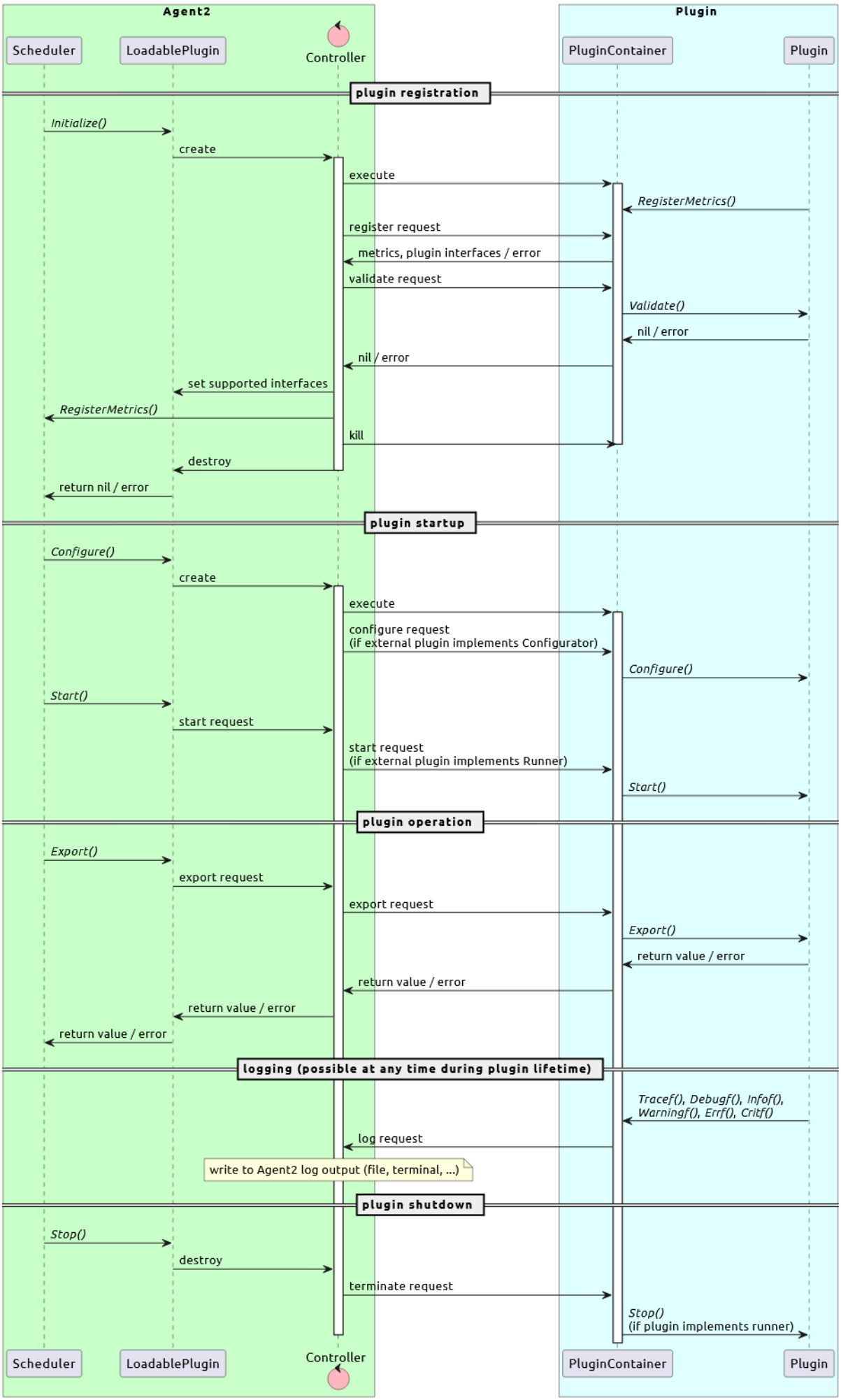
plugin.Exporter *Exporter* est l'interface la plus simple qui effectue un sondage et renvoie une ou plusieurs valeurs, rien ou une erreur. Il accepte une clé d'élément préparé, des paramètres et un contexte préparés. L'accès à toutes les autres interfaces du plugin est exclusif et aucune méthode ne peut être appelée si un plugin exécute déjà une tâche. En outre, il existe une limite de 100 appels *Export()* simultanés maximum par plugin, qui peut être réduite en fonction des exigences de chaque plugin.

plugin.Configurator L'interface *Configurator* fournit les paramètres de configuration du plugin à partir des fichiers de configuration de l'agent Zabbix 2.

plugin.Runner L'interface *Runner* fournit les moyens d'effectuer l'initialisation lorsqu'un plugin est démarré (activé) et la désinitialisation lorsqu'un plugin est arrêté (désactivé). Par exemple, un plugin peut démarrer/arrêter une *goroutine* en arrière-plan en implémentant l'interface Runner.

Schéma de connexion Zabbix agent 2 se connecte de manière bidirectionnelle aux plugins à l'aide de sockets UNIX sous Linux et de Named Pipes sous Windows.

Le schéma de connexion ci-dessous illustre le processus de communication entre Zabbix agent 2 et un plugin chargeable, ainsi que le processus de collecte des métriques.


Agent2
Plugin
Scheduler
LoadablePlugin
Controller
PluginContainer
Plugin
plugin registration
Initialize()
create
execute
RegisterMetrics()
register request
metrics, plugin interfaces / error
validate request
Validate()
nil / error
nil / error
set supported interfaces
RegisterMetrics()
kill
destroy
return nil / error
plugin startup
Configure()
create
execute
configure request
(if external plugin implements Configurator)
Configure()
Start()
start request
start request
(if external plugin implements Runner)
Start()
plugin operation
Export()
export request
export request
Export()
return value / error
return value / error
return value / error
return value / error
logging (possible at any time during plugin lifetime)
Tracef(), Debugf(), Infof(),
Warningf(), Errf(), Critf()
log request
write to Agent2 log output (file, terminal, ...)
plugin shutdown
Stop()
destroy
terminate request
Stop()
(if plugin implements runner)
Scheduler
LoadablePlugin
Controller
PluginContainer
Plugin

Bibliothèque Python pour Zabbix

zabbix_utils est la bibliothèque Python officielle pour Zabbix.

Avec zabbix_utils, vous pouvez :

- Utiliser l'API Zabbix pour gérer les objets Zabbix (créer des hôtes, mettre à jour des éléments, obtenir des événements, etc.).
- Collecter des données depuis l'agent Zabbix (de manière similaire à Zabbix get).
- Envoyer des données au serveur ou au proxy Zabbix (de manière similaire à Zabbix sender).

La bibliothèque prend en charge l'exécution de scripts synchrone et asynchrone, ce qui lui permet de gérer aussi bien des flux de travail simples que des opérations parallèles à grande échelle telles que la collecte massive de données, les exportations en masse, le traitement des alertes et une mise à l'échelle efficace du système.

La bibliothèque est compatible avec les versions prises en charge de Zabbix server et les versions modernes de Python.

Pour commencer, installez la bibliothèque, puis suivez le guide de démarrage rapide.

Installation

zabbix_utils peut être installé à l'aide de l'une des méthodes décrites ci-dessous.

Prérequis :

- Zabbix 6.0 ou version ultérieure (testé sur 6.0, 7.0, 7.2, 7.4)
- Python 3.8 ou version ultérieure (testé sur 3.8–3.13)
- Framework HTTP asynchrone aiohttp (requis uniquement pour le mode asynchrone)

Python Package Index (PyPI)

Il s'agit de la méthode la plus courante pour la plupart des environnements :

```
pip install zabbix_utils
pip install zabbix_utils[async] # Uniquement pour la prise en charge asynchrone
```

Dépôt Zabbix

Utilisez cette méthode si vous préférez gérer les dépendances via le gestionnaire de paquets de votre système.

Téléchargez les paquets Zabbix pour votre distribution Linux, puis exécutez les commandes suivantes.

Sur RHEL et ses dérivés :

```
dnf install python3-zabbix-utils
dnf install epel-release # Uniquement pour la prise en charge asynchrone
dnf install python3-aiohttp # Uniquement pour la prise en charge asynchrone
```

Sur Debian/Ubuntu et ses dérivés :

```
apt install python3-zabbix-utils
apt install python3-aiohttp # Uniquement pour la prise en charge asynchrone
```

Source (GitHub)

Utilisez cette méthode si vous préférez la dernière version de développement :

```
git clone https://github.com/zabbix/python-zabbix-utils
cd python-zabbix-utils/
pip install -r requirements.txt # Uniquement pour la prise en charge asynchrone
python3 setup.py install
```

Pour installer les dépendances requises pour le mode asynchrone, vous pouvez également utiliser l'une des méthodes décrites ci-dessus.

Guide de démarrage rapide

Après avoir installé zabbix_utils, vous pouvez l'utiliser dans votre script.

Ce guide de démarrage rapide vous montre comment :

- Récupérer le nom d'hôte depuis l'agent Zabbix.
- Créer un hôte et un élément trapper à l'aide de l'API Zabbix.
- Envoyer une valeur à l'élément.

Le guide présente le script étape par étape, en expliquant chaque partie au fur et à mesure de son introduction. Le script complet est fourni à la fin de la page.

Le guide suppose également que votre serveur Zabbix, l'agent et l'API s'exécutent localement.

Récupérer le nom d'hôte depuis l'agent Zabbix

Commencez par récupérer le nom d'hôte du système sur lequel l'agent Zabbix est en cours d'exécution. Vous aurez besoin de ce nom d'hôte pour créer un hôte à l'aide de l'API Zabbix.

1. Importez la classe `Getter` depuis zabbix_utils. Cette classe fonctionne comme Zabbix get et vous permet de demander des données à l'agent Zabbix.

2. Créez une instance de `Getter` qui se connecte à l'agent Zabbix local à l'adresse `127.0.0.1` sur le port 10050.

3. Appelez la méthode `get()` sur l'instance `Getter` pour demander le nom d'hôte à l'agent Zabbix.

- La méthode `get()` prend une clé d'élément comme paramètre et envoie une requête à l'agent Zabbix pour cet élément.
- La méthode `get()` renvoie un objet, et le nom d'hôte est stocké dans l'attribut `value` de cet objet.
- `system.hostname` est une clé intégrée de l'agent Zabbix qui renvoie le nom de l'hôte.

```
from zabbix_utils import Getter

agent = Getter(host='127.0.0.1', port=10050)
hostname = agent.get('system.hostname').value
```

Se connecter et se connecter à l'API Zabbix

Ensuite, connectez-vous à l'API Zabbix. Cela permet à votre script de gérer des objets Zabbix tels que les hôtes et les éléments.

1. Importez la classe `ZabbixAPI` depuis zabbix_utils.

2. Créez une instance de `ZabbixAPI` et indiquez l'URL de votre interface web Zabbix.

3. Appelez la méthode `login()` sur l'instance `ZabbixAPI` et fournissez votre nom d'utilisateur et votre mot de passe. Le compte doit disposer de la permission d'accéder à l'API Zabbix.

4. Appelez la méthode `logout()` pour fermer la session après les opérations de l'API.

```
from zabbix_utils import ZabbixAPI

api = ZabbixAPI(url='127.0.0.1/zabbix')
api.login(user='Admin', password='zabbix')

### API operations go here

api.logout()
```

Créer un hôte dans Zabbix

Maintenant que vous avez le nom d'hôte depuis l'agent Zabbix et que vous êtes connecté à l'API Zabbix, vous pouvez créer un nouvel hôte dans Zabbix.

1. Appelez la méthode d'API `host.create()` sur l'instance `ZabbixAPI` et fournissez les détails de l'hôte :

- `host` - définissez-le sur la variable `hostname` contenant le nom d'hôte récupéré depuis l'agent Zabbix.
- `interfaces` - contient les détails de connexion de l'agent Zabbix exécuté sur l'hôte.
- `groups` - contient au moins un groupe d'hôtes auquel l'hôte doit appartenir.

2. Puisque l'API Zabbix renvoie l'ID de l'hôte nouvellement créé, enregistrez cet ID dans la variable `host_id` pour une utilisation ultérieure.

3. Affichez un message pour confirmer que l'hôte a été créé.

```
api.host.create(
    host=hostname,
    interfaces=[{
```

```
        'type': 1,
        'main': 1,
        'useip': 1,
        'ip': '127.0.0.1',
        'dns': '',
        'port': '10050',
    }],
    groups=[{'groupid': '2'}]
)

host_id = host['hostids'][0]

print(f"Host '{hostname}' created with ID {host_id}")
```

Créer un nouvel élément dans Zabbix

Après avoir créé un hôte, vous pouvez lui ajouter un élément.

1. Définissez une clé unique pour l'élément à créer.

2. Appelez la méthode `item.create()` sur l'instance `ZabbixAPI` et fournissez les détails de l'élément :

- `hostid` - définissez-le sur la variable `host_id` contenant l'ID de l'hôte que vous venez de créer.
- `name` - un nom pour l'élément.
- `key_` - définissez-le sur la variable `item_key` que vous venez de définir.
- `type` - définissez-le sur 2 (élément trapper), requis pour recevoir les valeurs envoyées par votre script dans les étapes suivantes.
- `value_type` - définissez-le sur 3 (numérique non signé), le type de données que cet élément stocke.

3. Comme l'API Zabbix renvoie l'ID de l'élément nouvellement créé, enregistrez cet ID dans la variable `item_id` pour une utilisation ultérieure.

4. Affichez un message pour confirmer que l'élément a été créé.

```
item_key = 'app.myservice.heartbeat'

item = api.item.create(
    hostid=host_id,
    name='App heartbeat',
    key_=item_key,
    type=2,
    value_type=3,
)

item_id = item['itemids'][0]

print(f"Item '{item_key}' created with ID {item_id}")
```

Envoyer une valeur à l'hôte

Maintenant que vous avez créé votre hôte et votre élément de données, vous pouvez lui envoyer des données.

1. Importez la classe `time` et attendez quelques secondes avant d'envoyer les données. Cela garantit que Zabbix a entièrement traité votre nouvel hôte et votre nouvel élément de données.

2. Importez la classe `Sender` depuis zabbix_utils. Cette classe fonctionne comme Zabbix sender et permet à votre script d'envoyer des données à Zabbix.

3. Créez une instance de `Sender` qui se connecte au serveur Zabbix local à l'adresse `127.0.0.1` sur le port `10051`.

4. Appelez la méthode `send_value()` sur l'instance `Sender` pour envoyer une valeur à l'élément de données de votre hôte et fournissez les détails suivants :

- `hostname` - définissez-le sur la variable `hostname` contenant le nom d'hôte que vous avez récupéré depuis Zabbix agent.
- `item_key` - définissez-le sur la variable `item_key` que vous venez de définir plus tôt dans votre script.
- 1 - la valeur à envoyer.

4. Affichez un message pour confirmer que la valeur a été envoyée.

```
import time
time.sleep(10)

from zabbix_utils import Sender

sender = Sender(server='127.0.0.1', port=10051)

response = sender.send_value(hostname, item_key, 1)

print(f"Sender response: {response}")
```

Une fois ce code exécuté avec succès, vous devriez voir un message de réussite indiquant que Zabbix a reçu votre valeur :

```
Sender response: {"processed": 1, "failed": 0, "total": 1, "time": "0.000151", "chunk": 1}
```

Vous pouvez maintenant également vérifier votre interface web Zabbix (*Monitoring* > *Latest data*) pour voir la valeur.

Script complet

Ci-dessous se trouve le script complet qui combine toutes les étapes : récupération du nom d'hôte, création de l'hôte et de l'élément à l'aide de l'API Zabbix, puis envoi des données à Zabbix.

```
import time
from zabbix_utils import Getter, ZabbixAPI, Sender

### Récupérer le nom d'hôte depuis l'agent Zabbix
agent = Getter(host='127.0.0.1', port=10050)
hostname = agent.get('system.hostname').value

### Se connecter et s'authentifier à l'API Zabbix
api = ZabbixAPI(url='127.0.0.1/zabbix')
api.login(user='Admin', password='zabbix')

### Créer l'hôte dans Zabbix
host = api.host.create(
    host=hostname,
    interfaces=[{
        'type': 1,
        'main': 1,
        'useip': 1,
        'ip': '127.0.0.1',
        'dns': '',
        'port': '10050',
    }],
    groups=[{'groupid': '2'}]
)

host_id = host['hostids'][0]

print(f"Hôte '{hostname}' créé avec l'ID {host_id}")

### Créer l'élément dans Zabbix
item_key = 'app.myservice.heartbeat'

item = api.item.create(
    hostid=host_id,
    name='App heartbeat',
    key_=item_key,
    type=2,
    value_type=3
)

item_id = item['itemids'][0]

print(f"Élément '{item_key}' créé avec l'ID {item_id}")
```

```
### Se déconnecter de l'API
api.logout()

### Attendre que Zabbix traite le nouvel hôte
time.sleep(10)

### Envoyer une valeur à l'hôte
sender = Sender(server='127.0.0.1', port=10051)
response = sender.send_value(hostname, item_key, 1)

print(f"Réponse de l'expéditeur : {response}")
```

Utiliser l'API Zabbix

zabbix_utils vous permet d'utiliser l'API Zabbix pour gérer les objets Zabbix, notamment créer des hôtes, mettre à jour des éléments, récupérer des événements, et bien plus encore.

Les requêtes API peuvent être effectuées en mode synchrone ou asynchrone :

- En mode synchrone, votre script Python envoie une requête et attend une réponse avant de continuer, ce qui convient aux opérations simples, séquentielles et prévisibles.
- En mode asynchrone, le script envoie des requêtes sans attendre chaque réponse, ce qui permet à d'autres opérations de s'exécuter en parallèle ; cela est plus efficace pour les requêtes lentes ou les lots de données volumineux.

Les exemples de cette page se concentrent sur le mode synchrone, bien que le mode asynchrone suive des principes similaires. Des exemples supplémentaires sont disponibles dans le dépôt GitHub zabbix_utils.

Importation

Pour utiliser zabbix_utils afin de travailler avec l'API Zabbix, importez la classe `ZabbixAPI` dans votre script :

```
from zabbix_utils import ZabbixAPI
```

> **Note:**
> Si vous utilisez déjà une bibliothèque Python communautaire, vous pouvez généralement remplacer cet import par `ZabbixAPI` depuis zabbix_utils.

Se connecter

Avant d'effectuer des requêtes API, vous devez créer une instance `ZabbixAPI` et vous connecter à l'API Zabbix.

Choisissez la méthode qui correspond le mieux à votre façon de gérer les identifiants. Vous pouvez utiliser votre nom d'utilisateur et votre mot de passe ou des jetons d'API.

Lors de l'initialisation

Lors de la création d'une instance `ZabbixAPI`, vous pouvez fournir en une seule fois l'URL de l'interface web Zabbix ainsi que vos identifiants :

```
### Nom d'utilisateur et mot de passe :
api = ZabbixAPI(url="127.0.0.1/zabbix", user="Admin", password="zabbix")

### Jeton d'API :
api = ZabbixAPI(url="127.0.0.1/zabbix", token="your_api_token")
```

Si vous préférez regrouper les paramètres de connexion, vous pouvez les transmettre sous la forme d'un dictionnaire Python et le décompresser à l'aide de la syntaxe des arguments nommés de Python :

```
ZABBIX_AUTH = {
    "url": "127.0.0.1/zabbix",
    "user": "Admin",
    "password": "zabbix"
}

api = ZabbixAPI(**ZABBIX_AUTH)
```

Utilisation de `login()`

Vous pouvez d'abord créer une instance de `ZabbixAPI` avec uniquement l'URL de l'interface web Zabbix, puis utiliser la méthode `login()` avec vos identifiants :

```
### Nom d'utilisateur et mot de passe :
api = ZabbixAPI(url="127.0.0.1/zabbix")
api.login(user="Admin", password="zabbix")

### Jeton d'API :
api = ZabbixAPI(url="127.0.0.1/zabbix")
api.login(token="your_api_token")
```

Utilisation des variables d'environnement

Vous pouvez d'abord stocker vos identifiants comme variables d'environnement, par exemple en utilisant un terminal :

```
### Nom d'utilisateur et mot de passe :
export ZABBIX_USER="Admin"
export ZABBIX_PASSWORD="zabbix"

### Jeton :
export ZABBIX_TOKEN="your_api_token"
```

Ensuite, si vous démarrez votre script depuis la même session de terminal, l'instance `ZabbixAPI` dans le script ne nécessite que l'URL de l'interface web Zabbix :

```
api = ZabbixAPI(url="127.0.0.1/zabbix")
```

Vous pouvez également stocker l'URL dans une variable d'environnement :

```
### Nom d'utilisateur et mot de passe :
export ZABBIX_USER="Admin"
export ZABBIX_PASSWORD="zabbix"
export ZABBIX_URL="https://127.0.0.1/zabbix"

### Jeton :
export ZABBIX_TOKEN="your_api_token"
export ZABBIX_URL="https://127.0.0.1/zabbix"
```

Lorsque l'URL et les identifiants sont tous définis comme variables d'environnement, l'instance `ZabbixAPI` dans le script ne nécessite aucun argument :

```
api = ZabbixAPI()
```

Déconnexion

Si vous vous êtes connecté à l'aide d'un nom d'utilisateur et d'un mot de passe, appelez la méthode `logout()` après avoir terminé vos opérations API :

```
from zabbix_utils import ZabbixAPI

api = ZabbixAPI(url="127.0.0.1/zabbix")
api.login(user="Admin", password="zabbix")

### API actions go here

api.logout()
```

> **Note:**
> L'appel de la méthode `logout()` n'est pas nécessaire lors de l'utilisation de jetons API.

Requêtes API

Après vous être connecté, vous pouvez effectuer n'importe quelles requêtes API en appelant les méthodes décrites dans la référence des méthodes de l'API Zabbix.

Les méthodes de l'API sont appelées en utilisant le format suivant :

```
api_instance.zabbix_object.method(parameters)
```

> **Note:**
> Certains noms de méthodes ou d'objets de l'API Zabbix utilisent des mots qui sont des mots-clés réservés en Python (par exemple, `import`). Pour éviter les erreurs Python, ajoutez un trait de soulignement au nom de la méthode ou de l'objet lors de son utilisation en Python (par exemple, `api.configuration.import_`).

Par exemple, pour afficher une liste d'hôtes avec `host.get` :

```
### 1. Importer ZabbixAPI depuis zabbix_utils :
from zabbix_utils import ZabbixAPI

### 2. Créer l'instance ZabbixAPI et se connecter :
api = ZabbixAPI(url="127.0.0.1/zabbix")
api.login(user="Admin", password="zabbix")

### 3. Récupérer les hôtes correspondant à un filtre :
hosts = api.host.get(
    filter={
        "host": [
            "Zabbix server",
            "Linux server"
        ]
    }
)

### 4. Parcourir les hôtes renvoyés et afficher les détails de chaque hôte :
for host in hosts:
    print(host)

### 5. Se déconnecter de l'API pour fermer la session :
api.logout()
```

Mode asynchrone

Le mode asynchrone permet à votre script d'envoyer des requêtes API sans attendre que chacune d'elles se termine. Cela peut rendre votre script plus efficace lorsqu'il doit effectuer de nombreuses requêtes API ou lorsque certaines requêtes prennent beaucoup de temps.

Lors de l'utilisation du mode asynchrone, il existe des différences importantes par rapport au mode synchrone :

- Importez le module `asyncio` de Python (vous devez d'abord installer les dépendances requises).
- Importez `AsyncZabbixAPI` au lieu de `ZabbixAPI`.
- Écrivez votre code dans une fonction `async`.
- Utilisez `await` lors de l'appel des méthodes de l'API, y compris `login()` et `logout()`.

> **Attention:**
> Lors de la création d'une instance `AsyncZabbixAPI`, vous ne pouvez pas vous connecter pendant l'initialisation.

Par exemple, pour afficher une liste d'hôtes avec `host.get` en mode asynchrone :

```
### 1. Importer asyncio pour le mode asynchrone, et AsyncZabbixAPI depuis zabbix_utils :
import asyncio
from zabbix_utils import AsyncZabbixAPI

### 2. Définir la fonction asynchrone principale où toutes les opérations API seront exécutées :
async def main():

    # 3. Créer une instance AsyncZabbixAPI et se connecter (await obligatoire) :
    api = AsyncZabbixAPI(url="127.0.0.1")
    await api.login(user="User", password="zabbix")

    # 4. Récupérer les hôtes correspondant à un filtre (await obligatoire) :
    hosts = await api.host.get(
```

```
        filter={
            "host": [
                "Zabbix server",
                "Linux server"
            ]
        }
    )

    # 5. Parcourir les hôtes renvoyés et afficher les détails de chaque hôte :
    for host in hosts:
        print(host)

    # 6. Se déconnecter de l'API pour fermer la session (await obligatoire) :
    await api.logout()

### 7. Exécuter la fonction asynchrone main() à l'aide de la boucle d'événements d'asyncio :
asyncio.run(main())
```

Exemples

Les exemples suivants montrent des tâches courantes de l'API Zabbix utilisant la bibliothèque zabbix_utils.

Des exemples supplémentaires sont disponibles dans le répertoire `examples` du dépôt GitHub zabbix_utils.

Collecter des données depuis l'agent Zabbix

Vue d'ensemble

zabbix_utils vous permet de collecter des données depuis l'agent Zabbix (de manière similaire à Zabbix get).

Les données peuvent être collectées en mode synchrone ou asynchrone :

- En mode synchrone, votre script Python demande les données et attend de les recevoir avant de continuer, ce qui convient aux opérations simples, séquentielles et prévisibles.
- En mode asynchrone, le script demande les données sans attendre chaque réponse, ce qui permet à d'autres opérations de se poursuivre en parallèle ; cela est plus efficace pour les requêtes lentes ou les lots de données volumineux.

Les exemples de cette page se concentrent sur le mode synchrone, bien que le mode asynchrone suive des principes similaires. Des exemples supplémentaires sont disponibles dans le dépôt GitHub zabbix_utils.

Importation

Pour utiliser zabbix_utils afin de collecter les valeurs des éléments, importez la classe `Getter` dans votre script Python :

```
from zabbix_utils import Getter
```

Données de requête

Pour demander la valeur d'un élément :

1. Créez une instance `Getter`, en spécifiant l'adresse IP et le port de votre agent Zabbix.
2. Appelez la méthode `get()` sur l'instance `Getter`, en spécifiant la clé de l'élément que vous souhaitez récupérer.

Par exemple, pour demander des données pour l'élément `system.uname` :

```
agent = Getter(host='192.0.2.1', port=10050)
response = agent.get('system.uname')
```

Utilisation d'une adresse IP non par défaut

Si le serveur exécutant votre script possède plusieurs adresses IP, vous pouvez spécifier une `source_ip` pour que `Getter` l'utilise lors de la connexion à l'agent Zabbix :

```
agent = Getter(
    host='192.0.2.1',
    port=10050,
    source_ip='10.10.7.1'
)
```

Utilisation du délai d'attente

Vous pouvez définir un `timeout` de réponse pour le `Getter` afin de contrôler combien de temps votre script doit attendre une réponse de l'agent Zabbix avant d'abandonner :

```
agent = Getter(
    host='192.0.2.1',
    port=10050,
    timeout=30
)
```

Utilisation du chiffrement

`Getter` n'inclut pas de prise en charge intégrée du chiffrement, mais vous pouvez l'ajouter en créant un wrapper à l'aide de bibliothèques tierces :

```
def psk_wrapper(sock, tls):
    # ...
    # Implementation of TLS PSK wrapper for the socket
    # ...

agent = Getter(
    host='192.0.2.1',
    port=10050,
    socket_wrapper=psk_wrapper
)
```

Réponse

La réponse de l'agent Zabbix est traitée par la bibliothèque et renvoyée sous la forme d'un objet `AgentResponse` :

```
print(response)
### {
###     "error": null,
###     "raw": "Linux zabbix_server 5.15.0-3.60.5.1.el9uek.x86_64",
###     "value": "Linux zabbix_server 5.15.0-3.60.5.1.el9uek.x86_64"
### }

print(response.value)
### Linux zabbix_server 5.15.0-3.60.5.1.el9uek.x86_64

print(response.error)
### None
```

Mode asynchrone

Le mode asynchrone permet à votre script de collecter des valeurs sans attendre que chacune d'elles arrive. Cela peut rendre votre script plus efficace lorsqu'il doit collecter de nombreuses valeurs ou lorsque certaines valeurs prennent beaucoup de temps à être collectées.

Lors de l'utilisation du mode asynchrone, il existe des différences importantes par rapport au mode synchrone :

- Importez le module Python `asyncio` (vous devez d'abord installer les dépendances requises).
- Importez `AsyncGetter` au lieu de `Getter`.
- Écrivez votre code à l'intérieur d'une fonction `async`.
- Utilisez `await` lors de l'appel de la méthode `get()`.

Par exemple, pour collecter une seule valeur en utilisant le mode asynchrone :

```
### 1. Import asyncio for asynchronous mode, and AsyncGetter from zabbix_utils:
import asyncio
from zabbix_utils import AsyncGetter

### 2. Define the main async function where all data requests will be executed:
async def main():
    agent = AsyncGetter(host='192.0.2.1', port=10050)

    # 3. Fetch the system.uname value from Zabbix agent (must await):
    response = await agent.get('system.uname')
```

```
    # 4. Print the value returned by Zabbix agent:
    print(response.value)

### 5. Run the async main() function using asyncio's event loop:
asyncio.run(main())
```

Envoyer des données au serveur ou au proxy Zabbix

zabbix_utils vous permet d'envoyer des valeurs d'élément à un élément trapper sur le serveur ou le proxy Zabbix (de manière similaire à Zabbix sender).

Vous pouvez envoyer une seule valeur, plusieurs valeurs, ou même cibler plusieurs clusters Zabbix.

Les données peuvent être envoyées en mode synchrone ou asynchrone :

- En mode synchrone, votre script Python envoie des valeurs et attend une réponse avant de continuer ; cela convient aux opérations simples, séquentielles et prévisibles.
- En mode asynchrone, le script envoie des valeurs sans attendre chaque réponse, ce qui permet à d'autres opérations de se poursuivre en parallèle ; cela est plus efficace pour les requêtes lentes ou les lots de données volumineux.

Les exemples de cette page se concentrent sur le mode synchrone, bien que le mode asynchrone suive des principes similaires. Des exemples supplémentaires sont disponibles dans le dépôt GitHub zabbix_utils.

Importation

Pour utiliser zabbix_utils afin d'envoyer des valeurs d'élément, importez la classe `Sender` dans votre script :

```
from zabbix_utils import Sender
```

Pour envoyer plusieurs valeurs, vous pouvez également importer la classe `ItemValue` :

```
from zabbix_utils import Sender, ItemValue
```

Envoyer une valeur unique

Pour envoyer une valeur d'élément :

1. Créez une instance de `Sender`, en spécifiant l'adresse IP et le port de votre serveur ou proxy Zabbix.
2. Appelez la méthode `send_value()` sur l'instance `Sender` en utilisant le format suivant :

```
sender_instance.send_value('host', 'item.key', 'value', optional_timestamp, optional_nanoseconds)
```

Par exemple, pour envoyer 1 à l'élément trapper `service.status` sur l'hôte `Linux server` :

```
sender = Sender(server='127.0.0.1', port=10051)
response = sender.send_value('Linux server', 'service.status', 1)
```

Utilisation d'une IP non par défaut

Si le serveur exécutant votre script possède plusieurs adresses IP, vous pouvez spécifier un `source_ip` que le `Sender` utilisera lors de l'envoi des valeurs au serveur Zabbix ou au proxy :

```
sender = Sender(
    server='127.0.0.1',
    port=10051,
    source_ip='10.10.7.1'
)
```

Utilisation du délai d'expiration

Vous pouvez définir un `timeout` de réponse pour le `Sender` afin de contrôler combien de temps votre script doit attendre une réponse du serveur Zabbix ou du proxy avant d'abandonner :

```
sender = Sender(
    server='127.0.0.1',
    port=10051,
    timeout=30
)
```

Utilisation du fichier de configuration de l'agent

Vous pouvez laisser zabbix_utils lire les paramètres Server ou ServerActive à partir d'un fichier de configuration local de l'agent ou de l'agent 2 Zabbix. Dans ce cas, vous n'avez pas besoin de spécifier les paramètres de connexion lors de la création d'une instance Sender :

```
sender = Sender(
    use_config=True,
    config_path='/etc/zabbix/zabbix_agent2.conf'
)
```

Attention:
Si ServerActive contient un ou plusieurs clusters Zabbix avec plusieurs instances de serveur, Sender envoie les données au premier serveur disponible de chaque cluster. Si ServerActive n'est pas défini, l'adresse indiquée dans Server avec le port par défaut (10051) est utilisée.

Utilisation du chiffrement

Le Sender n'inclut pas de prise en charge intégrée du chiffrement, mais vous pouvez l'ajouter en créant un wrapper à l'aide de bibliothèques tierces :

```
def psk_wrapper(sock, tls):
    # ...
    # Implémentation du wrapper TLS PSK pour le socket
    # ...

sender = Sender(
    server='127.0.0.1',
    port=10051,
    socket_wrapper=psk_wrapper
)
```

Réponse pour une valeur unique

La réponse renvoyée par le serveur Zabbix ou le proxy est traitée par la bibliothèque et renvoyée sous la forme d'un objet TrapperResponse :

```
print(response)
### {"processed": 1, "failed": 0, "total": 1, "time": "0.000123", "chunk": 1}

print(response.processed)
### 1

print(response.failed)
### 0

print(response.total)
### 1
```

Mode asynchrone

Le mode asynchrone permet à votre script Python d'envoyer des valeurs sans attendre de réponse du serveur Zabbix ou du proxy. Cela peut rendre votre script plus efficace lorsqu'il doit envoyer de nombreuses valeurs ou lorsque certaines valeurs prennent beaucoup de temps à être envoyées.

Lors de l'utilisation du mode asynchrone, il existe quelques différences importantes par rapport au mode synchrone :

- Importez le module asyncio de Python (vous devez d'abord installer les dépendances requises).
- Importez AsyncSender au lieu de Sender.
- Écrivez votre code dans une fonction async.
- Utilisez await lors de l'appel de la méthode send_value().

Par exemple, pour envoyer une seule valeur en utilisant le mode asynchrone :

```
### 1. Import asyncio for asynchronous mode, and AsyncSender from zabbix_utils:
import asyncio
from zabbix_utils import AsyncSender

### 2. Define the main async function where all data sending operations (must await) will be executed:
```

```
async def main():
    sender = AsyncSender(server='127.0.0.1', port=10051)
    response = await sender.send_value('Linux server', 'service.status', 1)

    # 3. Print the response returned by Zabbix server or proxy:
    print(response)

### 4. Run the async main() function using asyncio's event loop:
asyncio.run(main())
```

Envoyer plusieurs valeurs

Pour envoyer plusieurs valeurs :

1. Préparez un tableau d'objets `ItemValue`, chacun utilisant le même format que la méthode `send_value()`.
2. Créez une instance de `Sender`, en spécifiant l'adresse IP et le port de votre serveur Zabbix ou proxy.
3. Appelez la méthode `send()` (au lieu de `send_value()`) sur l'instance `Sender`, en indiquant le tableau d'objets contenant les valeurs à envoyer.

Par exemple, pour envoyer cinq valeurs à différents hôtes :

```
items = [
    ItemValue('server-de', 'service.status', 'up', 1770887205, 100),
    ItemValue('server-fr', 'service.status', 'up', 1770887205, 100),
    ItemValue('server-uk', 'service.status', 'up', 1770887205, 100),
    ItemValue('server-nl', 'service.status', 'up', 1770887205, 100),
    ItemValue('server-pl', 'service.status', 'up', 1770887205, 100),
]

sender = Sender(server='127.0.0.1', port=10051)
response = sender.send(items)
```

Utilisation d'une taille de bloc personnalisée

Si vous devez envoyer plus de valeurs qu'un élément collecteur ne peut en accepter dans une seule requête, vous pouvez les diviser en blocs.

Par défaut, la taille du bloc est de 250 valeurs. Vous pouvez la modifier en définissant le paramètre `chunk_size` lors de la création d'une instance de `Sender`.

Par exemple, pour envoyer cinq valeurs en trois blocs (2-2-1), définissez le paramètre `chunk_size` sur 2 :

```
items = [
    ItemValue('server-de', 'service.status', 'up'),
    ItemValue('server-fr', 'service.status', 'up'),
    ItemValue('server-uk', 'service.status', 'up'),
    ItemValue('server-nl', 'service.status', 'up'),
    ItemValue('server-pl', 'service.status', 'up'),
]

sender = Sender(server='127.0.0.1', port=10051, chunk_size=2)
response = sender.send(items)
```

Envoyer des valeurs à plusieurs clusters Zabbix

Pour envoyer des valeurs à plusieurs clusters Zabbix :

1. Préparez un tableau de clusters Zabbix. Si un cluster comporte plusieurs nœuds, la valeur sera envoyée au premier nœud **disponible** de chaque cluster.
2. Créez un `Sender` en spécifiant votre tableau de clusters Zabbix.
3. Appelez la méthode `send_value()` sur l'instance `Sender` en utilisant le même format que la méthode `send_value()`.

Par exemple, pour envoyer une valeur au premier nœud disponible de chaque cluster :

```
zabbix_clusters = [
    ['zabbix.cluster1.node1', 'zabbix.cluster1.node2:10051'],
    ['zabbix.cluster2.node1:10051', 'zabbix.cluster2.node2', 'zabbix.cluster2.node3']
]
```

```
sender = Sender(clusters=zabbix_clusters)
response = sender.send_value('Linux server', 'service.status', 1)
```

Réponse pour plusieurs valeurs

Par défaut, Sender renvoie un résultat agrégé de l'envoi des valeurs sur tous les hôtes ou clusters :

```
print(response)
### {"processed": 2, "failed": 0, "total": 2, "time": "0.000108", "chunk": 2}
```

Si vous avez besoin d'informations plus détaillées, vous pouvez examiner les résultats pour chaque cluster et chaque bloc à l'aide de l'attribut `response.details` :

```
print(response)
### {"processed": 2, "failed": 0, "total": 2, "time": "0.000108", "chunk": 2}

if response.failed == 0:
    print(f"Valeur envoyée avec succès en {response.time}")
else:
    print(response.details)
    # {
    #     127.0.0.1:10051: [
    #         {
    #             "processed": 1,
    #             "failed": 0,
    #             "total": 1,
    #             "time": "0.000051",
    #             "chunk": 1
    #         }
    #     ],
    #     zabbix.example.local:10051: [
    #         {
    #             "processed": 1,
    #             "failed": 0,
    #             "total": 1,
    #             "time": "0.000057",
    #             "chunk": 1
    #         }
    #     ]
    # }

    for node, chunks in response.details.items():
        for resp in chunks:
            print(f"Traité {resp.processed} sur {resp.total} à {node.address}:{node.port}")
            # Traité 1 sur 1 à 127.0.0.1:10051
            # Traité 1 sur 1 à zabbix.example.local:10051
```

Journalisation de débogage

Pour résoudre les problèmes lors de l'utilisation de l'API Zabbix, de la collecte de données depuis l'agent Zabbix ou de l'envoi de données au serveur ou au proxy Zabbix, vous pouvez activer la journalisation de débogage.

La bibliothèque zabbix_utils utilise le module `logging` standard de Python.

Pour afficher les messages de débogage détaillés de la bibliothèque :

1. Importez le module `logging` de Python dans votre script.
2. Configurez le système de journalisation pour afficher les messages dans un format lisible et définissez le niveau de journalisation sur DEBUG.

```
import logging
from zabbix_utils import Getter

logging.basicConfig(
```

```
    format=u'[%(asctime)s] %(levelname)s %(message)s',
    level=logging.DEBUG
)

agent = Getter(host='127.0.0.1', port=10050)
resp = agent.get('system.uname')

print(resp.value)
```

Une fois la journalisation de débogage activée, des informations détaillées sur les requêtes et les réponses sont affichées, par exemple :

```
[2023-10-01 12:00:01,587] DEBUG Content of the packet: b'ZBXD\x01\x0c\x00\x00\x00\x00\x00\x00\x00system.un
[2023-10-01 12:00:01,722] DEBUG Zabbix response header: b'ZBXD\x01C\x00\x00\x00C\x00\x00\x00'
[2023-10-01 12:00:01,723] DEBUG Zabbix response body: Linux test_server 5.15.0-3.60.5.1.el9uek.x86_64
[2023-10-01 12:00:01,724] DEBUG Response from [127.0.0.1:10050]: Linux test_server 5.15.0-3.60.5.1.el9uek.
Linux test_server 5.15.0-3.60.5.1.el9uek.x86_64
```

Modules

Qu'est-ce qu'un module frontend PHP ?

- Un module est une entité avec un identifiant unique, un nom, une description, un auteur et d'autres champs définis dans son fichier manifeste, avec PHP, Javascript et d'autres fichiers situés dans un seul répertoire à l'intérieur du répertoire *modules* de votre installation frontend Zabbix (par exemple, *zabbix/ui/modules*).
- Un module doit respecter des règles simples pour garantir un bon fonctionnement.
- Un module doit être installé (déballé) et activé dans le frontend par l'administrateur.

À quoi peut servir un module

- Ajout de nouvelles fonctionnalités via des sections frontend personnalisées ;
- Création de types de widgets de tableau de bord personnalisés (voir modules de widgets) ;
- Remplacer ou étendre les fonctionnalités existantes.

À quoi sert un module

- Enregistrer une nouvelle méthode API ou modifier une méthode existante.

Comment fonctionnent les modules

- Un module activé est lancé sur chaque requête HTTP, avant d'exécuter le code d'action.
- Le module enregistrera de nouvelles actions ou redéfinira celles existantes.
- Le module ajoutera de nouvelles sections frontend et supprimera ou redéfinira celles existantes.
- Le module s'accrochera aux événements frontend comme onBeforeAction et onTerminate, si nécessaire.
- L'action demandée est finalement exécutée en exécutant le code d'action - soit celui par défaut, soit défini par le module.

Où aller ensuite Que vous préfériez apprendre par la pratique ou lire d'abord les directives, ces pages contiennent les informations et les étapes nécessaires pour créer vos propres modules :

- Tutoriels pas à pas pour écrire votre premier module
- Structure des fichiers du module
- Spécificités du module Widget
- Exemples de modules à réutiliser

Structure des fichiers des modules

Tout le code lié à un module est stocké dans un seul répertoire à l'intérieur du répertoire **modules** de votre installation frontale Zabbix (par exemple, *zabbix/ui/modules*).

```
example_module_directory/ (obligatoire)
   manifest.json (obligatoire) Métadonnées et définition des actions.
   Module.php Initialisation du module et gestion des événements.
   actions/ Fichiers du contrôleur d'action.
   SomethingView.php
   SomethingCreate.php
   SomethingDelete.php
   data_export/
   ExportAsXml.php
   ExporterAsExcel.php
   views/ Afficher les fichiers.
   example.something.view.php
   example.something.delete.php
   assets/ Tous les fichiers supplémentaires à utiliser dans les vues. Doit être spécifié dans manifest.js
   js/ JavaScript utilisés dans les vues.
   example.something.view.js.php
   css/ Fichiers CSS utilisés dans les vues.
   example.something.css
   image.png Images utilisées dans les vues.
   example.something.file Tout fichier à utiliser dans les vues.
```

Arborescence des fichiers du module

Écriture d'un module Un processus type d'écriture d'un module consiste en les étapes suivantes (lorsque disponible, cliquez sur le nom du fichier ou du dossier pour afficher des détails supplémentaires sur l'étape) :

1. Créez un nouveau répertoire pour le module dans **zabbix/ui/modules/**.
2. Ajoutez le fichier manifest.json avec les métadonnées du module.
3. Créez le dossier *views* et définissez une ou plusieurs vues du module.
4. Créez le dossier *actions* et définissez une ou plusieurs actions du module.
5. Créez le fichier Module.php (ou Widget.php pour les widgets de tableau de bord) et définissez les règles d'initialisation et de gestion des événements.
6. Créez le dossier *assets* pour les fichiers JavaScript (à placer dans *assets/js*), les styles CSS (à placer dans *assets/css*) ou tout autre fichier supplémentaire.
7. Assurez-vous de spécifier les vues, actions et fichiers d'assets requis dans manifest.json.
8. Enregistrez le module dans l'interface Zabbix et commencez à l'utiliser.

> **Note:**
> Vous pouvez enregistrer et activer un module dès que vous créez le fichier manifest.json. Une fois le module activé, vous pouvez prévisualiser immédiatement toutes les modifications apportées aux fichiers du module en actualisant l'interface Zabbix.

manifest.json

Tout module a besoin du fichier manifest.json. Le fichier doit se trouver dans le répertoire principal du module (par exemple, *zabbix/ui/modules/module_name/manifest.json*).

Au minimum, manifest.json doit spécifier ces champs :

```
{
    "manifest_version": 2.0,
    "id": "my_ip_address",
    "name": "My IP Address",
    "namespace": "MyIPAddress",
    "version": "1.0"
}
```

Paramètres pris en charge dans **manifest.json** (cliquez sur le nom du paramètre pour obtenir une description détaillée) :

Parameter	Description	Required
manifest_version	Version du manifeste du module.	Yes
id	ID unique du module.	

Parameter	Description	Required
name	Nom du module qui sera affiché dans la section Administration.	
namespace	Espace de noms PHP pour les classes du module.	
version	Version du module.	
type	Type du module. Pour un widget, doit être défini sur *widget*	Yes for widgets, otherwise no
widget	Configuration du widget. Utilisé uniquement pour les widgets.	
actions	Actions à enregistrer avec le module.	
assets	Feuilles de style CSS et fichiers JavaScript à inclure.	No
author	Auteur du module.	
config	Valeurs par défaut pour les options personnalisées du module.	
description	Description du module.	
url	Lien vers la description du module.	

manifest_version

Version manifeste du module. Actuellement, la version prise en charge est **2.0**.

Type : Double

Exemple:

```
"manifest_version": 2.0
```

identifiant

ID du module. Doit être unique. Pour éviter de futurs conflits de noms, il est recommandé d'utiliser un préfixe pour les modules (nom de l'auteur ou de l'entreprise, ou tout autre). Par exemple, si un module est un exemple de leçons et que le nom du module est « Mon module », alors l'ID sera « exemple_mon_module ».

Type : Chaîne

Exemple:

```
"id": "exemple_mon_module"
```

nom

Nom du module qui sera affiché dans la section Administration.

Type : Chaîne

Exemple:

```
"name": "Mon module"
```

namespace

Namespace PHP pour les classes de modules.

Type : Chaîne

Exemple:

```
"namespace": "ClockWidget"
```

version

Version du module. La version sera affichée dans la section Administration.

Type : Chaîne

Exemple:

```
"version": "1.0"
```

type

Type du module. Obligatoire pour les widgets et doit être égal à « widget ».

Type : Chaîne

Default : "module"

Exemple:

```
"type": "widget"
```

Actions

Actions pour s'inscrire avec le module. La définition de la clé d'objet *class* pour chaque action est requise, les autres clés d'action sont facultatives.

Type : Objet

Clés d'objet prises en charge si *type* est un *module* :

- **write.your.action.name** (object) - le nom de l'action, doit être écrit en minuscules [a-z], en séparant les mots par des points. Prend en charge les clés :
  - **class** (string ; obligatoire) - nom de la classe d'action.
  - **layout** (string) - disposition des actions. Valeurs prises en charge : *layout.json*, *layout.htmlpage* (par défaut), *null*.
  - **view** (string) - vue des actions.

Exemple:

```
"Actions": {
    "module.exqmple.list": {
        "class": "ExampleList",
        "view": "example.list",
        "layout": "layout.htmlpage"
        }
    }
```

Clés d'objet prises en charge si *type* est un *widget* :

- **widget.{id}.view** (object) - nom de fichier et de classe pour la vue widget. Remplacez **{id}** par la valeur id du widget (par exemple, *widget.example_clock.view*). Prend en charge les clés :
  - **class** (string ; obligatoire) - nom de la classe d'action pour le mode d'affichage du widget afin d'étendre la classe CControllerDashboardWidgetView par défaut. Le fichier source de la classe doit être situé dans le répertoire *actions*.
  - **view** (string) - vue widget. Doit être situé dans le répertoire *views*. Si le fichier de vue est *widget.view.php*, ce qui est attendu par défaut, ce paramètre peut être omis. Si vous utilisez un nom différent, spécifiez-le ici.
- **widget.{id}.edit** (object) - nom de fichier pour la vue de configuration du widget. Remplacez **{id}** par la valeur id du widget (par exemple, *widget.example_clock.edit*). Prend en charge les clés :
  - **class** (string ; obligatoire) - nom de la classe d'action pour le mode d'affichage de la configuration du widget. Le fichier source de la classe doit être situé dans le répertoire *actions*.
  - **view** (string) - vue de configuration du widget. Doit être situé dans le répertoire *views*. Si le fichier de vue est *widget.edit.php*, ce qui est attendu par défaut, ce paramètre peut être omis. Si vous utilisez un nom différent, spécifiez-le ici.

Exemple:

```
"Actions": {
    "widget.tophosts.view": {
        "class": "WidgetView"
    },
    "widget.tophosts.column.edit": {
        "class": "ColumnEdit",
        "view": "column.edit",,
        "layout": "layout.json"
    }
}
```

actifs

Styles CSS et fichiers JavaScript à inclure.

Type : Objet

Clés d'objet prises en charge :

- **css** (array) - Fichiers CSS à inclure. Les fichiers doivent être situés dans le répertoire *assets/css*.
- **js** (array) - Fichiers JavaScript à inclure. Les fichiers doivent être situés dans le répertoire *assets/js*.

Exemple :

```
"assets": {
    "css": ["widget.css"],
```

```
    "js": ["class.widget.js"]
}
```

author

Auteur du module. L'auteur sera affiché dans la section Administration.

Type : Chaîne

Exemple:

```
"author": "John Smith"
```

configuration

Valeurs par défaut pour les options du module. L'objet peut contenir des clés personnalisées. Si spécifiées, ces valeurs seront écrites dans la base de données lors de l'inscription du module. Les nouvelles variables ajoutées ultérieurement seront écrites lors du premier appel. Ensuite, les valeurs des variables ne peuvent être modifiées que directement dans la base de données.

Type : Objet

Exemple :

```
"config": {
    "username": "Administrateur",
    "password": "",
    "auth_url": "https://exemple.com/auth"
}
```

description

Descriptif des modules.

Type : Chaîne

Exemple:

```
"description": "Ceci est un widget horloge."
```

widget

Configuration des widgets. Utilisé si *type* est défini sur *widget*.

Type : Objet

Clés d'objet prises en charge :

- **name** (string) - utilisé dans la liste des widgets et comme en-tête par défaut. S'il est vide, le paramètre "name" du module sera utilisé.
- ** size** (objet) - dimensions du widget par défaut. Clés prises en charge :
  - *width* (entier) - largeur du widget par défaut.
  - * height* (entier) - hauteur du widget par défaut.
- **form_class** (string) - classe avec formulaire de champs de widget. Doit être situé dans le répertoire *includes*. Si la classe est *WidgetForm.php*, ce qui est attendu par défaut, ce paramètre peut être omis. Si vous utilisez un nom différent, spécifiez-le ici.
- **js_class** (string) - nom d'une classe JavaScript pour le mode d'affichage des widgets afin d'étendre la classe CWidget par défaut. La classe sera chargée avec le tableau de bord. Le fichier source de la classe doit se trouver dans le répertoire assets/js. Voir aussi : assets.
- **use_time_selector** (booléen) - détermine si le widget nécessite un sélecteur d'heure du tableau de bord. Valeurs prises en charge : *true*, *false* (par défaut).
- **refresh_rate** (entier) - taux de rafraîchissement du widget en secondes (par défaut : 60).

Exemple:

```
"widget": {
    "name": "",
    "size": {
    "width": 12,
    "height": 5
```

```
    },
    "form_class": "WidgetForm",
    "js_class": "CWidget",
    "use_time_selector": false,
    "refresh_rate": 60
}
```

url

Un lien vers la description du module. Pour les widgets, ce lien sera ouvert lorsque vous cliquerez sur l'icône d'aide ? dans la fenêtre *Ajouter un widget* ou *Modifier le widget*. Si **url** n'est pas spécifié, cliquer sur l'icône d'aide ouvrira la page générale Widgets des tableaux de bord.

Type : String

Exemple :

```
"url": "http://example.com"
```

Actions

Les actions sont responsables de la "logique métier" du module. Une action se compose généralement d'un controller et d'une vue d'action.

Un module peut :

- Appeler des actions déjà définies dans l'interface Zabbix.
- Remplacer les actions par défaut par des actions personnalisées.
- Définir des actions complètement nouvelles.

Pour remplacer un comportement d'action par défaut par un comportement personnalisé, définissez une action portant le même nom dans la configuration du module. Lorsque l'action est appelée, l'action du module sera exécutée à la place de l'action Zabbix par défaut.

Les fichiers d'action doivent être stockés dans le dossier *actions*. Les actions doivent être spécifiées dans le manifest.json.

Contrôleur

Flux de travail du contrôleur d'action :

1. Vérifier que tous les paramètres transmis dans une requête HTTP sont valides :
   - Appeler la méthode *checkInput()* du contrôleur ;
   - Utiliser les règles de validation définies dans CNewValidator.php ;
   - Appeler la méthode *validateInput()*.
2. Vérifier les autorisations de l'utilisateur.
3. Préparer les données en fonction des paramètres transmis : si *checkInput()* renvoie true, Zabbix appelle la méthode *doAction()* du contrôleur.
4. Préparer le tableau **$data** pour la vue. Utiliser *CControllerResponseData* et la méthode *setResponse()* pour stocker la réponse dans le tableau **$data**.

Exemple :

```
/**
 * Valider les paramètres d'entrée.
 *
 * @return bool
 */
protected function checkInput(): bool {
    $ret = $this->validateInput([
        'status' => 'in '.implode(',', [HOST_STATUS_MONITORED, HOST_STATUS_NOT_MONITORED])
    ]);

    if (!$ret) {
        $this->setResponse(new CControllerResponseFatal());
    }
```

```
    return $ret;
}

/**
 * Vérifier les autorisations de l'utilisateur.
 *
 * @return bool
 */
protected function checkPermissions() {
    return $this->getUserType() >= USER_TYPE_ZABBIX_ADMIN;
}

/**
 * Exécuter l'action et générer l'objet de réponse.
 */
protected function do Action(): void {
    $data = [
        'hosts_count' => API::Host()->get([
            'countOutput' => true,
            'filter' => [
                'status' => $this->getInput('status')
            ]
        ])
    ];

    $this->setResponse(new CControllerResponseData($data));
}
```

Note:
Vous pouvez consulter la liste complète des classes de contrôleur disponibles dans le code source de Zabbix.

Vues

Le fichier de vue reçoit les données d'un contrôleur, puis prépare leur rendu HTML.

Note:
La définition de vue(s) pour un module d'interface est facultative, sauf si le module est un widget.
 Les widgets de tableau de bord nécessitent au moins deux vues : une pour le mode d'édition et une pour le mode d'affichage (elles doivent être stockées dans le répertoire *views*).

Il est possible d'utiliser dans la vue des classes HTML Zabbix prédéfinies (depuis */zabbix/ui/include/classes/html*), ainsi que d'ajouter de nouvelles classes HTML et CSS. Les nouvelles classes doivent être stockées dans le dossier *assets* du module.

Exemple :

```
...
(new CColHeader(_('Name')))
```

Cela ajoutera une nouvelle colonne nommée *Name* et appliquera à la ligne supérieure du tableau le même style que sur les autres pages Zabbix.

Vue des actions

Il s'agit d'un fichier de référence pour définir une vue d'action.

```
<?php declare(strict_types = 1);

/**
* @var CView $this
*/

$this->includeJsFile('example.something.view.js.php');

(new CWidget())
```

```
    ->setTitle(_('Something view'))
    ->addItem(new CDiv($data['name']))
    ->addItem(new CPartial('module.example.something.reusable', [
        'contacts' => $data['contacts']
    ]))
    ->show();
```

Actifs

Le dossier *assets* peut contenir des fichiers et des sous-dossiers qui n'appartiennent pas à d'autres répertoires. Vous pouvez l'utiliser pour :

- Styles JavaScript (doit être à l'intérieur de *assets/js*);
- Styles CSS (doit être à l'intérieur de *assets/css*);
- Images;
- Polices;
- Tout ce dont vous avez besoin d'inclure.

actifs/js

Le répertoire *assets/js* est réservé et ne doit contenir que des fichiers JavaScript. Pour être utilisés par le widget, spécifiez ces fichiers dans le fichier *manifest.json*.

Par exemple:

```
"assets": {
     "js": ["class.widget.js"]
}
```

actifs/css

assets/css est réservé et ne doit contenir que des fichiers de style CSS. Pour être utilisés par le widget, spécifiez ces fichiers dans le fichier *manifest.json*.

Par exemple:

```
"assets": {
     "css": ["monwidget.css"]
}
```

Styles CSS

Les fichiers CSS peuvent contenir un attribut personnalisé `theme` pour définir un style différent pour un thème d'interface spécifique.

Thèmes disponibles et valeurs de leurs attributs :

- **Bleu** - [theme='blue-theme']
- **Bleu (classique)** - [theme='blue-classic-theme']
- **Sombre** - [theme='dark-theme']
- **Bleu sombre** - [theme='dark-blue-theme']
- **Sombre (classique)** - [theme='dark-classic-theme']
- **Clair à contraste élevé** - [theme='hc-light']
- **Sombre à contraste élevé** - [theme='hc-dark']

Exemple :

```
.widget {
    background-color: red;
}

[theme='dark-theme'] .widget {
    background-color: green;
}
```

Enregistrez un nouveau module

Cette section explique comment ajouter un nouveau module à l'interface Zabbix.

Prérequis Avant de continuer, assurez-vous que :

- Le module est situé dans le répertoire *modules* de votre installation frontend Zabbix (par exemple, *zabbix/ui/modules*).
- Le module dispose d'au moins une version de base du fichier manifest.json.
- Vous avez accès à la section du menu Administration dans Zabbix (nécessite le type de rôle utilisateur Super administrateur).

> **Note:**
> Le frontend n'installera pas ni même ne reconnaîtra les modules incompatibles.

Ajout d'un module Ouvrez la page *Administration* > *Général* > *Modules* et cliquez sur *Scanner le répertoire*.



Repérez votre module dans la liste et activez-le.

Pour activer un module, cliquez sur le lien hypertexte *Disabled* - l'état du module passera à *Enabled*.

Cliquez sur le nom du module pour afficher des informations supplémentaires sur le module, telles que l'auteur, la version ou une brève description (si elles sont définies dans le manifeste).

Aperçu du widget Une fois ajoutés, les modules de widget deviennent immédiatement visibles dans la liste des widgets du tableau de bord.

Vous pouvez ouvrir un tableau de bord, passer en mode édition et ajouter le widget au tableau de bord comme d'habitude.

Lorsque vous apportez des modifications au widget, actualisez le tableau de bord pour voir à quoi ressemble le widget avec les mises à jour les plus récentes.

Widgets

Les widgets sont des modules frontaux Zabbix utilisés pour les tableaux de bord. Sauf indication contraire, toutes les directives du module sont également applicables aux widgets.

Cependant, un widget est sensiblement différent d'un module. Pour créer un widget :

- préciser le type "widget" dans le fichier manifest.json ("type": "widget");
- inclure au moins deux vues : une pour le mode de présentation du widget et une pour le mode de configuration du widget (exemple.widget.view.php et exemple.widget.edit.php) ;
- et un contrôleur pour la présentation des widgets (WidgetView.php) ;
- utiliser et étendre les classes de widgets par défaut.

Configuration

Cette page décrit les classes qui peuvent être utilisées pour créer une vue de configuration de widget avec des champs de configuration personnalisés. La vue de configuration du widget est la partie du widget qui permet à l'utilisateur de configurer les paramètres du widget pour présentation.

Widget

La classe de widget principale, étend la classe de base de tous les widgets du tableau de bord - *CWidget*. Requis pour remplacer le comportement par défaut du widget.

La classe *Widget* doit être située dans le répertoire racine du widget (par exemple, *ui/modules/my_custom_widget*).

Exemple de widget.php

```
<?php

namespace Modules\MyCustomWidget ;

use Zabbix\Core\CWidget ;

class Widget extends CWidget {

    public const MY_CONSTANT = 0 ;
```

```
    public function getTranslationStrings(): array {
        return [
            'class.widget.js' => [
                'No data' => _('No data')
            ]
        ];
    }
}
```

WidgetForm

La classe *WidgetForm* étend la classe par défaut *CWidgetForm* et contient un ensemble de champs *CWidgetField* nécessaires pour définir la structure de stockage de la configuration du widget dans la base de données et gérer la validation des entrées.

La classe *WidgetForm* doit être située dans le répertoire *includes*. Si la classe porte un nom différent, ce nom doit être spécifié dans le paramètre *widget/form_class* du fichier *manifest.json*.

Exemple de includes/WidgetForm.php

```
<?php

namespace Modules\MyCustomWidget\Includes;

use Modules\MyCustomWidget\Widget;

use Zabbix\Widgets\{
    CWidgetField,
    CWidgetForm
};

use Zabbix\Widgets\Fields\{
    CWidgetFieldMultiSelectItem,
    CWidgetFieldTextBox,
    CWidgetFieldColor
};


class WidgetForm extends CWidgetForm {

    public const DEFAULT_COLOR_PALETTE = [
        'FF465C', 'B0AF07', '0EC9AC', '524BBC', 'ED1248', 'D1E754', '2AB5FF', '385CC7', 'EC1594', 'BAE37D',
        '6AC8FF', 'EE2B29', '3CA20D', '6F4BBC', '00A1FF', 'F3601B', '1CAE59', '45CFDB', '894BBC', '6D6D6D'
    ];

    public function addFields(): self {
        return $this
            ->addField(
                (new CWidgetFieldMultiSelectItem('itemid', _('Item')))
                    ->setFlags(CWidgetField::FLAG_NOT_EMPTY | CWidgetField::FLAG_LABEL_ASTERISK)
                    ->setMultiple(false)
            )
            ->addField(
                new CWidgetFieldTextBox('description', _('Description'))
            )
            ->addField(
                (new CWidgetFieldColor('chart_color', _('Color')))->setDefault('FF0000')
            );
    }
}
```

CWidgetFormView

La classe *CWidgetFormView* est requise pour définir la logique de présentation des champs définis dans la classe *WidgetForm*, en déterminant leur apparence et leur comportement lors de leur affichage dans la vue de configuration.

La classe *CWidgetFormView* prend en charge les méthodes suivantes :

- *addField()* - reçoit une instance de la classe *CWidgetField* en tant que paramètre ; chaque classe *CWidgetField* possède une classe *CWidgetFieldView* correspondante à utiliser dans la vue de configuration du widget.
- *addFieldset()* - reçoit une instance de la classe *CWidgetFieldsGroupView*, qui regroupe les champs dans un conteneur réductible.
- *addFieldsGroup()* - reçoit une instance de *CWidgetFormFieldsetCollapsibleView*, qui regroupe visuellement les champs dans un groupe (avec une bordure).
- *includeJsFile()* - permet d'ajouter un fichier JavaScript à la vue de configuration du widget.
- *addJavaScript()* - permet d'ajouter du code JavaScript intégré qui sera exécuté dès le chargement de la vue de configuration du widget.

La classe *CWidgetFormView* doit se trouver dans le répertoire *views*.

Exemple de views/widget.edit.php

```
<?php

/**
 * My custom widget form view.
 *
 * @var CView $this
 * @var array $data
 */

use Modules\MyCustomWidget\Includes\WidgetForm;

(new CWidgetFormView($data))
    ->addField(
        (new CWidgetFieldMultiSelectItemView($data['fields']['itemid']))->setPopupParameter('numeric', tru
    )
    ->addFieldset(
        (new CWidgetFormFieldsetCollapsibleView(_('Advanced configuration')))
            ->addField(
                new CWidgetFieldTextBoxView($data['fields']['description'])
            )
            ->addField(
                new CWidgetFieldColorView($data['fields']['chart_color'])
            )
    )
    ->includeJsFile('widget.edit.js.php')
    ->addJavaScript('my_custom_widget_form.init('.json_encode([
        'color_palette' => WidgetForm::DEFAULT_COLOR_PALETTE
    ]).');')
    ->show();
```

######JavaScript

Une classe JavaScript peut être utilisée pour ajouter un comportement dynamique et une interactivité à la vue de configuration du widget. Par exemple, vous pouvez initialiser un sélecteur de couleurs, défini dans la classe *CWidgetFormView*.

La classe JavaScript doit être chargée avec le formulaire, elle doit donc être référencée dans la classe *CWidgetFormView* en utilisant les méthodes *includeJsFile()* et *addJavaScript()*.

Dans l'exemple ci-dessous, une instance de classe singleton est immédiatement créée et stockée sous le nom *window.my_custom_widget_form*. Ainsi, ouvrir le formulaire pour la deuxième fois recréera l'instance.

La classe JavaScript doit être située dans le répertoire *views*.

exemple de vues/widget.edit.js.php

```
<?php

use Modules\MyCustomWidget\Widget;

?>

window.my_custom_widget_form = new class {

    init({color_palette}) {
```

```
        colorPalette.setThemeColors(color_palette);

        for (const colorpicker of jQuery('.<?= ZBX_STYLE_COLOR_PICKER ?> input')) {
            jQuery(colorpicker).colorpicker();
        }

        const overlay = overlays_stack.getById('widget_properties');

        for (const event of ['overlay.reload', 'overlay.close']) {
            overlay.$dialogue[0].addEventListener(event, () => { jQuery.colorpicker('hide'); });
        }
    }
};
```

CWidgetField

La classe *CWidgetField* est une classe de base dont héritent toutes les classes de champs de formulaire (*CWidgetFieldCheckBox*, *CWidgetFieldTextArea*, *CWidgetFieldRadioButtonList*, etc.). Les classes qui étendent *CWidgetField* sont chargées de recevoir, d'enregistrer et de valider les valeurs de configuration des widgets.

Les classes *CWidgetField* suivantes sont disponibles.

CWidgetField class	Database field type	Description
CWidgetFieldCheckBox	int32	Case à cocher unique.
CWidgetFieldCheckBoxList	array of int32	Plusieurs cases à cocher sous un seul champ de configuration.
CWidgetFieldColor	string	Champ de sélection de couleur.
CWidgetFieldDatePicker	string	Champ de sélection de date.
CWidgetFieldIntegerBox	int32	Champ permettant de saisir un entier. Peut être utilisé pour configurer des valeurs minimales et maximales.
CWidgetFieldLatLng	string	Zone de texte permettant de saisir la latitude, la longitude et le niveau de zoom de la carte, séparés par des virgules.
CWidgetFieldMultiSelect	-	Classe de base des champs de sélection multiple, étendue par toutes les classes `CWidgetFieldMultiSelect*`.
CWidgetFieldMultiSelectAction	ID	Champ de sélection multiple permettant de sélectionner des actions dans la liste des actions définies dans *Alertes* > *Actions*.
CWidgetFieldMultiSelectGraph	ID	Champ de sélection multiple permettant de sélectionner des graphiques personnalisés.
CWidgetFieldMultiSelectGraphPrototype	ID	Champ de sélection multiple permettant de sélectionner des prototypes de graphiques personnalisés.
CWidgetFieldMultiSelectGroup	ID	Champ de sélection multiple permettant de sélectionner des groupes d'hôtes.
CWidgetFieldMultiSelectHost	ID	Champ de sélection multiple permettant de sélectionner des hôtes.
CWidgetFieldMultiSelectHostInventory	array of int32	Champ de sélection multiple permettant de sélectionner des champs d'inventaire des hôtes.
CWidgetFieldMultiSelectItem	ID	Champ de sélection multiple permettant de sélectionner des éléments.
CWidgetFieldMultiSelectItemPrototype	ID	Champ de sélection multiple permettant de sélectionner des prototypes d'éléments.
CWidgetFieldMultiSelectMap	ID	Champ de sélection multiple permettant de sélectionner des cartes.
CWidgetFieldMultiSelectMediaType	ID	Champ de sélection multiple permettant de sélectionner des types de média.
CWidgetFieldMultiSelectOverrideHost	ID	Champ de sélection multiple permettant de sélectionner une source de données pour les hôtes à partir d'un tableau de bord ou d'un autre widget.
CWidgetFieldMultiSelectService	ID	Champ de sélection multiple permettant de sélectionner des services.
CWidgetFieldMultiSelectSla	ID	Champ de sélection multiple permettant de sélectionner des SLA.
CWidgetFieldMultiSelectUser	ID	Champ de sélection multiple permettant de sélectionner des utilisateurs.
CWidgetFieldNavTree	array of (multiple mixed)	Stocke une structure d'arborescence de navigation des cartes à utiliser dans le widget *Arborescence de navigation des cartes*.

CWidgetField class	Database field type	Description
CWidgetFieldNumericBox	string	Champ permettant de saisir un nombre à virgule flottante.
CWidgetFieldPatternSelect	-	Classe de base des champs de sélection de modèles, étendue par toutes les classes `CWidgetPatternSelect*`.
CWidgetFieldPatternSelectHost	string	Champ de sélection multiple permettant de sélectionner des hôtes. Prend en charge les modèles de noms d'hôtes.
CWidgetFieldPatternSelectItem	string	Champ de sélection multiple permettant de sélectionner des éléments. Prend en charge les modèles de noms d'éléments.
CWidgetFieldRadioButtonList	int32	Groupe de boutons radio composé d'un ou plusieurs boutons radio.
CWidgetFieldRangeControl	int32	Curseur permettant de sélectionner une valeur de type entier.
CWidgetFieldReference	string	Stocke un identifiant unique pour ce widget sur le tableau de bord ; utilisé pour la communication entre widgets.
CWidgetFieldSelect	int32	Liste déroulante.
CWidgetFieldSeverities	array of int32	*CWidgetFieldCheckBoxList* prédéfini avec les niveaux de gravité des déclencheurs.
CWidgetFieldSparkline	array of (multiple mixed)	Permet de configurer un graphique sparkline.
CWidgetFieldTags	array of (string, int32, string)	Permet de configurer une ou plusieurs lignes de filtres par étiquette.
CWidgetFieldTextArea	string	Zone de texte permettant de saisir du texte sur plusieurs lignes.
CWidgetFieldTextBox	string	Zone de texte permettant de saisir du texte sur une seule ligne.
CWidgetFieldThresholds	array of (string, string)	Permet de configurer des paires de couleurs et de nombres.
CWidgetFieldTimePeriod	array of string	Champ de sélection d'une période.
CWidgetFieldTimeZone	string	Liste déroulante contenant les fuseaux horaires.
CWidgetFieldUrl	string	Zone de texte permettant de saisir des URL.

Note:
La longueur maximale des champs `int32` est de 10 caractères, celle des champs `ID` de 20 caractères et celle des champs `string` de 65535 caractères. Pour les types composites (par exemple, `array of (string, int32, string)`), la limite s'applique à chaque élément individuellement. Des restrictions de longueur propres à chaque champ peuvent être définies à l'aide de méthodes telles que `setMaxLength()`.

Présentation

Cette page décrit les composants pouvant être utilisés pour créer une vue de présentation de widget. La vue de présentation du widget est la partie du widget qui reçoit les données selon sa configuration et les affiche sur le tableau de bord dans un conteneur.

La vue de présentation se compose de trois parties :

- Action du widget
- Vue widget
- JavaScript

Action du widget

La classe d'action de widget (*WidgetView*) contient des méthodes pour les opérations avec les widgets en mode d'affichage de présentation. La majorité des actions de widget utilisent et/ou étendent la classe de contrôleur par défaut *CControllerDashboard-WidgetView*.

La classe d'action du widget doit être située dans le répertoire *actions* et spécifiée dans le paramètre *actions* (*actions/widget.{id}.view/class*) dans le fichier *manifest.json*.

Exemple actions/WidgetView.php (implémenté dans le widget natif Zabbix *Informations système*)

```
class WidgetView extends CControllerDashboardWidgetView {

    protected function doAction(): void {
        $this->setResponse(new CControllerResponseData([
            'name' => $this->getInput('name', $this->widget->getDefaultName()),
            'system_info' => CSystemInfoHelper::getData(),
            'info_type' => $this->fields_values['info_type'],
            'user_type' => CWebUser::getType(),
            'user' => [
```

```
                'debug_mode' => $this->getDebugMode()
            ]
        ]));
    }
}
```

Vue des widgets

La classe de vue du widget (*CWidgetView*) est responsable de la création de la vue de présentation du widget.

La classe de vue du widget doit être située dans le répertoire *views*. Si le fichier contenant la classe de vue du widget a un nom différent de celui par défaut (*widget.view.php*), alors il doit être spécifié dans le fichier *manifest.json*sous le parametre *actions* (*actions/widget.{id}.view/view*).

exemple de vues/widget.view.php

```
<?php

/**
 * My custom widget view.
 *
 * @var CView $this
 * @var array $data
 */

(new CWidgetView($data))
    ->addItem(
        new CTag('h1', true, $data['name'])
    )
    ->show();
```

JavaScript

La classe JavaScript est chargée de déterminer le comportement du widget, tel que la mise à jour des données du widget, le redimensionnement du widget, l'affichage des éléments du widget, etc.

Toutes les opérations JavaScript utilisent et/ou étendent la classe JavaScript de base de tous les widgets du tableau de bord - *CWidget*. La classe *CWidget* contient un ensemble de méthodes avec l'implémentation par défaut du comportement des widgets. En fonction de la complexité du widget, ces méthodes peuvent être utilisées telles quelles ou étendues.

La classe *CWidget* contient les méthodes suivantes :

- Méthodes qui définissent le cycle de vie du widget : *_init()*, *_registerEvents()*, *_doActivate()*, *_doDeactivate()*, *_doDestroy()*, *setEditMode()*.
- Méthodes qui gèrent la mise à jour et l'affichage des données du widget: *_promiseUpdate()*, *_getUpdateRequestData()*, *_processUpdateResponse(response)*, *_processUpdateErrorResponse(error)*, *_setContents()*.
- Méthodes qui modifient l'apparence du widget: *resize()*, *_hasPadding()*.

La classe JavaScript doit être située dans le répertoire *assets/js* et spécifiée dans le paramètre *assets* (*assets/js*) dans le fichier *manifest.json*.

Méthodes de cycle de vie

Les méthodes de cycle de vie du widget sont invoquées par le tableau de bord et à différentes étapes du cycle de vie du widget au cours de son existence au sein du tableau de bord.

La méthode ***onInitialize()*** définit l'état initial et/ou les valeurs du widget, sans effectuer de manipulation HTML ou de données. Cette méthode est invoquée lorsqu'un widget est créé (un objet widget est instancié), généralement en ajoutant le widget à une page de tableau de bord ou en chargeant la page du tableau de bord.

Exemple:

```
onInitialize() {
    this._time_offset = 0;
    this._interval_id = null;
    this._clock_type = CWidgetClock.TYPE_ANALOG;
    this._time_zone = null;
    this._show_seconds = true;
    this._time_format = 0;
    this._tzone_format = 0;
```

```
    this._show = [];
    this._has_contents = false;
    this._is_enabled = true;
}
```

La méthode ***onStart()*** définit la structure HTML du widget, sans effectuer aucune manipulation de données. Cette méthode est invoquée avant la première activation de la page du tableau de bord, c'est-à-dire avant que le tableau de bord et ses widgets ne soient entièrement affichés à l'utilisateur.

Exemple:

```
onStart() {
    this._events.resize = () => {
        const padding = 25;
        const header_height = this._view_mode === ZBX_WIDGET_VIEW_MODE_HIDDEN_HEADER
            ? 0
            : this._header.offsetHeight;

        this._target.style.setProperty(
            '--content-height',
            `${this._cell_height * this._pos.height - padding * 2 - header_height}px`
        );
    }
}
```

La méthode ***_doActivate()*** rend le widget actif et interactif en activant des écouteurs d'événements personnalisés (pour répondre aux actions de l'utilisateur) et en initiant le cycle de mise à jour du widget (pour maintenir son contenu à jour). Cette méthode est invoquée lorsque la page du tableau de bord est activée, c'est-à-dire lorsqu'elle s'affiche entièrement dans l'interface utilisateur.

Notez qu'avant que la méthode *_doActivate()* ne soit invoquée, le widget est dans l'état inactif (`WIDGET_STATE_INACTIVE`). Après un appel réussi, le widget passe à l'état actif (`WIDGET_STATE_ACTIVE`). À l'état actif, le widget est réactif, écoute les événements, met régulièrement à jour son contenu et peut interagir avec d'autres widgets.

Exemple :

```
_doActivate() {
    super._doActivate();

    if (this._has_contents) {
        this._activateContentsEvents();
    }
}
```

La méthode ***_doDeactivate()*** arrête toute activité et interactivité du widget en désactivant les écouteurs d'événements personnalisés et en arrêtant le cycle de mise à jour du widget. Cette méthode est invoquée lorsque la page du tableau de bord est désactivée, c'est-à-dire désactivée ou supprimée, ou lorsque le widget est supprimé de la page du tableau de bord.

Notez qu'avant que la méthode *_doDeactivate()* ne soit invoquée, le widget est dans l'état actif (`WIDGET_STATE_ACTIVE`). Après un appel réussi, le widget passe à l'état inactif (`WIDGET_STATE_INACTIVE`).

Exemple :

```
_doDeactivate() {
    super._doDeactivate();

    this._deactivateContentsEvents();
}
```

La méthode ***_doDestroy()*** effectue des tâches de nettoyage avant que le widget ne soit supprimé du tableau de bord, qui peut inclure la fermeture d'une connexion à la base de données établie lors de l'initialisation du widget, nettoyer les données temporaires pour libérer la mémoire système et éviter les fuites de ressources, désenregistrer les écouteurs d'événements liés au redimensionnement des événements ou aux clics sur les boutons pour éviter la gestion inutile des événements et les fuites de mémoire, etc. Cette méthode est invoquée lorsque le widget ou la page du tableau de bord qui le contient est supprimé.

Notez qu'avant l'invocation de la méthode *_doDestroy()*, un widget dans un état actif (`WIDGET_STATE_ACTIVE`) est toujours désactivé avec l'invocation de la méthode *_doDeactivate()*.

Exemple :

```
_doDestroy() {
    super._doDestroy();

    if (this._filter_widget) {
        this._filter_widget.off(CWidgetMap.WIDGET_NAVTREE_EVENT_MARK, this._events.mark);
        this._filter_widget.off(CWidgetMap.WIDGET_NAVTREE_EVENT_SELECT, this._events.select);
    }
}
```

La méthode ***onEdit()*** définit l'apparence et le comportement du widget lorsque le tableau de bord passe en mode édition. Cette méthode est appelée lorsque le tableau de bord passe en mode édition, généralement lorsqu'un utilisateur interagit avec le bouton *Edit* du widget ou le bouton *Edit dashboard* du tableau de bord.

Exemple :

```
onEdit() {
    this._deactivateGraph();
}
```

Méthodes de processus de mise à jour

Les méthodes du processus de mise à jour du widget sont chargées de récupérer les données mises à jour à partir du serveur Zabbix ou de toute autre source de données et de les afficher dans le widget.

La méthode ***promiseUpdate()*** initie le processus de mise à jour des données en récupérant les données, généralement à l'aide de requêtes Web ou d'appels API. Cette méthode est appelée lorsqu'une page de tableau de bord est affichée et périodiquement après, jusqu'à ce que la page de tableau de bord soit basculée vers une autre page de tableau de bord.

Ce qui suit est un exemple de l'implémentation par défaut de la méthode *promiseUpdate()* utilisée par la plupart des widgets natifs de Zabbix. Dans l'implémentation par défaut, la méthode *promiseUpdate()* suit un modèle général pour récupérer les données du serveur. Elle crée un nouvel objet `Curl` avec l'URL et les paramètres de requête appropriés, envoie une requête POST à l'aide de la méthode *fetch()* avec l'objet de données construit par la méthode *getUpdateRequestData()*, et traite la réponse (ou une réponse d'erreur) avec *processUpdateResponse(response)* ou *processUpdateErrorResponse(error)* en conséquence. Cette implémentation convient à la plupart des widgets car ils récupèrent généralement les données au format JSON et les gèrent de manière cohérente.

```
promiseUpdate() {
const curl = new Curl('zabbix.php');

curl.setArgument('action', `widget.${this._type}.view`);

return fetch(curl.getUrl(), {
method: 'POST',
headers: {'Content-Type': 'application/json'},
body: JSON.stringify(this.getUpdateRequestData()),
signal: this._update_abort_controller.signal
})
.then((response) => response.json())
.then((response) => {
if ('error' in response) {
this.processUpdateErrorResponse(response.error);

return;
}

this.processUpdateResponse(response);
});
}
```

La méthode ***getUpdateRequestData()*** prépare les données de requête du serveur pour la mise à jour du widget en rassemblant diverses propriétés et leurs valeurs correspondantes (identifiants de widget, paramètres de filtre, plages horaires, etc.) à partir de l'état et de la configuration du widget, et en construisant un objet de données qui représente les informations nécessaires à envoyer au serveur dans la demande de mise à jour. Cette méthode est invoquée uniquement dans le cadre de la méthode *promiseUpdate()* par défaut, c'est-à-dire pendant le processus de mise à jour du widget.

Implémentation par défaut :

```
getUpdateRequestData() {
return {
templateid: this._dashboard.templateid ?? undefined,
dashboardid: this._dashboard.dashboardid ?? undefined,
widgetid: this._widgetid ?? undefined,
name: this._name !== '' ? this._name : undefined,
fields: Object.keys(this._fields).length > 0 ? this._fields : indéfini,
view_mode : this._view_mode,
edit_mode : this._is_edit_mode ? 1 : 0,
dynamic_hostid : this._dashboard.templateid !== null || this.supportsDynamicHosts()
? (this._dynamic_hostid ?? indéfini)
: indéfini,
...this._contents_size
};
}
```

La méthode ***processUpdateResponse(response)*** gère la réponse reçue du serveur après la demande de mise à jour, et, si le processus de mise à jour a réussi et sans erreurs, efface les données du widget et affiche le nouveau contenu avec la méthode *setContents()*. Cette méthode est invoquée uniquement dans le cadre de la méthode *promiseUpdate()* par défaut, c'est-à-dire pendant le processus de mise à jour du widget.

Implémentation par défaut :

```
processUpdateResponse(réponse) {
    this._setHeaderName(response.name);

    this._updateMessages(response.messages);
    this._updateInfo(response.info);
    this._updateDebug(response.debug);

    this.setContents(réponse);
}
```

La méthode ***processUpdateErrorResponse(error)*** gère la réponse reçue du serveur après la demande de mise à jour si la réponse est une erreur et affiche le ou les messages d'erreur. Cette méthode est invoquée uniquement dans le cadre de la méthode *promiseUpdate()* par défaut, c'est-à-dire pendant le processus de mise à jour du widget.

Implémentation par défaut :

```
processUpdateErrorResponse(error) {
this._updateMessages(error.messages, error.title);
}
```

La méthode ***setContents(response)*** affiche le contenu du widget si le processus de mise à jour du widget a réussi et sans erreur, ce qui peut inclure la manipulation d'éléments DOM, la mise à jour de composants d'interface utilisateur, l'application de styles ou de formatage, etc. Cette méthode est invoquée uniquement dans le cadre de la méthode *processUpdateResponse(response)* par défaut, c'est-à-dire pendant le processus de gestion de la réponse reçue du serveur après la demande de mise à jour.

Implémentation par défaut :

```
setContents(response) {
this._body.innerHTML = response.body ?? '';
}
```

Méthodes de modification de la présentation

Les méthodes de modification de la présentation du widget sont chargées de modifier l'apparence du widget.

La méthode ***onResize()*** est chargée d'ajuster les éléments visuels du widget pour s'adapter à la nouvelle taille du widget, qui peut inclure la réorganisation des éléments, l'ajustement des dimensions des éléments, la troncature du texte, la mise en œuvre d'un chargement paresseux pour améliorer la réactivité lors du redimensionnement, etc. Cette méthode est invoquée lorsque le widget est redimensionné, par exemple, lorsque l'utilisateur redimensionne manuellement le widget ou lorsque la fenêtre du navigateur est redimensionnée.

Exemple :

```
onResize() {
    if (this.getState() === WIDGET_STATE_ACTIVE) {
        this._startUpdating();
```

```
    }
}
```

La méthode ***hasPadding()*** est chargée d'appliquer un remplissage vertical de 8 pixels au bas du widget lorsqu'elle est configurée pour afficher son en-tête. Cette méthode est appelée lorsque la page du tableau de bord est activée, c'est-à-dire lorsqu'elle devient la page affichée dans l'interface utilisateur.

Implémentation par défaut :

```
hasPadding() {
    renvoie this.getViewMode() !== ZBX_WIDGET_VIEW_MODE_HIDDEN_HEADER;
}
```

Pour certains widgets, il est nécessaire d'utiliser tout l'espace disponible pour configurer, par exemple, une couleur d'arrière-plan personnalisée. Ce qui suit est un exemple d'implémentation de la méthode *hasPadding()* utilisée dans le widget Zabbix-native *Valeur de l'élément*.

```
hasPadding() {
    renvoie faux ;
}
```

Tutoriels

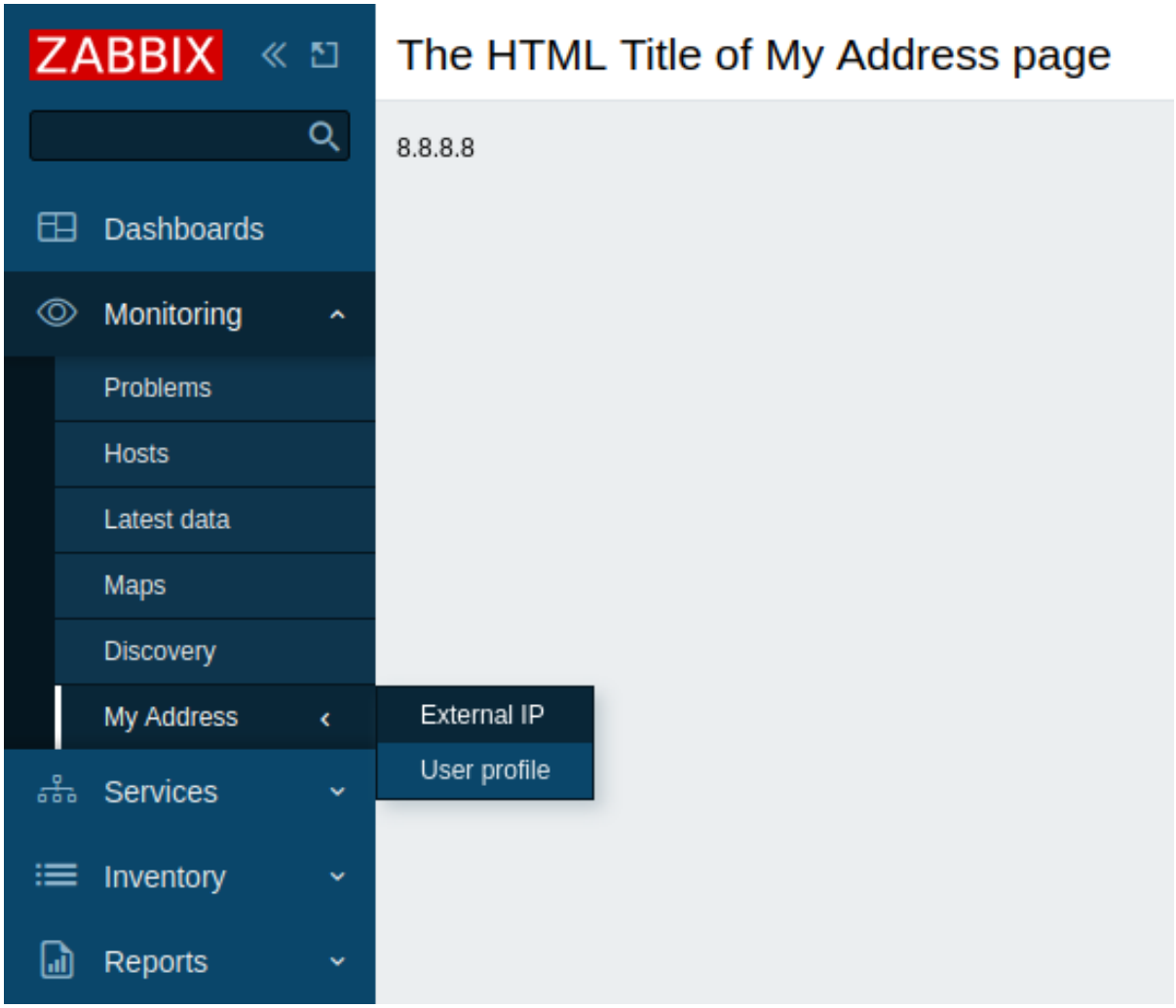
Cette section contient des tutoriels pratiques étape par étape pour illustrer comment créer un module personnalisé et un widget dans Zabbix.

Créer un module (tutoriel)

Il s'agit d'un tutoriel étape par étape qui montre comment créer un module frontal Zabbix simple. Vous pouvez télécharger tous les fichiers de ce module sous forme d'archive ZIP : MyAddress.zip.

Ce que vous allez créer

Au cours de ce tutoriel, vous allez d'abord créer un module d'interface qui ajoute une nouvelle section de menu *My Address*, puis le convertir en un module d'interface plus avancé qui effectue une requête HTTP vers *https://api.seeip.org* et affiche la réponse — l'adresse IP de votre ordinateur — sur une nouvelle page dans la section de menu *My Address* nouvellement créée. Voici à quoi ressemblera le module terminé :



Partie I - Nouvelle section de menu

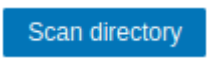
Ajouter un module vide à l'interface Zabbix

1. Créez un répertoire *MyAddress* dans le répertoire *modules* de votre installation de l'interface Zabbix (par exemple, *zabbix/ui/modules*).
2. Créez un fichier *manifest.json* avec les métadonnées de base du module (voir la description des paramètres pris en charge).

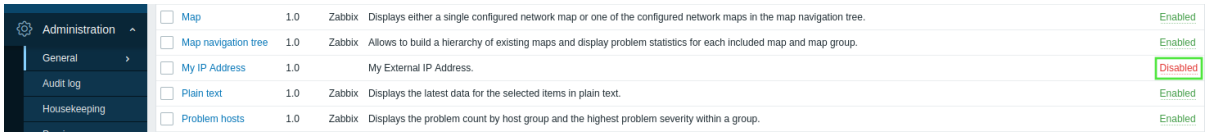
ui/modules/MyAddress/manifest.json

```
{
    "manifest_version": 2.0,
    "id": "my-address",
    "name": "My IP Address",
    "version": "1.0",
    "namespace": "MyAddress",
    "description": "My External IP Address."
}
```

3. Dans l'interface Zabbix, allez dans la section *Administration* > *Général* > *Modules* et cliquez sur le bouton *Scan directory*.



4. Repérez le nouveau module *My IP Address* dans la liste et cliquez sur le lien hypertexte "Disabled" pour changer l'état du module de "Disabled" à "Enabled" (si le module n'apparaît pas dans la liste, consultez la section dépannage).



Le module est maintenant enregistré dans l'interface. Cependant, il n'est pas encore visible, car vous devez toujours définir la fonctionnalité du module. Une fois que vous ajoutez du contenu au répertoire du module, vous verrez immédiatement les

modifications dans l'interface Zabbix après avoir actualisé la page.

Créer une section de menu

1. Créez un fichier *Module.php* dans le répertoire *MyAddress*.

Ce fichier implémente une nouvelle classe *Module* qui étend la classe par défaut *CModule*. La classe *Module* insérera une nouvelle section de menu *My Address* dans le menu principal.

La méthode *setAction()* spécifie une action à exécuter lors du clic sur la section de menu. Pour commencer, vous pouvez utiliser l'action prédéfinie *userprofile.edit*, qui ouvrira la page *Profil utilisateur*. Dans la partie III de ce tutoriel, vous apprendrez à créer une action personnalisée.

ui/modules/MyAddress/Module.php

```
<?php

namespace Modules\MyAddress;

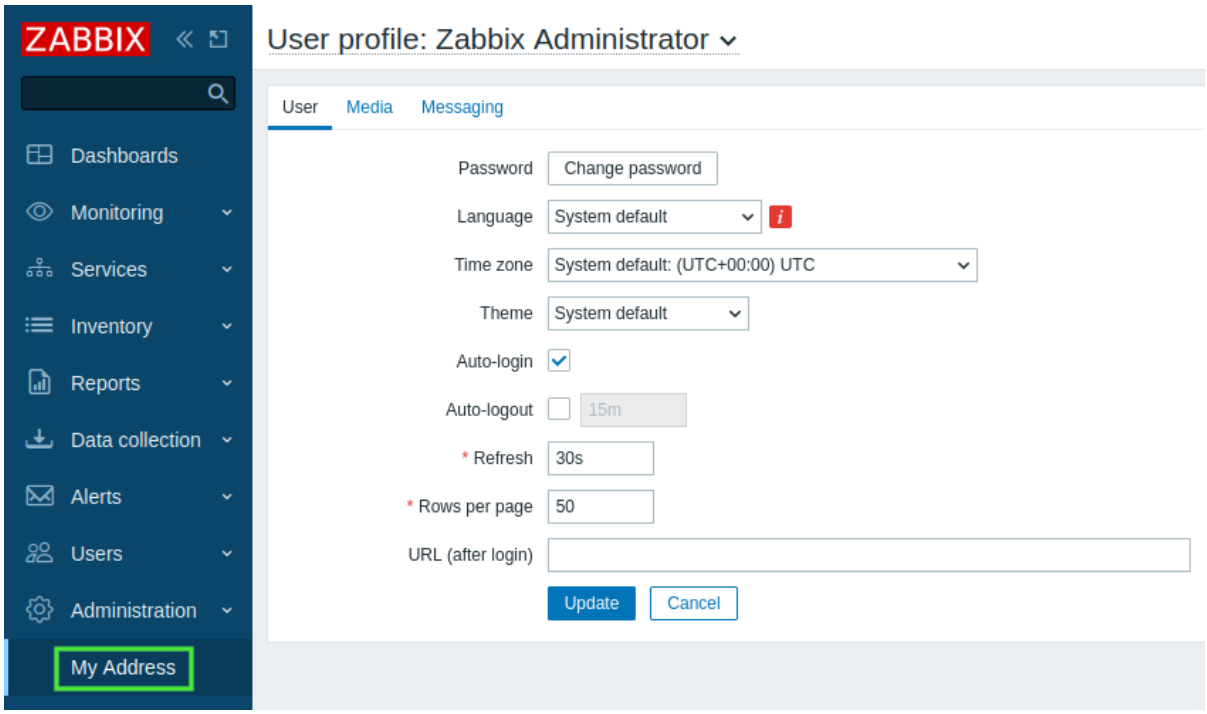
use Zabbix\Core\CModule,
    APP,
    CMenuItem;

class Module extends CModule {

    public function init(): void {
        APP::Component()->get('menu.main')
            ->add((new CMenuItem(_('My Address')))
            ->setAction('userprofile.edit'));
    }
}
```

Note:
Vous pouvez remplacer `'userprofile.edit'` par d'autres actions, par exemple `'charts.view'` (ouvre les graphiques personnalisés), `'problems.view'` (ouvre *Surveillance* > *Problèmes*), ou `'report.status'` (ouvre le rapport *Informations système*).

3. Actualisez l'interface Zabbix. Une nouvelle section *My Address* apparaît désormais en bas du menu principal de Zabbix. Cliquez sur *My Address* pour ouvrir la page *Profil utilisateur*.



Partie II - Changement d'emplacement de la section de menu

Dans cette partie, vous allez déplacer la section de menu *My Address* vers la section *Monitoring*, puis lui ajouter un menu imbriqué. Ainsi, les utilisateurs pourront accéder à deux pages de sous-menu depuis la section de menu *Monitoring* > *My Address*.

1. Ouvrez et modifiez le fichier *Module.php*.

ui/modules/MyAddress/Module.php

```
<?php

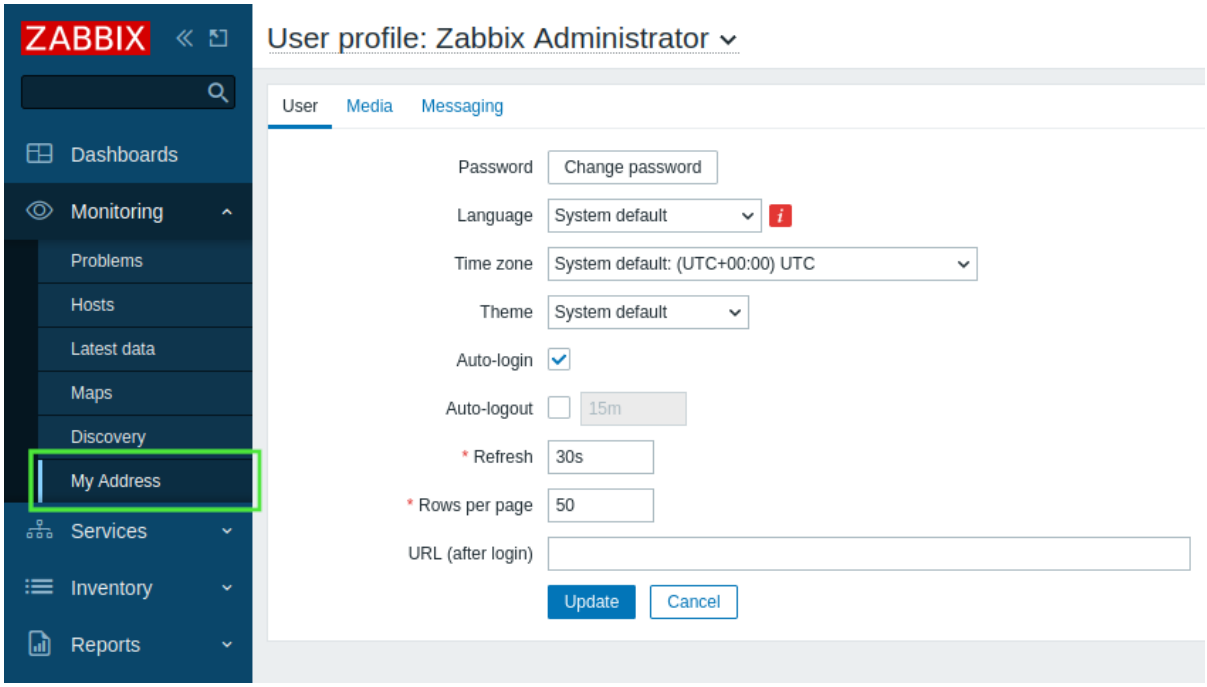
namespace Modules\MyAddress;

use Zabbix\Core\CModule,
    APP,
    CMenuItem;

class Module extends CModule {

    public function init(): void {
        APP::Component()->get('menu.main')
            ->findOrAdd(_('Monitoring'))
            ->getSubmenu()
            ->insertAfter(_('Discovery'),
                (new CMenuItem(_('My Address')))->setAction('userprofile.edit')
            );
    }
}
```

2. Actualisez l'interface Zabbix. Développez la section de menu *Monitoring* et observez que la section *My address* se trouve désormais sous la section *Discovery*.



3. Pour ajouter des pages imbriquées à la section de menu *My Address*, ouvrez à nouveau le fichier *Module.php* et modifiez-le.

Cette étape crée deux sous-sections :

- *External IP* qui exécute une nouvelle action 'my.address' qui sera définie dans les étapes suivantes ;
- *User profile* qui exécute l'action prédéfinie 'userprofile.edit' pour ouvrir la page *User profile*.

Notez que pour le menu imbriqué, vous devez utiliser la classe *CMenu* en plus des classes utilisées dans les étapes précédentes.

ui/modules/MyAddress/Module.php

```
<?php

namespace Modules\MyAddress;

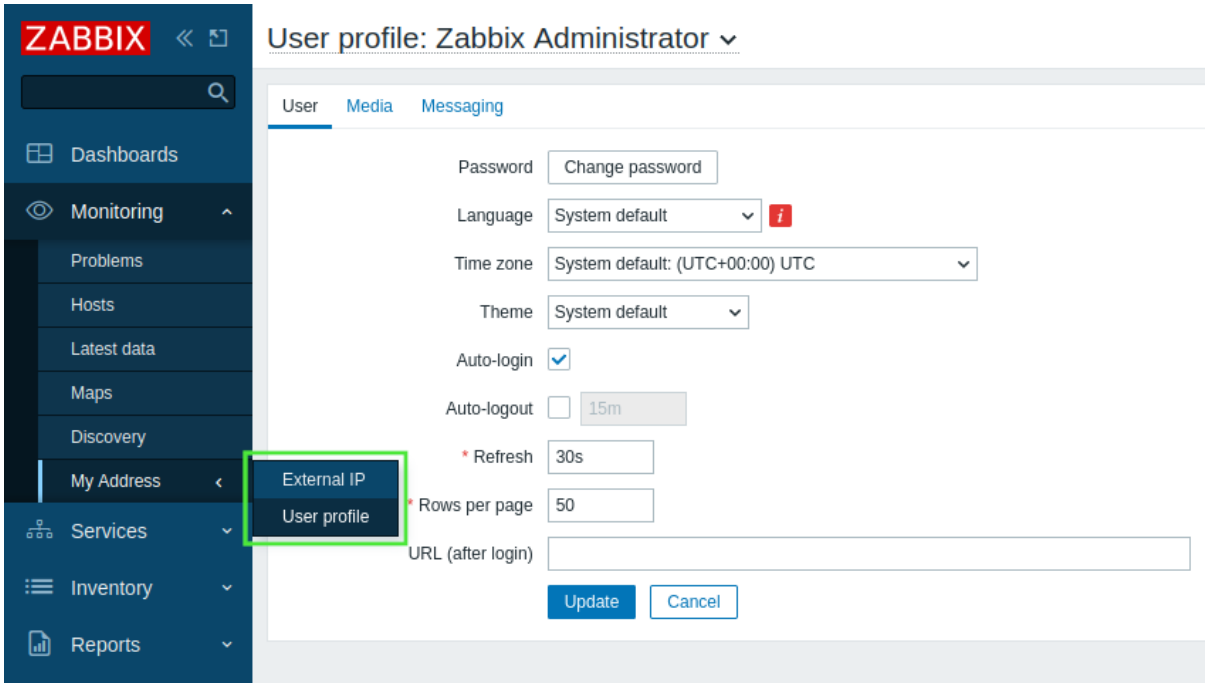
use Zabbix\Core\CModule,
    APP,
    CMenu,
```

```
    CMenuItem;

class Module extends CModule {

    public function init(): void {
        APP::Component()->get('menu.main')
            ->findOrAdd(_('Monitoring'))
            ->getSubmenu()
            ->insertAfter(_('Discovery'),
                (new CMenuItem(_('My Address')))->setSubMenu(
                    new CMenu([
                        (new CMenuItem(_('External IP')))->setAction('my.address'),
                        (new CMenuItem(_('User profile')))->setAction('userprofile.edit')
                    ])
                )
            );
    }
}
```

4. Actualisez l'interface Zabbix. Observez que la section de menu *My address* contient désormais un menu de troisième niveau avec deux pages - *External IP* et *User profile*.



Partie III - Action du module

Une action est implémentée dans deux fichiers - *actions/MyAddress.php* et *views/my.address.php*. Le fichier ***actions/MyAddress.php*** prend en charge l'implémentation de la logique métier, tandis que le fichier ***views/my.address.php*** est responsable de la vue.

1. Créez un répertoire *actions* dans le répertoire *MyAddress*.
2. Créez un fichier *MyAddress.php* dans le répertoire *actions*.

La logique de l'action sera définie dans la classe *MyAddress*. Cette classe d'action implémentera quatre fonctions : *init()*, *checkInput()*, *checkPermissions()* et *doAction()*. L'interface Zabbix appelle la fonction *doAction()* lorsque l'action est demandée. Cette fonction est responsable de la logique métier du module.

Attention:
Les données doivent être organisées sous forme de tableau associatif. Le tableau peut être multidimensionnel et peut contenir toutes les données attendues par la vue.

ui/modules/MyAddress/actions/MyAddress.php

```
<?php

namespace Modules\MyAddress\Actions;
```

```
use CController,
    CControllerResponseData;

class MyAddress extends CController {

    public function init(): void {
        $this->disableCsrfValidation();
    }

    protected function checkInput(): bool {
        return true;
    }

    protected function checkPermissions(): bool {
        return true;
    }

    protected function doAction(): void {
        $data = ['my-ip' => file_get_contents("https://api.seeip.org")];
        $response = new CControllerResponseData($data);
        $this->setResponse($response);
    }
}
```

3. Créez un nouveau répertoire *views* dans le répertoire *MyAddress*.
4. Créez un fichier *my.address.php* dans le répertoire *views* et définissez la vue du module.

Notez que la variable $data est disponible dans la vue sans avoir à la définir explicitement. Le framework transmet automatiquement le tableau associatif à la vue.

ui/modules/MyAddress/views/my.address.php

```
<?php

(new CHtmlPage())
    ->setTitle(_('Le titre HTML de la page My Address'))
    ->addItem(new CDiv($data['my-ip']))
    ->show();
```

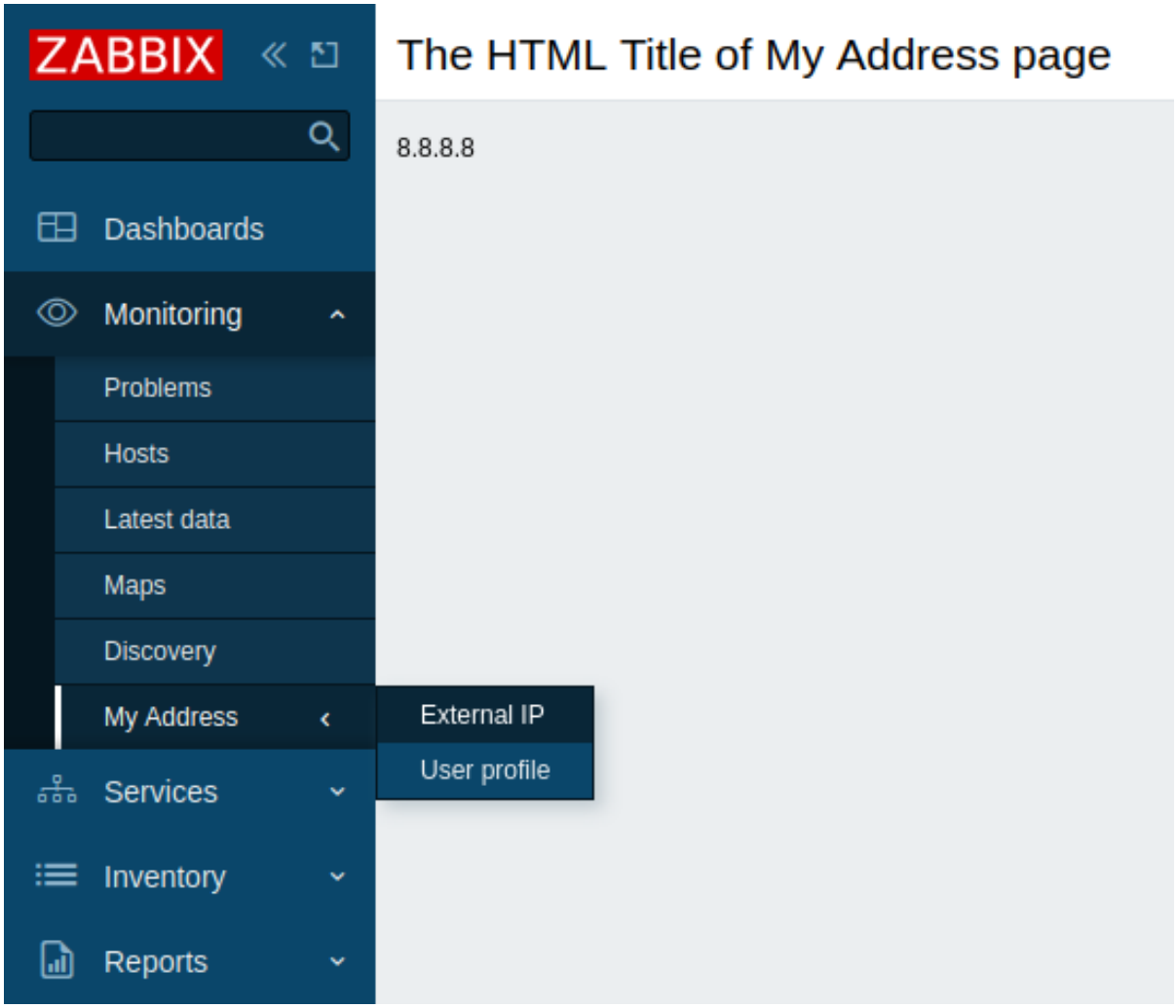
5. L'action du module doit être enregistrée dans le fichier *manifest.json*. Ouvrez *manifest.json* et ajoutez un nouvel objet `actions` qui contient :

- la clé de l'action avec le nom de l'action écrit en minuscules (a-z) et avec des mots séparés par des points (par exemple, `my.address`) ;
- le nom de la classe d'action (`MyAddress`) comme valeur de la clé `class` de l'objet `my.address` ;
- le nom de la vue de l'action (`my.address`) comme valeur de la clé `view` de l'objet `my.address`.

ui/modules/MyAddress/manifest.json

```
{
    "manifest_version": 2.0,
    "id": "my-address",
    "name": "My IP Address",
    "version": "1.0",
    "namespace": "MyAddress",
    "description": "My External IP Address.",
    "actions": {
        "my.address": {
            "class": "MyAddress",
            "view": "my.address"
        }
    }
}
```

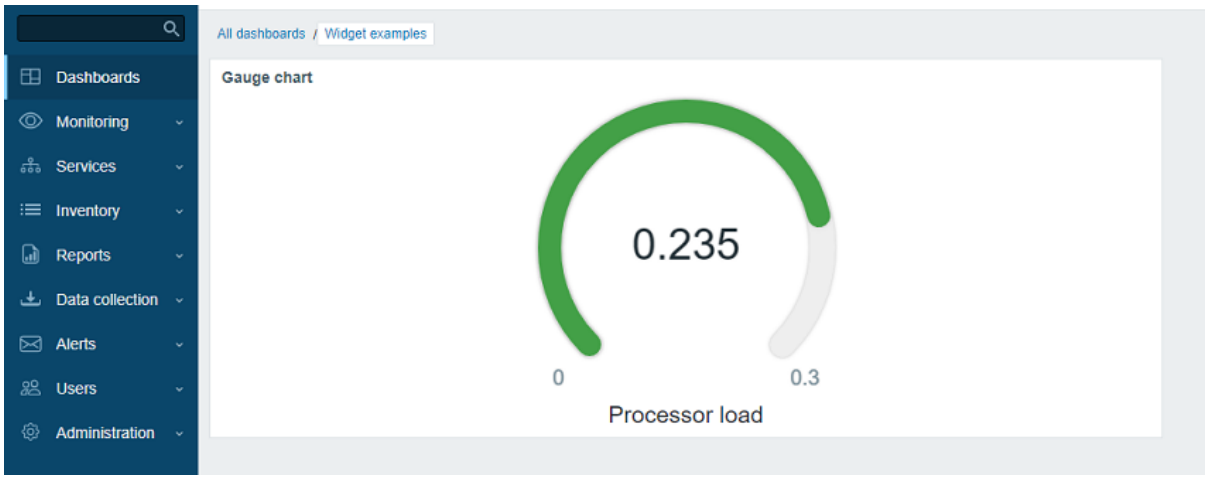
6. Actualisez l'interface Zabbix. Cliquez sur *My address* > *External IP* pour voir l'adresse IP de votre ordinateur.



Créer un widget (tutoriel)

Il s'agit d'un didacticiel étape par étape qui montre comment créer un simple widget de tableau de bord. Vous pouvez télécharger tous les fichiers de ce widget sous forme d'archive ZIP : lesson_gauge_chart.zip.

Ce que vous allez créer Au cours de ce tutoriel, vous allez d'abord créer un widget de base « Hello, world! », puis le convertir en un widget plus avancé qui affiche la valeur d'un élément sous forme de jauge. Voici à quoi ressemblera le widget final :



Partie I - "Hello, world!" Dans cette section, vous apprendrez comment créer les éléments de widget minimum requis et ajouter un nouveau widget à l'interface Zabbix.

Ajouter un widget vide à l'interface Zabbix

1. Créez un répertoire `lesson_gauge_chart` dans le répertoire `modules` de votre installation de l'interface Zabbix (par exemple, *zabbix/ui/modules*).

> **Note:**
> Tous les widgets personnalisés sont traités comme des modules externes et doivent être ajoutés au répertoire `modules` de votre installation de l'interface Zabbix (par exemple, *zabbix/ui/modules*). Le répertoire `zabbix/ui/widgets` est réservé aux widgets intégrés de Zabbix et est mis à jour en même temps que l'interface Zabbix.

2. Créez un fichier `manifest.json` avec les métadonnées de base du widget (voir la description des paramètres pris en charge).

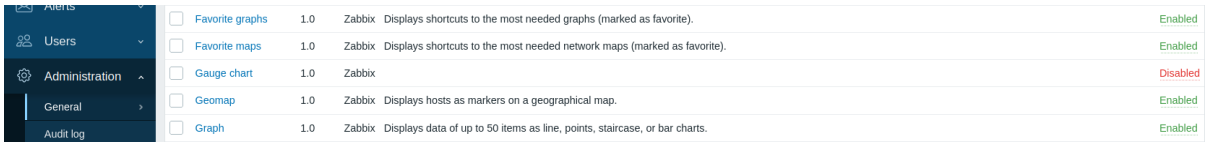
ui/modules/lesson_gauge_chart/manifest.json

```
{
    "manifest_version": 2.0,
    "id": "lesson_gauge_chart",
    "type": "widget",
    "name": "Gauge chart",
    "namespace": "LessonGaugeChart",
    "version": "1.1",
    "author": "Zabbix"
}
```

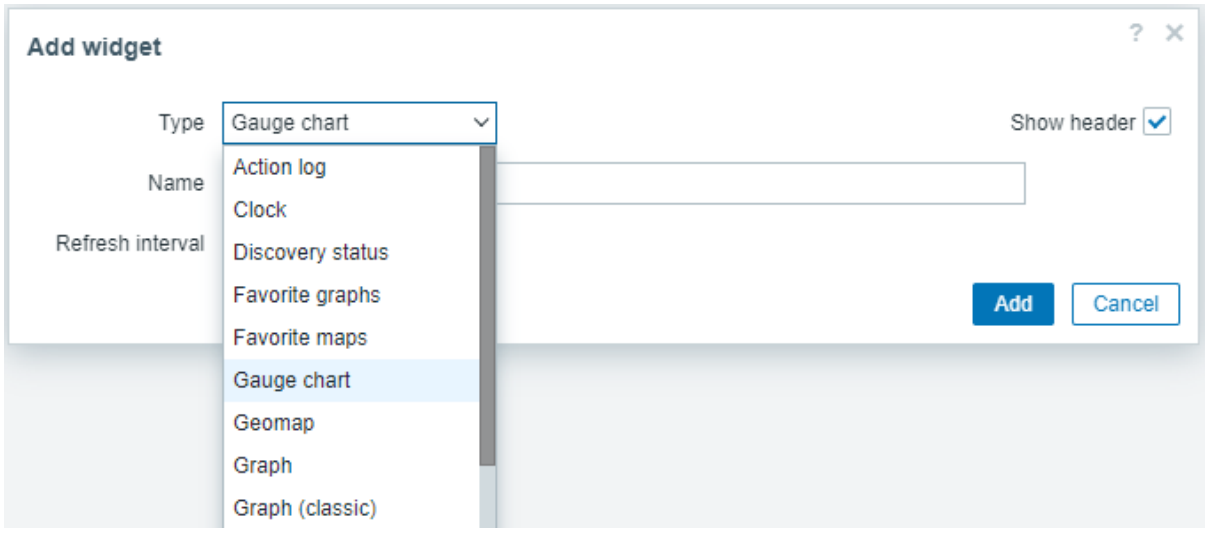
3. Dans l'interface Zabbix, allez dans la section *Administration* > *Général* > *Modules* et cliquez sur le bouton *Scan directory*.



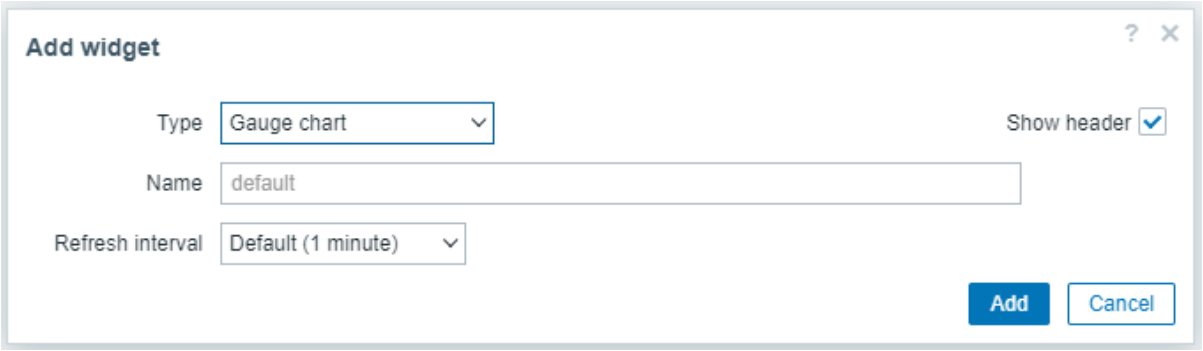
4. Repérez le nouveau module *Gauge chart* dans la liste et cliquez sur le lien hypertexte "Disabled" pour faire passer l'état du module de "Disabled" à "Enabled" (si le module n'apparaît pas dans la liste, consultez la section dépannage).



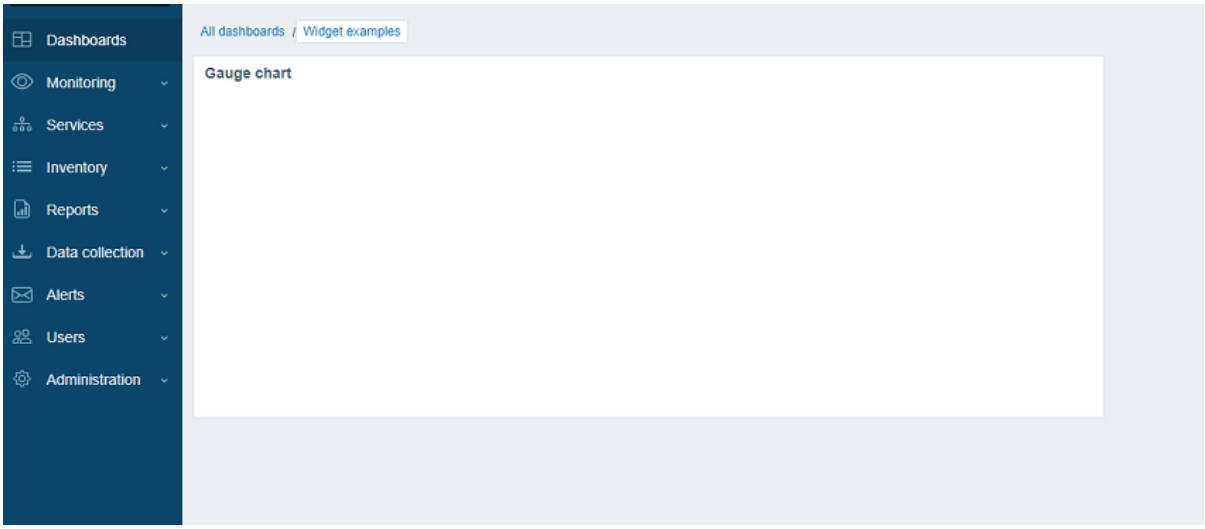
5. Ouvrez un tableau de bord, passez-le en mode édition et ajoutez un nouveau widget. Dans le champ *Type*, sélectionnez "Gauge chart".



6. À ce stade, la configuration du widget *Gauge chart* ne contient que les champs communs du widget, *Name* et *Refresh interval*. Cliquez sur *Add* pour ajouter le widget au tableau de bord.



7. Un widget vide doit apparaître sur le tableau de bord. Cliquez sur *Save changes* dans le coin supérieur droit pour enregistrer le tableau de bord.



Ajouter une vue de widget

Note:
Le fichier **view** du widget doit se trouver dans le répertoire `views` (pour ce tutoriel, `ui/modules/lesson_gauge_chart/views/`). Si le fichier porte le nom par défaut `widget.view.php`, vous n'avez pas besoin de l'enregistrer dans le fichier `manifest.json`. Si le fichier a un nom différent, indiquez-le dans la section `actions/widget.lesson_gauge_chart.view` du fichier `manifest.json`.

1. Créez un répertoire `views` dans le répertoire `lesson_gauge_chart`.
2. Créez un fichier `widget.view.php` dans le répertoire `views`.

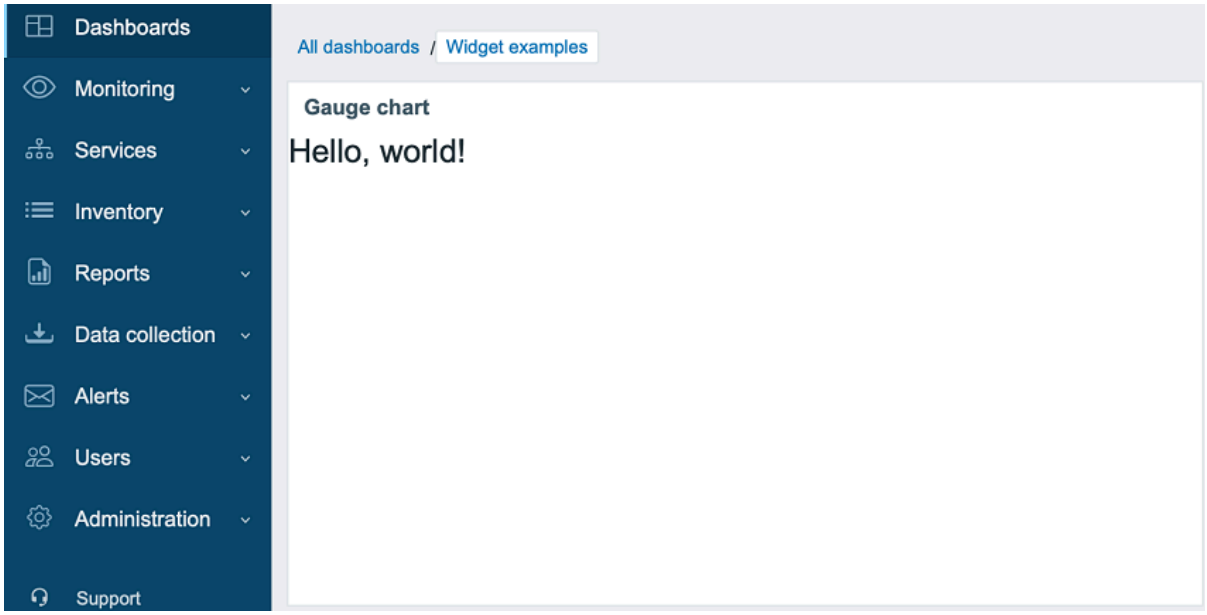
`ui/modules/lesson_gauge_chart/views/widget.view.php`

```
<?php

/**
 * Vue du widget de graphique en jauge.
 *
 * @var CView $this
 * @var array $data
 */

(new CWidgetView($data))
    ->addItem(
        new CTag('h1', true, 'Hello, world!')
    )
    ->show();
```

3. Actualisez le tableau de bord. Le widget *Gauge chart* affiche maintenant "Hello, world!".



Partie II - Tableau des jauges Ajouter des paramètres à une vue de configuration et les utiliser dans une vue de widget

Dans cette section, vous apprendrez à ajouter un champ de configuration de widget et à afficher la valeur saisie dans la vue du widget sous forme de texte.

La configuration du widget se compose d'un formulaire (`Zabbix\Widgets\CWidgetForm`) et d'une vue de formulaire de widget (`widget.edit.php`). Pour ajouter des champs (`Zabbix\Widgets\CWidgetField`), vous devez créer une classe `WidgetForm`, qui héritera de `Zabbix\Widgets\CWidgetForm`.

Le formulaire contient un ensemble de champs (`Zabbix\Widgets\CWidgetField`) de différents types, utilisés pour valider les valeurs saisies par l'utilisateur. Le champ de formulaire (`Zabbix\Widgets\CWidgetField`) de chaque type d'élément de saisie convertit la valeur dans un format unique afin de la stocker dans la base de données.

> **Note:**
> Le fichier **formulaire** du widget doit se trouver dans le répertoire `includes` (pour ce tutoriel, `ui/modules/lesson_gauge_chart/includes/`). Si le fichier porte le nom par défaut `WidgetForm.php`, vous n'avez pas besoin de l'enregistrer dans le fichier `manifest.json`. Si le fichier porte un autre nom, indiquez-le dans la section `widget/form_class` du fichier `manifest.json`.

1. Créez un nouveau répertoire `includes` dans le répertoire `lesson_gauge_chart`.
2. Créez un fichier `WidgetForm.php` dans le répertoire `includes`.

`ui/modules/lesson_gauge_chart/includes/WidgetForm.php`

```
<?php

namespace Modules\LessonGaugeChart\Includes;

use Zabbix\Widgets\CWidgetForm;

class WidgetForm extends CWidgetForm {
}
```

3. Ajoutez un champ *Description* au formulaire de configuration du widget. Il s'agit d'un champ texte standard, dans lequel l'utilisateur peut saisir n'importe quel jeu de caractères. Vous pouvez utiliser la classe `CWidgetFieldTextBox` pour cela.

`ui/modules/lesson_gauge_chart/includes/WidgetForm.php`

```
<?php

namespace Modules\LessonGaugeChart\Includes;

use Zabbix\Widgets\CWidgetForm;

use Zabbix\Widgets\Fields\CWidgetFieldTextBox;
```

```
class WidgetForm extends CWidgetForm {

    public function addFields(): self {
        return $this
            ->addField(
               new CWidgetFieldTextBox('description', _('Description'))
            );
    }
}
```

4. Dans le répertoire `views`, créez un fichier de vue de configuration de widget `widget.edit.php` et ajoutez une vue pour le nouveau champ *Description*. Pour la classe de champ `CWidgetFieldTextBox`, la vue est `CWidgetFieldTextBoxView`.

`ui/modules/lesson_gauge_chart/views/widget.edit.php`

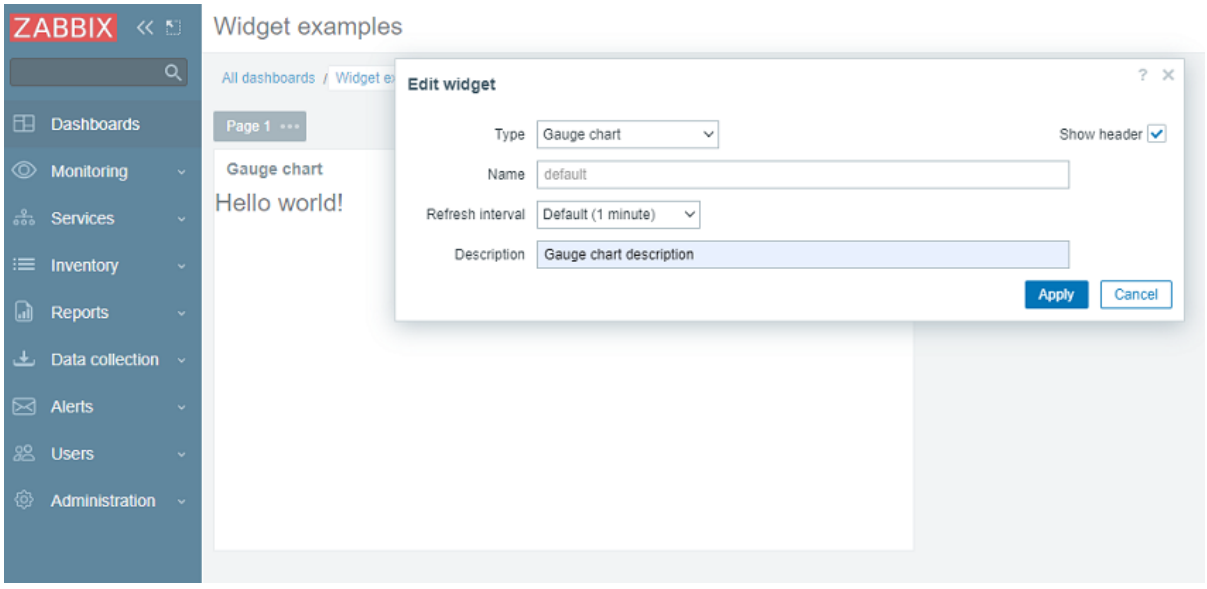
```
<?php

/**
 * Vue du formulaire du widget Gauge chart.
 *
 * @var CView $this
 * @var array $data
 */

(new CWidgetFormView($data))
    ->addField(
        new CWidgetFieldTextBoxView($data['fields']['description'])
    )
    ->show();
```

5. Allez sur le tableau de bord et cliquez sur l'icône en forme d'engrenage dans le widget pour ouvrir le formulaire de configuration du widget.

6. Le formulaire de configuration du widget contient maintenant un nouveau champ texte *Description*. Saisissez une valeur, par exemple *Gauge chart description*.



7. Cliquez sur *Apply* dans le formulaire de configuration du widget. Cliquez ensuite sur *Save changes* dans le coin supérieur droit pour enregistrer le tableau de bord. Notez que la nouvelle description n'est visible nulle part, et que le widget affiche toujours "Hello, world!".

Pour que la nouvelle description apparaisse dans le widget, la valeur du champ *Description* doit être récupérée depuis la base de données et transmise à la vue du widget. Pour cela, vous devez créer une classe d'action.

8. Créez un nouveau répertoire `actions` dans le répertoire `lesson_gauge_chart`.

9. Créez un fichier `WidgetView.php` dans le répertoire `actions`. La classe d'action `WidgetView` héritera de la classe `CControllerDashboardWidgetView`.

Les valeurs des champs de configuration du widget sont stockées dans la propriété `$fields_values` de la classe d'action.

`ui/modules/lesson_gauge_chart/actions/WidgetView.php`

```
<?php

namespace Modules\LessonGaugeChart\Actions;

use CControllerDashboardWidgetView,
    CControllerResponseData;

class WidgetView extends CControllerDashboardWidgetView {

    protected function doAction(): void {
        $this->setResponse(new CControllerResponseData([
            'name' => $this->getInput('name', $this->widget->getName()),
            'description' => $this->fields_values['description'],
            'user' => [
                'debug_mode' => $this->getDebugMode()
            ]
        ]));
    }
}
```

10. Ouvrez `manifest.json` et enregistrez `WidgetView` comme classe d'action dans la section `actions/widget.lesson_gauge_chart`

`ui/modules/lesson_gauge_chart/manifest.json`

```
{
    "manifest_version": 2.0,
    "id": "lesson_gauge_chart",
    "type": "widget",
    "name": "Gauge chart",
    "namespace": "LessonGaugeChart",
    "version": "1.0",
    "author": "Zabbix",
    "actions": {
        "widget.lesson_gauge_chart.view": {
            "class": "WidgetView"
        }
    }
}
```

11. Vous pouvez maintenant utiliser la valeur du champ description, contenue dans `$data['description']`, dans la vue du widget. Ouvrez `views/widget.view.php` et remplacez le texte statique "Hello, world!" par `$data['description']`.

`ui/modules/lesson_gauge_chart/views/widget.view.php`

```
<?php

/**
 * Vue du widget Gauge chart.
 *
 * @var CView $this
 * @var array $data
 */

(new CWidgetView($data))
    ->addItem(
        new CTag('h1', true, $data['description'])
    )
    ->show();
```

12. Actualisez la page du tableau de bord. Vous devriez maintenant voir le texte de description du widget à la place de "Hello, world!".

Récupérer la valeur d'un élément via l'API

Le widget doit afficher la dernière valeur d'un élément choisi par l'utilisateur. Pour cela, vous devez ajouter la possibilité de sélectionner des éléments dans la configuration du widget.

Dans cette section, vous apprendrez à ajouter un champ de sélection d'élément au formulaire du widget et à ajouter la partie visuelle de ce champ à la vue de configuration. Ensuite, le contrôleur du widget pourra récupérer les données de l'élément et sa valeur via une requête API. Une fois reçue, la valeur pourra être affichée dans la vue du widget.

1. Ouvrez `includes/WidgetForm.php` et ajoutez le champ `CWidgetFieldMultiSelectItem`. Cela permettra de sélectionner un élément dans le formulaire de configuration.

`ui/modules/lesson_gauge_chart/includes/WidgetForm.php`

```
<?php

namespace Modules\LessonGaugeChart\Includes;

use Zabbix\Widgets\{
    CWidgetField,
    CWidgetForm
};

use Zabbix\Widgets\Fields\{
    CWidgetFieldMultiSelectItem,
    CWidgetFieldTextBox
};

/**
 * Gauge chart widget form.
 */
class WidgetForm extends CWidgetForm {

    public function addFields(): self {
        return $this
            ->addField(
                (new CWidgetFieldMultiSelectItem('itemid', _('Élément')))
                    ->setFlags(CWidgetField::FLAG_NOT_EMPTY | CWidgetField::FLAG_LABEL_ASTERISK)
                    ->setMultiple(false)
            )
            ->addField(
                new CWidgetFieldTextBox('description', _('Description'))
            );
    }
}
```

2. Ouvrez `views/widget.edit.php` et ajoutez le composant visuel du champ à la vue de configuration.

`ui/modules/lesson_gauge_chart/views/widget.edit.php`

```
<?php

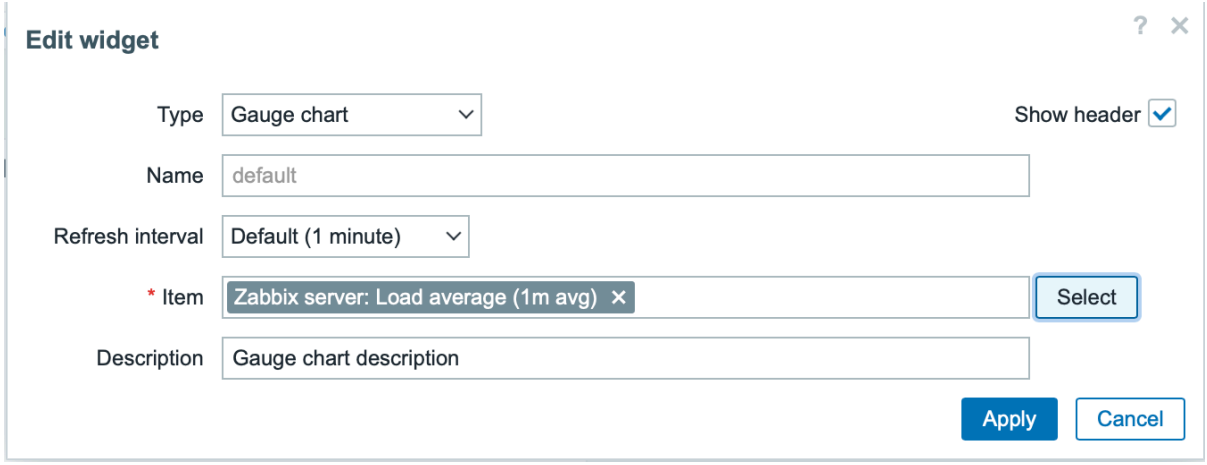
/**
 * Gauge chart widget form view.
 *
 * @var CView $this
 * @var array $data
 */

(new CWidgetFormView($data))
    ->addField(
        new CWidgetFieldMultiSelectItemView($data['fields']['itemid'])
    )
    ->addField(
        new CWidgetFieldTextBoxView($data['fields']['description'])
    )
    ->show();
```

3. Revenez au tableau de bord et cliquez sur l'icône d'engrenage dans le widget pour ouvrir le formulaire de configuration du

widget.

4. Le formulaire de configuration du widget contient maintenant un nouveau champ de saisie *Élément*. Sélectionnez l'hôte "Zabbix server" et l'élément "Load average (1m avg)".



5. Cliquez sur *Apply* dans le formulaire de configuration du widget. Puis cliquez sur *Save changes* dans le coin supérieur droit pour enregistrer le tableau de bord.

6. Ouvrez et modifiez `actions/WidgetView.php`.

Désormais, l'ID de l'élément sera disponible dans le contrôleur du widget dans `$this->fields_values['itemid']`. La méthode du contrôleur `doAction()` collecte les données de l'élément (nom, type de valeur, unités) à l'aide de la méthode API `item.get` et la dernière valeur de l'élément à l'aide de la méthode API `history.get`.

`ui/modules/lesson_gauge_chart/actions/WidgetView.php`

```
<?php

namespace Modules\LessonGaugeChart\Actions;

use API,
    CControllerDashboardWidgetView,
    CControllerResponseData;

class WidgetView extends CControllerDashboardWidgetView {

    protected function doAction(): void {
        $db_items = API::Item()->get([
            'output' => ['itemid', 'value_type', 'name', 'units'],
            'itemids' => $this->fields_values['itemid'],
            'webitems' => true,
            'filter' => [
                'value_type' => [ITEM_VALUE_TYPE_UINT64, ITEM_VALUE_TYPE_FLOAT]
            ]
        ]);

        $value = null;

        if ($db_items) {
            $item = $db_items[0];

            $history = API::History()->get([
                'output' => API_OUTPUT_EXTEND,
                'itemids' => $item['itemid'],
                'history' => $item['value_type'],
                'sortfield' => 'clock',
                'sortorder' => ZBX_SORT_DOWN,
                'limit' => 1
            ]);

            if ($history) {
```

```
                $value = convertUnitsRaw([
                    'value' => $history[0]['value'],
                    'units' => $item['units']
                ]);
            }
        }

        $this->setResponse(new CControllerResponseData([
            'name' => $this->getInput('name', $this->widget->getName()),
            'value' => $value,
            'description' => $this->fields_values['description'],
            'user' => [
                'debug_mode' => $this->getDebugMode()
            ]
        ]));
    }
}
```

7. Ouvrez `views/widget.view.php` et ajoutez la valeur de l'élément à la vue du widget.

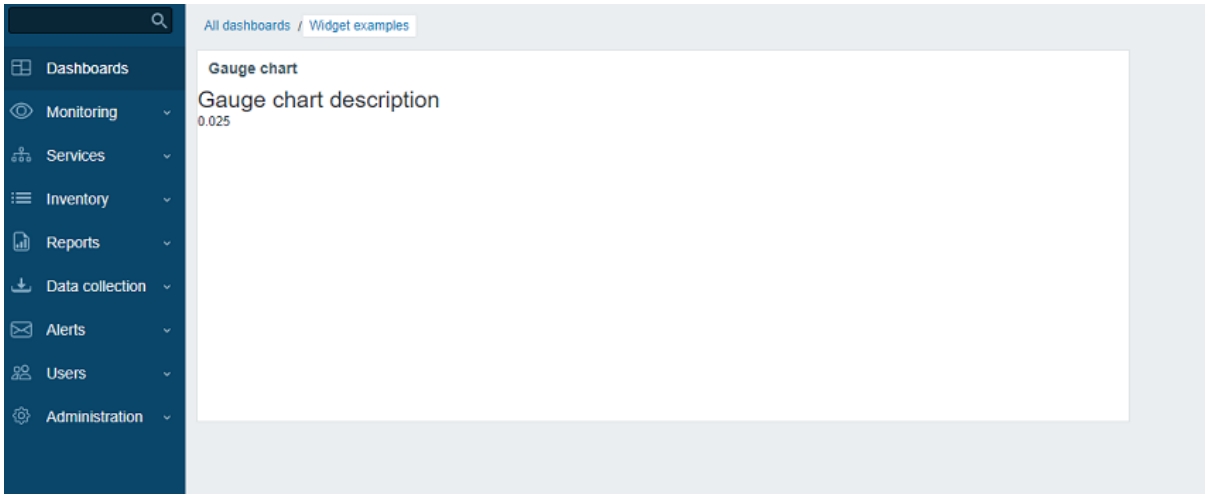
`ui/modules/lesson_gauge_chart/views/widget.view.php`

```
<?php

/**
 * Gauge chart widget view.
 *
 * @var CView $this
 * @var array $data
 */

(new CWidgetView($data))
    ->addItem([
        new CTag('h1', true, $data['description']),
        new CDiv($data['value'] !== null ? $data['value']['value'] : _('No data'))
    ])
    ->show();
```

8. Actualisez la page du tableau de bord. Le widget affichera la dernière valeur de l'élément.



Ajouter des paramètres de configuration avancés à une vue de configuration

Dans cette section, vous apprendrez à ajouter une section *Configuration avancée* extensible/réductible avec des paramètres facultatifs, tels que la couleur, les valeurs minimale et maximale, les unités et le champ *Description* créé précédemment.

1. Créez un fichier `Widget.php` dans le répertoire principal du widget `lesson_gauge_chart` pour créer une nouvelle classe `Widget`.

La classe `Widget` étendra la classe de base `CWidget` afin d'ajouter ou de remplacer les paramètres par défaut du widget (dans ce cas, les traductions). Le JavaScript, fourni ci-dessous, affiche la chaîne "No data" en cas de données manquantes. La chaîne

”No data” est présente dans les fichiers de traduction de l’interface utilisateur de Zabbix.

S’il existe des constantes de widget, il est recommandé de les spécifier également dans la classe `Widget`.

ui/modules/lesson_gauge_chart/Widget.php

```
<?php

namespace Modules\LessonGaugeChart;

use Zabbix\Core\CWidget;

class Widget extends CWidget {

    public const UNIT_AUTO = 0;
    public const UNIT_STATIC = 1;

    public function getTranslationStrings(): array {
        return [
            'class.widget.js' => [
                'No data' => _('No data')
            ]
        ];
    }
}
```

2. Ouvrez `includes/WidgetForm.php` et ajoutez les nouveaux champs *Color* (sélecteur de couleur), *Min* (champ numérique), *Max* (champ numérique) et *Units* (sélecteur), puis définissez la palette de couleurs par défaut pour le sélecteur de couleur, afin qu’elle puisse être utilisée dans les étapes suivantes.

ui/modules/lesson_gauge_chart/includes/WidgetForm.php

```
<?php

namespace Modules\LessonGaugeChart\Includes;

use Modules\LessonGaugeChart\Widget;

use Zabbix\Widgets\{
    CWidgetField,
    CWidgetForm
};

use Zabbix\Widgets\Fields\{
    CWidgetFieldColor,
    CWidgetFieldMultiSelectItem,
    CWidgetFieldNumericBox,
    CWidgetFieldSelect,
    CWidgetFieldTextBox
};

/**
 * Gauge chart widget form.
 */
class WidgetForm extends CWidgetForm {

    public const DEFAULT_COLOR_PALETTE = [
        'FF465C', 'B0AF07', '0EC9AC', '524BBC', 'ED1248', 'D1E754', '2AB5FF', '385CC7', 'EC1594', 'BAE37D',
        '6AC8FF', 'EE2B29', '3CA20D', '6F4BBC', '00A1FF', 'F3601B', '1CAE59', '45CFDB', '894BBC', '6D6D6D'
    ];

    public function addFields(): self {
        return $this
            ->addField(
                (new CWidgetFieldMultiSelectItem('itemid', _('Item')))
                    ->setFlags(CWidgetField::FLAG_NOT_EMPTY | CWidgetField::FLAG_LABEL_ASTERISK)
```

```
                ->setMultiple(false)
            )
            ->addField(
                (new CWidgetFieldColor('chart_color', _('Color')))->setDefault('FF0000')
            )
            ->addField(
                (new CWidgetFieldNumericBox('value_min', _('Min')))
                    ->setDefault(0)
                    ->setFlags(CWidgetField::FLAG_NOT_EMPTY | CWidgetField::FLAG_LABEL_ASTERISK)
            )
            ->addField(
                (new CWidgetFieldNumericBox('value_max', _('Max')))
                    ->setDefault(100)
                    ->setFlags(CWidgetField::FLAG_NOT_EMPTY | CWidgetField::FLAG_LABEL_ASTERISK)
            )
            ->addField(
                (new CWidgetFieldSelect('value_units', _('Units'), [
                    Widget::UNIT_AUTO => _x('Auto', 'history source selection method'),
                    Widget::UNIT_STATIC => _x('Static', 'history source selection method')
                ]))->setDefault(Widget::UNIT_AUTO)
            )
            ->addField(
                (new CWidgetFieldTextBox('value_static_units'))
            )
            ->addField(
                new CWidgetFieldTextBox('description', _('Description'))
            );
    }
}
```

3. Ouvrez `views/widget.edit.php` et ajoutez les composants visuels des champs à la vue de configuration.

`ui/modules/lesson_gauge_chart/views/widget.edit.php`

```
<?php

/**
 * Gauge chart widget form view.
 *
 * @var CView $this
 * @var array $data
 */

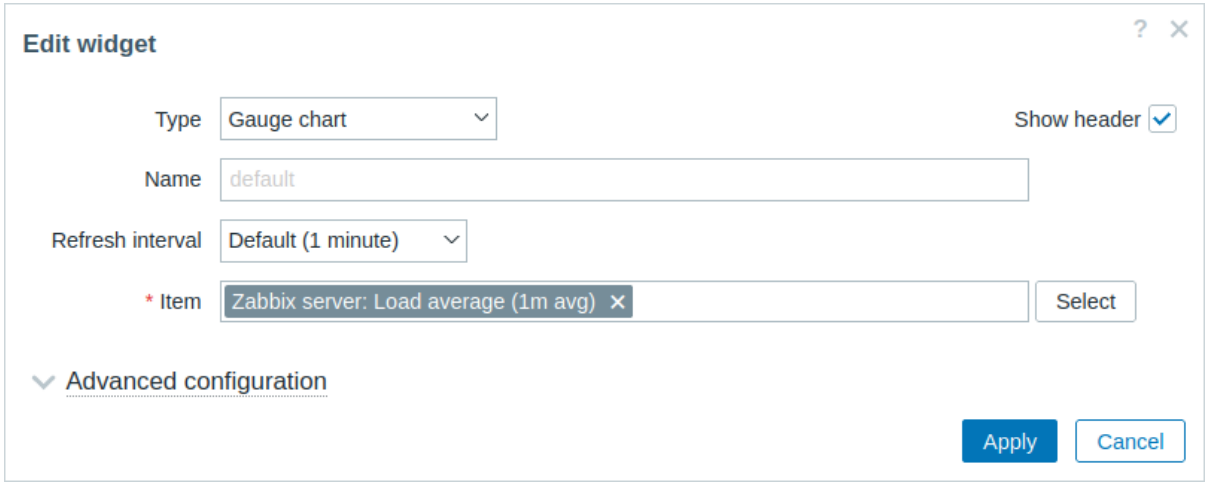
$lefty_units = new CWidgetFieldSelectView($data['fields']['value_units']);
$lefty_static_units = (new CWidgetFieldTextBoxView($data['fields']['value_static_units']))
    ->setPlaceholder(_('value'))
    ->setWidth(ZBX_TEXTAREA_TINY_WIDTH);

(new CWidgetFormView($data))
    ->addField(
        (new CWidgetFieldMultiSelectItemView($data['fields']['itemid']))
            ->setPopupParameter('numeric', true)
    )
    ->addFieldset(
        (new CWidgetFormFieldsetCollapsibleView(_('Advanced configuration')))
            ->addField(
                new CWidgetFieldColorView($data['fields']['chart_color'])
            )
            ->addField(
                new CWidgetFieldNumericBoxView($data['fields']['value_min'])
            )
            ->addField(
                new CWidgetFieldNumericBoxView($data['fields']['value_max'])
            )
```

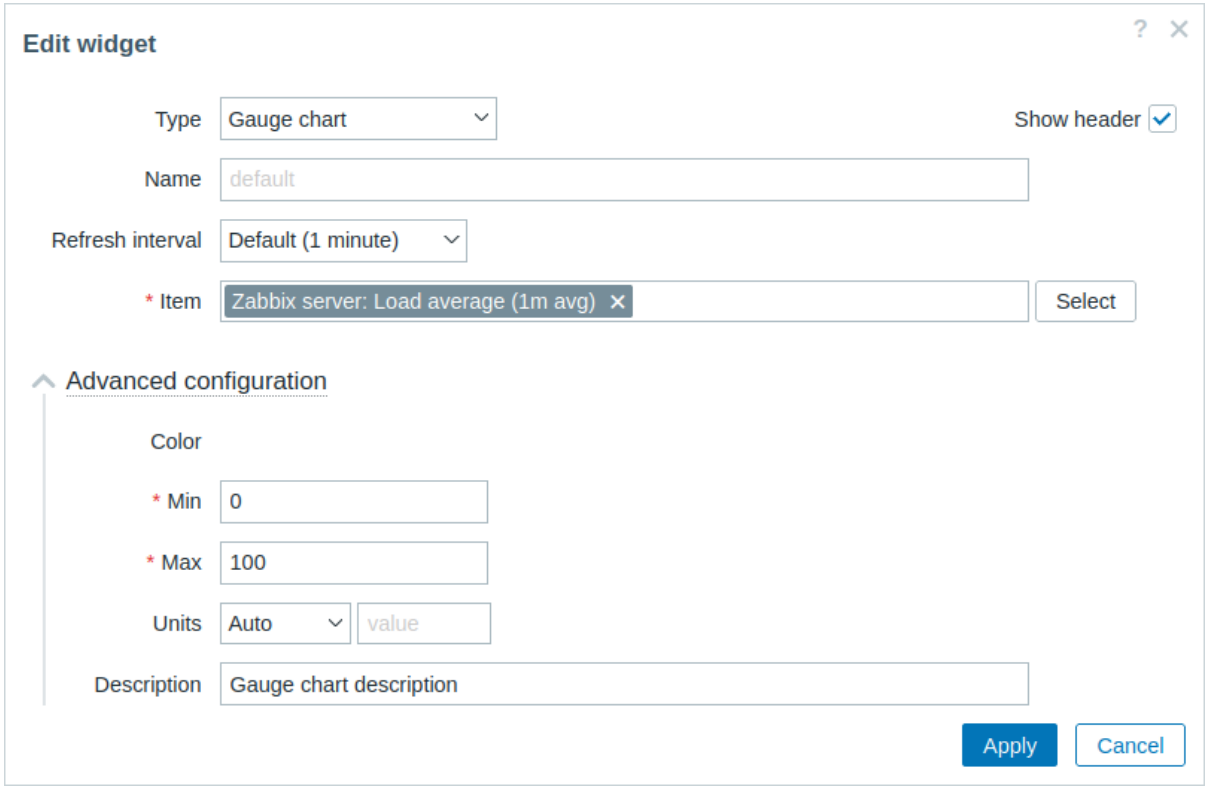
```
            ->addItem([
                $lefty_units->getLabel(),
                (new CFormField([
                    $lefty_units->getView()->addClass(ZBX_STYLE_FORM_INPUT_MARGIN),
                    $lefty_static_units->getView()
                ]))
            ])
            ->addField(
                new CWidgetFieldTextBoxView($data['fields']['description'])
            )
    )
    ->show();
```

Note:
La méthode `addField()` de la classe `CWidgetFormView` prend une chaîne de classe CSS comme deuxième paramètre.

4. Revenez au tableau de bord, passez en mode édition et cliquez sur l'icône en forme d'engrenage dans le widget pour ouvrir le formulaire de configuration du widget. Le formulaire de configuration du widget contient désormais une nouvelle section extensible/réductible *Configuration avancée*.



5. Développez la section *Configuration avancée* pour voir les champs de configuration supplémentaires du widget. Notez que le champ *Color* n'a pas encore de sélecteur de couleur. Cela est dû au fait que le sélecteur de couleur doit être initialisé avec JavaScript, qui sera ajouté dans la section suivante - *Ajouter du JavaScript au widget*.



Ajouter du JavaScript au widget

Dans cette section, vous apprendrez à ajouter un graphique en jauge - créé à l'aide de JavaScript - qui indique si la dernière valeur est normale ou trop élevée/trop faible.

1. Créez un fichier `widget.edit.js.php` dans le répertoire `views`.

JavaScript sera chargé d'initialiser le sélecteur de couleur dans la vue de configuration.

`ui/modules/lesson_gauge_chart/views/widget.edit.js.php`

```
<?php

use Modules\LessonGaugeChart\Widget;

?>

window.widget_lesson_gauge_chart_form = new class {

    init({color_palette}) {
        this._unit_select = document.getElementById('value_units');
        this._unit_value = document.getElementById('value_static_units');

        this._unit_select.addEventListener('change', () => this.updateForm());

        colorPalette.setThemeColors(color_palette);

        for (const colorpicker of jQuery('.<?= ZBX_STYLE_COLOR_PICKER ?> input')) {
            jQuery(colorpicker).colorpicker();
        }

        const overlay = overlays_stack.getById('widget_properties');

        for (const event of ['overlay.reload', 'overlay.close']) {
            overlay.$dialogue[0].addEventListener(event, () => { jQuery.colorpicker('hide'); });
        }

        this.updateForm();
    }
```

```
    updateForm() {
        this._unit_value.disabled = this._unit_select.value == <?= Widget::UNIT_AUTO ?>;
    }
};
```

2. Ouvrez `views/widget.edit.php` et ajoutez le fichier `widget.edit.js.php` avec le JavaScript à la vue de configuration. Pour cela, utilisez la méthode `includeJsFile()`. Pour ajouter du JavaScript inline, utilisez la méthode `addJavaScript()`.

ui/modules/lesson_gauge_chart/views/widget.edit.php

```
<?php

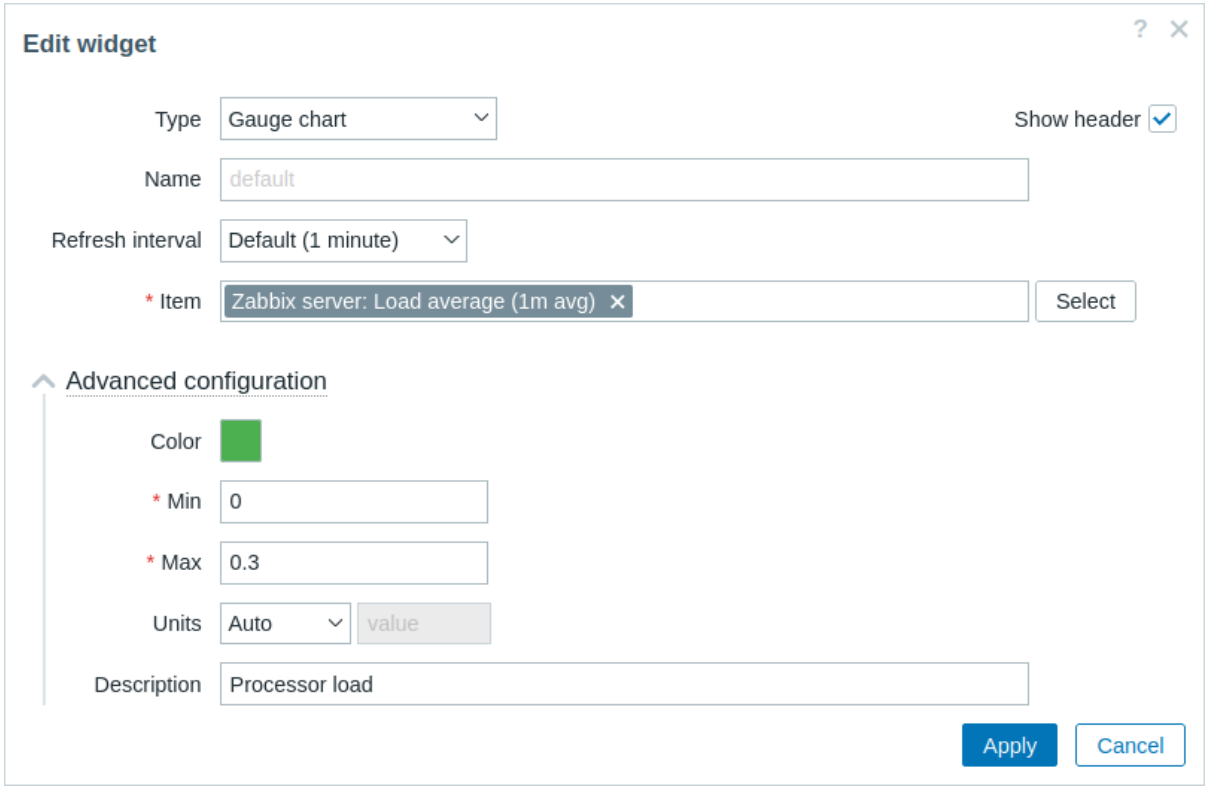
/**
 * Gauge chart widget form view.
 *
 * @var CView $this
 * @var array $data
 */

use Modules\LessonGaugeChart\Includes\WidgetForm;

$lefty_units = new CWidgetFieldSelectView($data['fields']['value_units']);
$lefty_static_units = (new CWidgetFieldTextBoxView($data['fields']['value_static_units']))
    ->setPlaceholder(_('value'))
    ->setWidth(ZBX_TEXTAREA_TINY_WIDTH);

(new CWidgetFormView($data))
    ->addField(
        (new CWidgetFieldMultiSelectItemView($data['fields']['itemid']))
            ->setPopupParameter('numeric', true)
    )
    ->addFieldset(
        (new CWidgetFormFieldsetCollapsibleView(_('Advanced configuration')))
            ->addField(
                new CWidgetFieldColorView($data['fields']['chart_color'])
            )
            ->addField(
                new CWidgetFieldNumericBoxView($data['fields']['value_min'])
            )
            ->addField(
                new CWidgetFieldNumericBoxView($data['fields']['value_max'])
            )
            ->addItem([
                $lefty_units->getLabel(),
                (new CFormField([
                    $lefty_units->getView()->addClass(ZBX_STYLE_FORM_INPUT_MARGIN),
                    $lefty_static_units->getView()
                ]))
            ])
            ->addField(
                new CWidgetFieldTextBoxView($data['fields']['description'])
            )
    )
    ->includeJsFile('widget.edit.js.php')
    ->addJavaScript('widget_lesson_gauge_chart_form.init('.json_encode([
        'color_palette' => WidgetForm::DEFAULT_COLOR_PALETTE
    ], JSON_THROW_ON_ERROR).');')
    ->show();
```

3. Revenez au tableau de bord, cliquez sur l'icône en forme d'engrenage dans le widget pour ouvrir le formulaire de configuration du widget. Développez ensuite la section *Advanced configuration* pour voir le sélecteur de couleur initialisé. Renseignez les champs avec des valeurs et sélectionnez une couleur pour le graphique en jauge.



4. Cliquez sur *Apply* dans le formulaire de configuration du widget. Cliquez ensuite sur *Save changes* dans le coin supérieur droit pour enregistrer le tableau de bord.

5. Ouvrez `actions/WidgetView.php` et mettez à jour le contrôleur.

La propriété `$this->fields_values` contient désormais les valeurs de tous les champs de *Advanced configuration*. Finalisez le contrôleur pour permettre de transmettre la configuration et la valeur de l'élément sélectionné à la vue du widget.

`ui/modules/lesson_gauge_chart/actions/WidgetView.php`

```
<?php

namespace Modules\LessonGaugeChart\Actions;

use API,
    CControllerDashboardWidgetView,
    CControllerResponseData;

class WidgetView extends CControllerDashboardWidgetView {

    protected function doAction(): void {
        $db_items = API::Item()->get([
            'output' => ['itemid', 'value_type', 'name', 'units'],
            'itemids' => $this->fields_values['itemid'],
            'webitems' => true,
            'filter' => [
                'value_type' => [ITEM_VALUE_TYPE_UINT64, ITEM_VALUE_TYPE_FLOAT]
            ]
        ]);

        $history_value = null;

        if ($db_items) {
            $item = $db_items[0];

            $history = API::History()->get([
                'output' => API_OUTPUT_EXTEND,
                'itemids' => $item['itemid'],
                'history' => $item['value_type'],
                'sortfield' => 'clock',
```

```
                'sortorder' => ZBX_SORT_DOWN,
                'limit' => 1
            ]);

            if ($history) {
                $history_value = convertUnitsRaw([
                    'value' => $history[0]['value'],
                    'units' => $item['units']
                ]);
            }
        }

        $this->setResponse(new CControllerResponseData([
            'name' => $this->getInput('name', $this->widget->getName()),
            'history' => $history_value,
            'fields_values' => $this->fields_values,
            'user' => [
                'debug_mode' => $this->getDebugMode()
            ]
        ]));
    }
}
```

6. Ouvrez et modifiez `views/widget.view.php`.

Vous devez créer un conteneur pour le graphique en jauge, que vous dessinerez dans les étapes suivantes, ainsi qu'un conteneur pour la description.

Pour transmettre des valeurs à JavaScript sous forme d'objet JSON, utilisez la méthode `setVar()`.

ui/modules/lesson_gauge_chart/views/widget.view.php

```
<?php

/**
 * Gauge chart widget view.
 *
 * @var CView $this
 * @var array $data
 */

(new CWidgetView($data))
    ->addItem([
        (new CDiv())->addClass('chart'),
        $data['fields_values']['description']
            ? (new CDiv($data['fields_values']['description']))->addClass('description')
            : null
    ])
    ->setVar('history', $data['history'])
    ->setVar('fields_values', $data['fields_values'])
    ->show();
```

7. Créez un nouveau répertoire `assets` dans le répertoire `lesson_gauge_chart`. Ce répertoire servira à stocker du JavaScript, du CSS et éventuellement d'autres ressources, comme des polices ou des images.
8. Pour le JavaScript de la vue du widget, créez un répertoire `js` dans le répertoire `assets`.
9. Créez un fichier `class.widget.js` dans le répertoire `assets/js`.

Cette classe JavaScript du widget étendra la classe JavaScript de base de tous les widgets du tableau de bord - `CWidget`.

Le tableau de bord repose sur une implémentation correcte d'un widget et lui communique toute information pertinente en appelant les méthodes JavaScript correspondantes. Le tableau de bord attend également du widget qu'il génère des événements lorsqu'une interaction se produit. Ainsi, la classe `CWidget` contient un ensemble de méthodes avec l'implémentation par défaut du comportement du widget, qui peut être personnalisée en étendant la classe.

Dans ce cas, une certaine personnalisation est nécessaire ; une logique personnalisée sera donc implémentée pour le comportement suivant du widget :

- l'initialisation du widget, qui est responsable de la définition de l'état initial du widget (voir la méthode `onInitialize()`) ;
- l'affichage du contenu du widget (c'est-à-dire le dessin du graphique en jauge) si le processus de mise à jour du widget s'est déroulé avec succès et sans erreurs (voir la méthode `processUpdateResponse(response)` et les méthodes associées `_resizeChart()` et `_updatedChart()`) ;
- le redimensionnement du widget (voir la méthode `onResize()` et la méthode associée `_resizeChart()`)

Pour les autres aspects du widget de graphique en jauge, l'implémentation par défaut du comportement du widget sera utilisée. Pour en savoir plus sur les méthodes JavaScript de la classe `CWidget`, consultez : JavaScript.

Comme ce JavaScript est requis pour la vue du widget, il doit être chargé avec la page du tableau de bord. Pour activer le chargement du JavaScript, vous devrez mettre à jour les paramètres `assets/js` et `js_class` dans le fichier `manifest.json` comme indiqué à l'étape 10.

`ui/modules/lesson_gauge_chart/assets/js/class.widget.js`

```
class WidgetLessonGaugeChart extends CWidget {

    static UNIT_AUTO = 0;
    static UNIT_STATIC = 1;

    onInitialize() {
        super.onInitialize();

        this._refresh_frame = null;
        this._chart_container = null;
        this._canvas = null;
        this._chart_color = null;
        this._min = null;
        this._max = null;
        this._value = null;
        this._last_value = null;
        this._units = '';
    }

    processUpdateResponse(response) {
        if (response.history === null) {
            this._value = null;
            this._units = '';
        }
        else {
            this._value = Number(response.history.value);
            this._units = response.fields_values.value_units == WidgetLessonGaugeChart.UNIT_AUTO
                ? response.history.units
                : response.fields_values.value_static_units;
        }

        this._chart_color = response.fields_values.chart_color;
        this._min = Number(response.fields_values.value_min);
        this._max = Number(response.fields_values.value_max);

        super.processUpdateResponse(response);
    }

    setContents(response) {
        if (this._canvas === null) {
            super.setContents(response);

            this._chart_container = this._body.querySelector('.chart');
            this._chart_container.style.height =
                `${this._getContentsSize().height - this._body.querySelector('.description').clientHeight}
            this._canvas = document.createElement('canvas');

            this._chart_container.appendChild(this._canvas);

            this._resizeChart();
```

```
        }

        this._updatedChart();
    }

    onResize() {
        super.onResize();

        if (this._state === WIDGET_STATE_ACTIVE) {
            this._resizeChart();
        }
    }

    _resizeChart() {
        const ctx = this._canvas.getContext('2d');
        const dpr = window.devicePixelRatio;

        this._canvas.style.display = 'none';
        const size = Math.min(this._chart_container.offsetWidth, this._chart_container.offsetHeight);
        this._canvas.style.display = '';

        this._canvas.width = size * dpr;
        this._canvas.height = size * dpr;

        ctx.scale(dpr, dpr);

        this._canvas.style.width = `${size}px`;
        this._canvas.style.height = `${size}px`;

        this._refresh_frame = null;

        this._updatedChart();
    }

    _updatedChart() {
        if (this._last_value === null) {
            this._last_value = this._min;
        }

        const start_time = Date.now();
        const end_time = start_time + 400;

        const animate = () => {
            const time = Date.now();

            if (time <= end_time) {
                const progress = (time - start_time) / (end_time - start_time);
                const smooth_progress = 0.5 + Math.sin(Math.PI * (progress - 0.5)) / 2;
                let value = this._value !== null ? this._value : this._min;
                value = (this._last_value + (value - this._last_value) * smooth_progress - this._min) / (t

                const ctx = this._canvas.getContext('2d');
                const size = this._canvas.width;
                const char_weight = size / 12;
                const char_shadow = 3;
                const char_x = size / 2;
                const char_y = size / 2;
                const char_radius = (size - char_weight) / 2 - char_shadow;

                const font_ratio = 32 / 100;

                ctx.clearRect(0, 0, size, size);
```

```
                ctx.beginPath();
                ctx.shadowBlur = char_shadow;
                ctx.shadowColor = '#bbb';
                ctx.strokeStyle = '#eee';
                ctx.lineWidth = char_weight;
                ctx.lineCap = 'round';
                ctx.arc(char_x, char_y, char_radius, Math.PI * 0.749, Math.PI * 2.251, false);
                ctx.stroke();

                ctx.beginPath();
                ctx.strokeStyle = `#${this._chart_color}`;
                ctx.lineWidth = char_weight - 2;
                ctx.lineCap = 'round';
                ctx.arc(char_x, char_y, char_radius, Math.PI * 0.75,
                    Math.PI * (0.75 + (1.5 * Math.min(1, Math.max(0, value)))), false
                    );
                ctx.stroke();

                ctx.shadowBlur = 2;
                ctx.fillStyle = '#1f2c33';
                ctx.font = `${(char_radius * font_ratio)|0}px Arial`;
                ctx.textAlign = 'center';
                ctx.textBaseline = 'middle';
                ctx.fillText(`${this._value !== null ? this._value : t('No data')}${this._units}`,
                    char_x, char_y, size - char_shadow * 4 - char_weight * 2
                );

                ctx.fillStyle = '#768d99';
                ctx.font = `${(char_radius * font_ratio * .5)|0}px Arial`;
                ctx.textBaseline = 'top';

                ctx.textAlign = 'left';
                ctx.fillText(`${this._min}${this._min != '' ? this._units : ''}`,
                    char_weight * .75, size - char_weight * 1.25, size / 2 - char_weight
                );

                ctx.textAlign = 'right';
                ctx.fillText(`${this._max}${this._max != '' ? this._units : ''}`,
                    size - char_weight * .75, size - char_weight * 1.25, size / 2 - char_weight
                );

                requestAnimationFrame(animate);
            }
            else {
                this._last_value = this._value;
            }
        };

        requestAnimationFrame(animate);
    }
}
```

10. Ouvrez `manifest.json` et ajoutez :

- le nom du fichier (`class.widget.js`) au tableau dans la section `assets/js` ;
- le nom de la classe (`WidgetLessonGaugeChart`) au paramètre `js_class` dans la section `widget`.

La classe `WidgetLessonGaugeChart` sera désormais chargée automatiquement avec le tableau de bord.

`ui/modules/lesson_gauge_chart/manifest.json`

```
{
    "manifest_version": 2.0,
    "id": "lesson_gauge_chart",
```

```
    "type": "widget",
    "name": "Gauge chart",
    "namespace": "LessonGaugeChart",
    "version": "1.0",
    "author": "Zabbix",
    "actions": {
        "widget.lesson_gauge_chart.view": {
            "class": "WidgetView"
        }
    },
    "widget": {
        "js_class": "WidgetLessonGaugeChart"
    },
    "assets": {
        "js": ["class.widget.js"]
    }
}
```

Ajouter des styles CSS au widget

Dans cette section, vous apprendrez à ajouter des styles CSS personnalisés pour rendre le widget plus attrayant.

1. Pour les styles du widget, créez un nouveau répertoire `css` dans le répertoire `assets`.
2. Créez un fichier `widget.css` dans le répertoire `assets/css`. Pour styliser les éléments du widget, utilisez le sélecteur `div.dashboard-widget-{widget id}`. Pour configurer le CSS pour l'ensemble du widget, utilisez le sélecteur `form.dashboard-widget-{widget id}`

ui/modules/lesson_gauge_chart/assets/css/widget.css

```
div.dashboard-widget-lesson_gauge_chart {
    display: grid;
    grid-template-rows: 1fr;
    padding: 0;
}

div.dashboard-widget-lesson_gauge_chart .chart {
    display: grid;
    align-items: center;
    justify-items: center;
}

div.dashboard-widget-lesson_gauge_chart .chart canvas {
    background: white;
}

div.dashboard-widget-lesson_gauge_chart .description {
    padding-bottom: 8px;
    font-size: 1.750em;
    line-height: 1.2;
    text-align: center;
}

.dashboard-grid-widget-hidden-header div.dashboard-widget-lesson_gauge_chart .chart {
    margin-top: 8px;
}
```

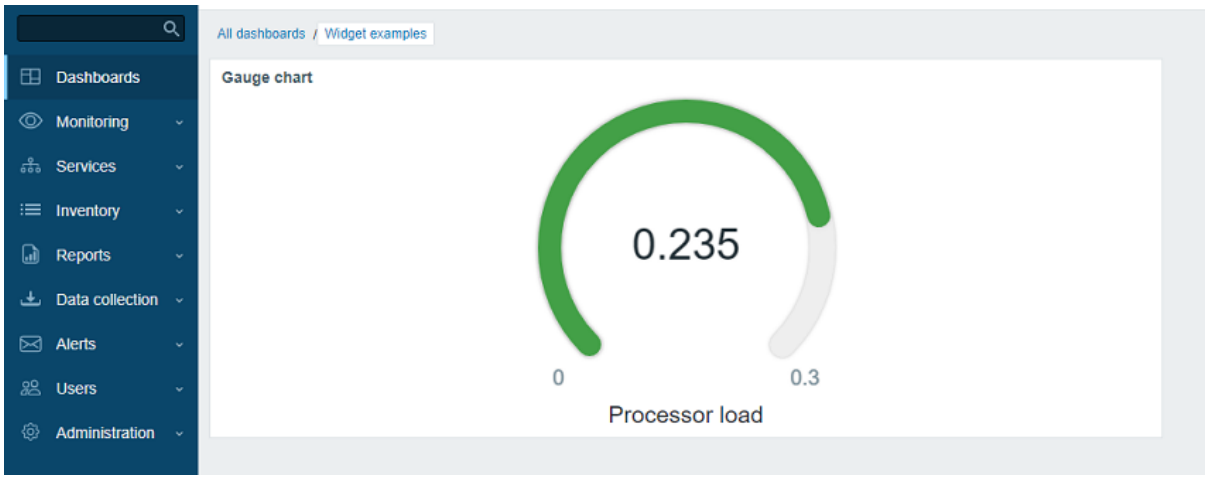
3. Ouvrez `manifest.json` et ajoutez le nom du fichier CSS (`widget.css`) au tableau dans la section `assets/css`. Cela permettra de charger les styles CSS définis dans `widget.css` avec la page du tableau de bord.

ui/modules/lesson_gauge_chart/manifest.json

```
{
    "manifest_version": 2.0,
    "id": "lesson_gauge_chart",
    "type": "widget",
    "name": "Gauge chart",
```

```
    "namespace": "LessonGaugeChart",
    "version": "1.0",
    "author": "Zabbix",
    "actions": {
        "widget.lesson_gauge_chart.view": {
            "class": "WidgetView"
        }
    },
    "widget": {
        "js_class": "WidgetLessonGaugeChart"
    },
    "assets": {
        "css": ["widget.css"],
        "js": ["class.widget.js"]
    }
}
```

4. Actualisez la page du tableau de bord pour voir la version finale du widget.



Exemples

Cette section fournit des fichiers d'exemples de modules et de widgets, que vous pouvez utiliser comme base pour vos modules personnalisés.

Pour utiliser un module :

1. Téléchargez l'archive ZIP.
2. Décompressez le contenu dans un répertoire séparé à l'intérieur du répertoire *modules* de votre installation frontale Zabbix (par exemple, *zabbix/ui/modules*).
3. Enregistrez le module dans l'interface Zabbix.

Exemple de modules

- Lors de la création d'un groupe d'hôtes, accordez des autorisations de lecture aux groupes d'utilisateurs configurés - hg_auto_perm.zip

Exemples de widgets

- Widget minimal - widget_min.zip
- Widget "Hello, world" utilisant uniquement CSS - hello_world_css.zip
- Widget "Hello, world" utilisant uniquement JavaScript - hello_world_js.zip
- Widget "Hello, world" utilisant PHP - hello_world_php.zip

Note:
Vous pouvez également utiliser les widgets natifs Zabbix comme exemples.

Modifications du développement des extensions

Changements de 7.4 à 8.0

Pages de manuel Zabbix

Documentation de référence pour les outils et services en ligne de commande de Zabbix, répertoriant les commandes disponibles, les options et des exemples d'utilisation.

zabbix_agent2

```
ZABBIX_AGENT2(8)            Manuel de l'administrateur système           ZABBIX_AGENT2(8)

NOM
       zabbix_agent2 - agent Zabbix 2

SYNOPSIS
       zabbix_agent2 [-c config-file]
       zabbix_agent2 [-c config-file] [-v] -p
       zabbix_agent2 [-c config-file] [-v] -t item-key
       zabbix_agent2 [-c config-file] -R runtime-option
       zabbix_agent2 [-c config-file] -T
       zabbix_agent2 -h
       zabbix_agent2 -V

DESCRIPTION
       zabbix_agent2  est  une  application de surveillance des paramètres de
       divers services.

OPTIONS
       -c, --config config-file
              Utiliser le config-file alternatif au lieu de celui par défaut.

       -R, --runtime-control runtime-option
              Exécuter des fonctions administratives selon runtime-option.

           Options de contrôle à l'exécution
               userparameter_reload
                   Recharger les paramètres utilisateur depuis le fichier de configuration

               log_level_increase
                   Augmenter le niveau de journalisation

               log_level_decrease
                   Diminuer le niveau de journalisation

               help
                   Lister les options de contrôle à l'exécution disponibles

               metrics
                   Lister les métriques disponibles
               version
                   Afficher la version

       -p, --print
              Afficher les éléments connus et quitter. Pour chaque élément, soit des
              valeurs  par  défaut  génériques  sont  utilisées,  soit  des valeurs
```

spécifiques par défaut pour les tests sont fournies.
Ces valeurs par défaut sont listées entre crochets comme paramètres
de clé d'élément. Les valeurs renvoyées sont placées entre crochets et
précédées du type de la valeur renvoyée, séparé par un caractère pipe.
Pour les paramètres utilisateur, le type est toujours t, car l'agent
ne peut pas déterminer toutes les valeurs de retour possibles. Les
éléments affichés comme fonctionnels ne sont pas garantis de fonctionner
depuis le serveur Zabbix ou zabbix_get lors de l'interrogation d'un
démon agent en cours d'exécution, car les permissions ou l'environnement
peuvent être différents. Les types de valeurs renvoyées sont :

d Nombre avec une partie décimale.

m Non pris en charge. Cela peut être causé par l'interrogation d'un
élément qui ne fonctionne qu'en mode actif, comme un élément de
surveillance de journal, ou d'un élément qui nécessite plusieurs
valeurs collectées. Des problèmes de permissions ou des paramètres
utilisateur incorrects peuvent également entraîner l'état non pris
en charge.

s Texte. Longueur maximale non limitée.

t Texte. Identique à s.

u Entier non signé.

-t, --test item-key
Tester un seul élément et quitter. Voir --print pour la description
de la sortie.

-v, --verbose
Activer la sortie détaillée lors du test d'un élément ou de l'affichage
des éléments connus.

-T, --test-config
Valider le fichier de configuration et quitter.

-h, --help
Afficher cette aide et quitter.

-V, --version
Afficher les informations de version et quitter.

FICHIERS
/usr/local/etc/zabbix_agent2.conf
Emplacement par défaut du fichier de configuration de l'agent
Zabbix 2 (s'il n'a pas été modifié lors de la compilation).

VOIR AUSSI
Documentation

zabbix_agentd(8), zabbix_get(1), zabbix_proxy(8), zabbix_sender(1),
zabbix_server(8), zabbix_js(1), zabbix_web_service(8)

Zabbix 2022-01-06 ZABBIX_AGENT2(8)

zabbix_agentd

ZABBIX_AGENTD(8) Manuel de l'administrateur système ZABBIX_AGENTD(8)

NOM

zabbix_agentd - démon agent Zabbix

SYNOPSIS

zabbix_agentd [-c config-file]
zabbix_agentd [-c config-file] -p
zabbix_agentd [-c config-file] -t item-key
zabbix_agentd [-c config-file] -R runtime-option
zabbix_agentd [-c config-file] -T
zabbix_agentd -h
zabbix_agentd -V

DESCRIPTION

zabbix_agentd est un démon permettant de surveiller divers paramètres du serveur.

OPTIONS

-c, --config config-file
Utiliser le fichier de configuration alternatif au lieu de celui par défaut.

-f, --foreground
Exécuter l'agent Zabbix au premier plan.

-R, --runtime-control runtime-option
Exécuter des fonctions administratives conformément à runtime-option.

Options de contrôle à l'exécution
userparameter_reload
Recharger les paramètres utilisateur depuis le fichier de configuration

log_level_increase[=target]
Augmenter le niveau de journalisation, affecte tous les processus si target n'est pas spécifié

log_level_decrease[=target]
Diminuer le niveau de journalisation, affecte tous les processus si target n'est pas spécifié

Cibles du contrôle du niveau de journalisation
process-type
Tous les processus du type spécifié (contrôles actifs, collecteur, écouteur)

process-type,N
Type de processus et numéro (par ex., listener,3)
pid Identifiant du processus, jusqu'à 65535. Pour des valeurs plus grandes, spécifier target comme "process-type,N"

-p, --print
Afficher les éléments connus et quitter. Pour chaque élément, soit des valeurs par défaut génériques sont utilisées, soit des valeurs par défaut spécifiques pour les tests sont fournies. Ces valeurs par défaut sont listées entre crochets comme paramètres de clé d'élément. Les valeurs renvoyées sont placées entre crochets et précédées du type de la valeur renvoyée, séparé par une barre verticale. Pour les paramètres utilisateur, le type est toujours t, car l'agent ne peut pas déterminer toutes les valeurs de retour possibles. Les éléments affichés comme fonctionnels ne sont pas garantis de fonctionner depuis le serveur Zabb zabbix_get lors de l'interrogation d'un démon agent en cours d'exécution, car les permission ou l'environnement peuvent être différents. Les types de valeurs renvoyées sont :

d Nombre avec une partie décimale.

m Non pris en charge. Cela peut être causé par l'interrogation d'un élément qui fonctionne uniquement en mode actif, comme un élément de surveillance de journal ou un élément qui nécessite plusieurs valeurs collectées. Des problèmes de permission ou des paramètres utilisateur incorrects peuvent également entraîner l'état non

```
               pris en charge.

           s   Texte. Longueur maximale non limitée.

           t   Texte. Identique à s.

           u   Entier non signé.

       -t, --test item-key
              Tester un seul élément et quitter. Voir --print pour la description de la sortie.

       -T, --test-config
              Valider le fichier de configuration et quitter.

       -h, --help
              Afficher cette aide et quitter.

       -V, --version
              Afficher les informations de version et quitter.

FICHIERS
       /usr/local/etc/zabbix_agentd.conf
              Emplacement par défaut du fichier de configuration de l'agent Zabbix (s'il n'a pas été mo-
              difié lors de la compilation).

VOIR AUSSI
       Documentation

       zabbix_get(1),  zabbix_proxy(8),  zabbix_sender(1),   zabbix_server(8),
       zabbix_js(1), zabbix_agent2(8), zabbix_web_service(8)

Zabbix                            2022-01-06                  ZABBIX_AGENTD(8)
```

zabbix_get

```
ZABBIX_GET(1)               Manuel général des commandes              ZABBIX_GET(1)

NOM
       zabbix_get - Utilitaire de récupération Zabbix

SYNOPSIS
       zabbix_get  -s  nom-d-hôte-ou-IP  [-p  numéro-de-port]  [-I adresse-IP] [-t
       délai-d-expiration] -k clé-d-élément
       zabbix_get -s nom-d-hôte-ou-IP  [-p  numéro-de-port]  [-I  adresse-IP]  [-t
       délai-d-expiration]   --tls-connect  cert  --tls-ca-file  fichier-CA  [--tls-crl-file
       fichier-CRL] [--tls-agent-cert-issuer émetteur-cert]  [--tls-agent-cert-sub-
       ject  sujet-cert]  --tls-cert-file  fichier-cert --tls-key-file fichier-clé
       [--tls-cipher13 chaîne-de-chiffrement] [--tls-cipher chaîne-de-chiffrement] -k clé-d-élément
       zabbix_get -s nom-d-hôte-ou-IP  [-p  numéro-de-port]  [-I  adresse-IP]  [-t
       délai-d-expiration]     --tls-connect    psk    --tls-psk-identity    identité-PSK
       --tls-psk-file fichier-PSK  [--tls-cipher13  chaîne-de-chiffrement]  [--tls-cipher
       chaîne-de-chiffrement] -k clé-d-élément
       zabbix_get -h
       zabbix_get -V

DESCRIPTION
       zabbix_get  est  un  utilitaire  en ligne de commande permettant d'obtenir des données depuis un Za
       agent.

OPTIONS
       -s, --host nom-d-hôte-ou-IP
```

```
       Spécifie le nom d'hôte ou l'adresse IP d'un hôte.

-p, --port numéro-de-port
       Spécifie le numéro de port de l'agent exécuté sur l'hôte. La valeur par
       défaut est 10050.

-I, --source-address adresse-IP
       Spécifie l'adresse IP source.

-t, --timeout secondes
       Spécifie le délai d'expiration. Plage valide : 1-600 secondes (par défaut : 30)

-k, --key clé-d-élément
       Spécifie la clé de l'élément dont la valeur doit être récupérée.

-P, --protocol valeur
       Protocole utilisé pour communiquer avec l'agent. Valeurs :

            auto        se connecter en utilisant le protocole JSON, puis revenir
                        au protocole en texte brut et réessayer si nécessaire
                        (par défaut)

            json        se connecter en utilisant le protocole JSON

            plaintext   se connecter en utilisant le protocole en texte brut où seule
                        la clé de l'élément est envoyée (versions 6.4.x et antérieures)

--tls-connect valeur
       Méthode de connexion à l'agent. Valeurs :

            unencrypted
                        se connecter sans chiffrement (par défaut)

            psk         se connecter en utilisant TLS et une clé pré-partagée

            cert        se connecter en utilisant TLS et un certificat

--tls-ca-file fichier-CA
       Chemin complet d'un fichier contenant les certificats de la ou des autorités de certificatio
       de niveau supérieur pour la vérification du certificat du pair.

--tls-crl-file fichier-CRL
       Chemin complet d'un fichier contenant les certificats révoqués.
--tls-agent-cert-issuer émetteur-cert
       Émetteur de certificat de l'agent autorisé.

--tls-agent-cert-subject sujet-cert
       Sujet de certificat de l'agent autorisé.

--tls-cert-file fichier-cert
       Chemin  complet  d'un fichier contenant le certificat ou la chaîne de certifi-
       cats.

--tls-key-file fichier-clé
       Chemin complet d'un fichier contenant la clé privée.

--tls-psk-identity identité-PSK
       Chaîne d'identité PSK.

--tls-psk-file fichier-PSK
       Chemin complet d'un fichier contenant la clé pré-partagée.

--tls-cipher13 chaîne-de-chiffrement
```

Chaîne de chiffrement pour OpenSSL 1.1.1 ou version plus récente pour TLS 1.3. Remplace les critères de sélection par défaut de la suite de chiffrement. Cette option n'est pas disponible si la version d'OpenSSL est inférieure à 1.1.1.

--tls-cipher chaîne-de-chiffrement
Chaîne de priorité GnuTLS (pour TLS 1.2 et supérieur) ou chaîne de chiffrement OpenSSL (uniquement pour TLS 1.2). Remplace les critères de sélection par défaut de la suite de chiffrement.

-h, --help
Affiche cette aide et quitte.

-V, --version
Affiche les informations de version et quitte.

STATUT DE SORTIE
Le statut de sortie est 0 si zabbix_get récupère avec succès une valeur depuis le Zabbix agent. Si la récupération de la valeur échoue, le statut de sortie est 1.

EXEMPLES
zabbix_get -s 127.0.0.1 -p 10050 -k "system.cpu.load[all,avg1]"
zabbix_get -s 127.0.0.1 -p 10050 -k "system.cpu.load[all,avg1]" --tls-connect cert --tls-ca-file /home/zabbix/zabbix_ca_file --tls-agent-cert-issuer "CN=Signing CA,OU=IT operations,O=Example Corp,DC=example,DC=com" --tls-agent-cert-subject "CN=server1,OU=IT operations,O=Example Corp,DC=example,DC=com" --tls-cert-file /home/zabbix/zabbix_get.crt --tls-key-file /home/zabbix/zabbix_get.key
zabbix_get -s 127.0.0.1 -p 10050 -k "system.cpu.load[all,avg1]" --tls-connect psk --tls-psk-identity "PSK ID Zabbix agentd" --tls-psk-file /home/zabbix/zabbix_agentd.psk

VOIR AUSSI
Documentation

zabbix_agentd(8), zabbix_proxy(8), zabbix_sender(1), zabbix_server(8), zabbix_js(1), zabbix_agent2(8), zabbix_web_service(8)

Zabbix 2022-01-06 ZABBIX_GET(1)

zabbix_js

ZABBIX_JS(1) Manuel général des commandes ZABBIX_JS(1)

NOM
zabbix_js - Utilitaire JS de Zabbix

SYNOPSIS
zabbix_js -s script-file -p input-param [-l log-level] [-t timeout]
zabbix_js -s script-file -i input-file [-l log-level] [-t timeout]
zabbix_js -h
zabbix_js -V

DESCRIPTION
zabbix_js est un utilitaire en ligne de commande qui peut être utilisé pour tester des scripts intégrés.

OPTIONS
-s, --script script-file
Indique le nom du fichier du script à exécuter. Si '-' est spécifié comme nom de fichier, le script sera lu depuis stdin.

-p, --param input-param

```
              Indique le paramètre d'entrée transmis au script. Le paramètre est
              transmis  tel  quel  sous forme de chaîne et est disponible dans le script via
              la variable 'value'. S'exclut mutuellement avec -i.

       -i, --input input-file
              Indique un fichier contenant le paramètre d'entrée transmis au
              script.  Le  contenu du fichier est transmis tel quel sous forme de chaîne et
              est disponible dans le script via la variable 'value'. Si '-' est
              spécifié comme nom de fichier, l'entrée sera lue depuis stdin. S'ex-
              clut mutuellement avec -p.

       -w, --webdriver url
              Indique l'URL du webdriver.

       -l, --loglevel log-level
              Indique le niveau de journalisation (plage : 0-5).

       -t, --timeout timeout
              Indique  le délai d'expiration en secondes. Plage valide : 1-600 secondes (par
              défaut : 10)

       -h, --help
              Affiche cette aide et quitte.

       -V, --version
              Affiche les informations de version et quitte.

EXEMPLES
       zabbix_js -s script-file.js -p example

VOIR AUSSI
       Documentation

       zabbix_agentd(8), zabbix_proxy(8), zabbix_sender(1),  zabbix_server(8),
       zabbix_get(1), zabbix_agent2(8), zabbix_web_service(8)

Zabbix                            2022-01-06                        ZABBIX_JS(1)
```

zabbix_proxy

```
ZABBIX_PROXY(8)             Manuel du gestionnaire système            ZABBIX_PROXY(8)

NOM
       zabbix_proxy - démon proxy Zabbix

SYNOPSIS
       zabbix_proxy [-c fichier-de-configuration]
       zabbix_proxy [-c fichier-de-configuration] -R option-d-exécution
       zabbix_proxy [-c fichier-de-configuration] -T
       zabbix_proxy -h
       zabbix_proxy -V

DESCRIPTION
       zabbix_proxy est un démon qui collecte les données de supervision à partir des périphériques et
       les envoie au serveur Zabbix.

OPTIONS
       -c, --config fichier-de-configuration
              Utiliser le fichier-de-configuration alternatif à la place de celui par défaut.

       -f, --foreground
```

```
       Exécuter le proxy Zabbix au premier plan.

-R, --runtime-control option-d-exécution
       Exécuter des fonctions administratives selon option-d-exécution.

    Options de contrôle à l'exécution
        config_cache_reload
            Recharger le cache de configuration. Ignoré si le cache est en cours de
            chargement. Le proxy Zabbix actif se connectera au
            serveur Zabbix et demandera les données de configuration. Le fichier de
            configuration par défaut (sauf si l'option -c est spécifiée) sera
            utilisé pour trouver le fichier PID et un signal sera envoyé au processus
            indiqué dans le fichier PID.

        snmp_cache_reload
            Recharger le cache SNMP.

        housekeeper_execute
            Exécuter le housekeeper. Ignoré si le housekeeper est
            actuellement en cours d'exécution.

        diaginfo[=section]
            Consigner les informations de diagnostic internes de la section spécifiée.
            La section peut être historycache, preprocessing, locks.
            Par défaut, les informations de diagnostic de toutes les sections sont
            consignées.

        prof_enable[=target]
            Activer le profilage, affecte tous les processus si target n'est pas
            spécifié

        prof_disable[=target]
            Désactiver le profilage, affecte tous les processus si target n'est pas
            spécifié

        log_level_increase[=target]
            Augmenter le niveau de journalisation, affecte tous les processus si target est
            pas spécifié.

        log_level_decrease[=target]
            Diminuer le niveau de journalisation, affecte tous les processus si target n'est pas
            spécifié.

    Cibles de contrôle du niveau de journalisation
        process-type
            Tous les processus du type spécifié (availability manager,
            browser poller, configuration syncer, data sender, discovery manager, history syncer, h
            mp pinger, internal poller, ipmi manager, ipmi poller, java
            poller, odbc poller, poller, agent poller, http ag-
            ent poller, snmp poller, preprocessing manager, prep-
            rocessing worker, self-monitoring, snmp trapper, super-
            visor, task manager, trapper, unreachable poller, vmware
            collector)

        process-type,N
            Type de processus et numéro (par exemple, poller,3)

        pid Identifiant du processus

    Cibles de contrôle du profilage
        process-type
            Tous les processus du type spécifié (availability manager,
            browser poller, configuration syncer, data sender, discovery manager, history syncer, h
```

```
                    mp pinger, internal poller, ipmi manager, ipmi poller, java
                    poller, odbc poller, poller, agent poller, http agent poll-
                    er, snmp poller, preprocessing manager, self-monitoring,
                    snmp trapper, task manager, trapper, unreachable poller,
                    vmware collector)

                process-type,N
                    Type de processus et numéro (par exemple, history syncer,1)

                pid Identifiant du processus

                scope
                    (rwlock, mutex, processing) peut être utilisé avec process-type
                    (par exemple, history syncer,1,processing)

       -T, --test-config
              Valider le fichier de configuration et quitter.

       -h, --help
              Afficher cette aide et quitter.

       -V, --version
              Afficher les informations de version et quitter.

FICHIERS
       /usr/local/etc/zabbix_proxy.conf
              Emplacement par défaut du fichier de configuration du proxy Zabbix (si non
              modifié à la compilation).

VOIR AUSSI
       Documentation

       zabbix_agentd(8), zabbix_get(1), zabbix_sender(1), zabbix_server(8),
       zabbix_js(1), zabbix_agent2(8), zabbix_web_service(8)

Zabbix                            2022-01-06                    ZABBIX_PROXY(8)
```

zabbix_sender

```
ZABBIX_SENDER(1)            Manuel des commandes générales           ZABBIX_SENDER(1)

NOM
       zabbix_sender - utilitaire d'envoi de données Zabbix

SYNOPSIS
       zabbix_sender [-v] -z serveur [-p port] [-I adresse-IP] [-t délai] -s
       hôte -k clé -o valeur
       zabbix_sender [-v] -z serveur [-p port] [-I adresse-IP] [-t délai] [-s
       hôte] [-T] [-N] [-r] -i fichier-d'entrée
       zabbix_sender [-v] -c fichier-de-configuration [-z serveur] [-p port] [-I adresse-IP]
       [-t délai] [-s hôte] -k clé -o valeur
       zabbix_sender [-v] -c fichier-de-configuration [-z serveur] [-p port] [-I adresse-IP]
       [-t délai] [-s hôte] [-T] [-N] [-r] -i fichier-d'entrée
       zabbix_sender [-v] -z serveur [-p port] [-I adresse-IP] [-t délai] -s
       hôte --tls-connect cert --tls-ca-file fichier-CA [--tls-crl-file fichier-CRL]
       [--tls-server-cert-issuer émetteur-du-certificat] [--tls-server-cert-subject
       sujet-du-certificat] --tls-cert-file fichier-du-certificat --tls-key-file fichier-de-clé
       [--tls-cipher13 chaîne-de-chiffrement] [--tls-cipher chaîne-de-chiffrement] -k clé -o
       valeur
       zabbix_sender [-v] -z serveur [-p port] [-I adresse-IP] [-t délai] [-s
       hôte] --tls-connect cert --tls-ca-file fichier-CA [--tls-crl-file
```

fichier-CRL] [--tls-server-cert-issuer émetteur-du-certificat]
[--tls-server-cert-subject sujet-du-certificat] --tls-cert-file fichier-du-certificat
--tls-key-file fichier-de-clé [--tls-cipher13 chaîne-de-chiffrement] [--tls-cipher
chaîne-de-chiffrement] [-T] [-N] [-r] -i fichier-d'entrée
zabbix_sender [-v] -c fichier-de-configuration [-z serveur] [-p port] [-I adresse-IP]
[-t délai] [-s hôte] --tls-connect cert --tls-ca-file fichier-CA
[--tls-crl-file fichier-CRL] [--tls-server-cert-issuer émetteur-du-certificat]
[--tls-server-cert-subject sujet-du-certificat] --tls-cert-file fichier-du-certificat
--tls-key-file fichier-de-clé [--tls-cipher13 chaîne-de-chiffrement] [--tls-cipher
chaîne-de-chiffrement] -k clé -o valeur
zabbix_sender [-v] -c fichier-de-configuration [-z serveur] [-p port] [-I adresse-IP]
[-t délai] [-s hôte] --tls-connect cert --tls-ca-file fichier-CA
[--tls-crl-file fichier-CRL] [--tls-server-cert-issuer émetteur-du-certificat]
[--tls-server-cert-subject sujet-du-certificat] --tls-cert-file fichier-du-certificat
--tls-key-file fichier-de-clé [--tls-cipher13 chaîne-de-chiffrement] [--tls-cipher
chaîne-de-chiffrement] [-T] [-N] [-r] -i fichier-d'entrée
zabbix_sender [-v] -z serveur [-p port] [-I adresse-IP] [-t délai] -s
hôte --tls-connect psk --tls-psk-identity identité-PSK --tls-psk-file
fichier-PSK [--tls-cipher13 chaîne-de-chiffrement] [--tls-cipher chaîne-de-chiffrement] -k
clé -o valeur
zabbix_sender [-v] -z serveur [-p port] [-I adresse-IP] [-t délai] [-s
hôte] --tls-connect psk --tls-psk-identity identité-PSK --tls-psk-file
fichier-PSK [--tls-cipher13 chaîne-de-chiffrement] [--tls-cipher chaîne-de-chiffrement]
[-T] [-N] [-r] -i fichier-d'entrée
zabbix_sender [-v] -c fichier-de-configuration [-z serveur] [-p port] [-I adresse-IP]
[-t délai] [-s hôte] --tls-connect psk --tls-psk-identity identité-PSK
--tls-psk-file fichier-PSK [--tls-cipher13 chaîne-de-chiffrement] [--tls-cipher
chaîne-de-chiffrement] -k clé -o valeur
zabbix_sender [-v] -c fichier-de-configuration [-z serveur] [-p port] [-I adresse-IP]
[-t délai] [-s hôte] --tls-connect psk --tls-psk-identity identité-PSK
--tls-psk-file fichier-PSK [--tls-cipher13 chaîne-de-chiffrement] [--tls-cipher
chaîne-de-chiffrement] [-T] [-N] [-r] -i fichier-d'entrée
zabbix_sender -h
zabbix_sender -V

DESCRIPTION

zabbix_sender est un utilitaire en ligne de commande permettant d'envoyer des données de supervision au serveur Zabbix ou au proxy. Sur le serveur Zabbix, un élément de type Zabbix trapper doit être créé avec la clé correspondante. Notez que les valeurs entrantes ne seront acceptées que depuis les hôtes indiqués dans le champ Hôtes autorisés de cet élément.

OPTIONS

-c, --config fichier-de-configuration

Utiliser fichier-de-configuration. Zabbix sender lit les détails du serveur dans le fichier de configuration de agentd. Par défaut, Zabbix sender ne lit aucun fichier de configuration. Seuls les paramètres Hostname, ServerActive, SourceIP, TLSConnect, TLSCAFile, TLSCRLFile, TLSServerCertIssuer, TLSServerCertSubject, TLSCertFile, TLSKeyFile, TLSPSKIdentity et TLSPSKFile sont pris en charge. Le Hostname défini via le paramètre HostnameItem ne sera pas récupéré ; dans ce cas, le nom d'hôte doit être indiqué via la ligne de commande (voir l'option -s). Toutes les adresses définies dans le paramètre de configuration ServerActive de l'agent sont utilisées pour l'envoi des données. Si l'envoi d'un lot de données échoue vers une adresse, les lots suivants ne sont pas envoyés à cette adresse.

-z, --zabbix-server serveur

Nom d'hôte ou adresse IP du serveur Zabbix. Si un hôte est supervisé par un proxy, le nom d'hôte ou l'adresse IP du proxy doit être utilisé à la place. Utilisé avec --config, ce paramètre remplace les entrées du paramètre ServerActive indiquées dans le fichier de configuration de agentd.

-p, --port port

Indiquer le numéro de port du trapper du serveur Zabbix exécuté sur le serveur. La valeur par défaut est 10051. Utilisé avec --config, ce paramètre remplace les entrées de port du paramètre ServerActive indiquées dans le fichier de configuration de agentd.

-I, --source-address adresse-IP
Indiquer l'adresse IP source. Utilisé avec --config, ce paramètre remplace le paramètre SourceIP indiqué dans le fichier de configuration de agentd.

-t, --timeout secondes
Indiquer le délai d'attente. Plage valide : 1 à 300 secondes (valeur par défaut : 60).

-s, --host hôte
Indiquer le nom de l'hôte auquel appartient l'élément (tel qu'il est enregistré dans l'interface Zabbix). L'adresse IP et le nom DNS de l'hôte ne fonctionneront pas. Utilisé avec --config, ce paramètre remplace le paramètre Hostname indiqué dans le fichier de configuration de agentd.

-k, --key clé
Indiquer la clé de l'élément auquel envoyer la valeur.

-o, --value valeur
Indiquer la valeur de l'élément.

-i, --input-file fichier-d'entrée
Charger les valeurs depuis un fichier d'entrée. Indiquer - comme <fichier-d'entrée> pour lire les valeurs depuis l'entrée standard. Chaque ligne du fichier contient, séparés par des espaces, les éléments suivants : <nom-hôte> <clé> <valeur>. Chaque valeur doit être indiquée sur sa propre ligne. Chaque ligne doit contenir 3 entrées séparées par des espaces : <nom-hôte> <clé> <valeur>, où « nom-hôte » est le nom de l'hôte supervisé tel qu'il est enregistré dans l'interface Zabbix, « clé » est la clé de l'élément cible et « valeur » est la valeur à envoyer. Indiquer - comme <nom-hôte> pour utiliser le nom d'hôte du fichier de configuration de agent ou de l'argument --host.

Exemple de ligne d'un fichier d'entrée :

```
 "Linux DB3" db.connections 43
```

Le type de valeur doit être correctement défini dans la configuration de l'élément de l'interface Zabbix. Zabbix sender envoie jusqu'à 250 valeurs par connexion. Le contenu du fichier d'entrée doit être encodé en UTF-8. Toutes les valeurs du fichier d'entrée sont envoyées dans l'ordre séquentiel, de haut en bas. Les entrées doivent être formatées selon les règles suivantes :

- Les entrées entre guillemets et sans guillemets sont prises en charge.
- Le guillemet double est le caractère de délimitation.
- Les entrées contenant des espaces doivent être placées entre guillemets.
- Les caractères guillemet double et barre oblique inverse à l'intérieur d'une entrée entre guillemets doivent être échappés avec une barre oblique inverse.
- L'échappement n'est pas pris en charge dans les entrées sans guillemets.
- Les séquences d'échappement de saut de ligne (\n) sont prises en charge dans les chaînes entre guillemets.
- Les séquences d'échappement de saut de ligne sont supprimées à la fin d'une entrée.

-T, --with-timestamps
       Cette option ne peut être utilisée qu'avec l'option --input-file.

       Chaque ligne du fichier d'entrée doit contenir 4 entrées séparées par des espaces :
       <nom-hôte> <clé> <horodatage> <valeur>. L'horodatage doit être indiqué au format
       d'horodatage Unix. Si l'élément cible possède des déclencheurs qui le référencent,
       tous les horodatages doivent être dans un ordre croissant ; sinon, le calcul de
       l'événement ne sera pas correct.

       Exemple de ligne d'un fichier d'entrée :

        "Linux DB3" db.connections 1429533600 43

       Pour plus de détails, consultez l'option --input-file.

       Si une valeur horodatée est envoyée pour un hôte qui se trouve dans une période de
       maintenance de type « aucune donnée », cette valeur sera ignorée ; toutefois, il est
       possible d'envoyer une valeur horodatée pendant une période de maintenance expirée,
       et elle sera acceptée.

-N, --with-ns
       Cette option ne peut être utilisée qu'avec l'option --with-timestamps.

       Chaque ligne du fichier d'entrée doit contenir 5 entrées séparées par des espaces :
       <nom-hôte> <clé> <horodatage> <ns> <valeur>.

       Exemple de ligne d'un fichier d'entrée :

        "Linux DB3" db.connections 1429533600 7402561 43

       Pour plus de détails, consultez l'option --input-file.

-r, --real-time
       Envoyer les valeurs une par une dès leur réception. Cette option peut être utilisée
       lors de la lecture depuis l'entrée standard.

--tls-connect valeur
       Méthode de connexion au serveur ou au proxy. Valeurs :

            unencrypted
                        se connecter sans chiffrement (valeur par défaut)

            psk   se connecter avec TLS et une clé prépartagée

            cert   se connecter avec TLS et un certificat

--tls-ca-file fichier-CA
       Chemin complet d'un fichier contenant les certificats des autorités de certification
       de niveau supérieur pour la vérification du certificat du pair.

--tls-crl-file fichier-CRL
       Chemin complet d'un fichier contenant les certificats révoqués.

--tls-server-cert-issuer émetteur-du-certificat
       Émetteur de certificat du serveur autorisé.

--tls-server-cert-subject sujet-du-certificat
       Sujet de certificat du serveur autorisé.

--tls-cert-file fichier-du-certificat
       Chemin complet d'un fichier contenant le certificat ou la chaîne de certificats.

--tls-key-file fichier-de-clé

Chemin complet d'un fichier contenant la clé privée.

--tls-psk-identity identité-PSK
Chaîne d'identité PSK.

--tls-psk-file fichier-PSK
Chemin complet d'un fichier contenant la clé prépartagée.

--tls-cipher13 chaîne-de-chiffrement
Chaîne de chiffrement pour OpenSSL 1.1.1 ou version ultérieure avec TLS 1.3. Remplace les critères de sélection de la suite de chiffrement par défaut. Cette option n'est pas disponible si la version d'OpenSSL est antérieure à 1.1.1.

--tls-cipher chaîne-de-chiffrement
Chaîne de priorité GnuTLS (pour TLS 1.2 et versions ultérieures) ou chaîne de chiffrement OpenSSL (uniquement pour TLS 1.2). Remplace les critères de sélection de la suite de chiffrement par défaut.

-v, --verbose
Mode détaillé ; -vv fournit davantage de détails.

-h, --help
Afficher cette aide et quitter.

-V, --version
Afficher les informations de version et quitter.

CODE DE SORTIE
Le code de sortie est 0 si les valeurs ont été envoyées et ont toutes été traitées avec succès par le serveur. Si les données ont été envoyées, mais que le traitement d'au moins une valeur a échoué, le code de sortie est 2. Si l'envoi des données a échoué, le code de sortie est 1.

EXEMPLES
zabbix_sender -c /etc/zabbix/zabbix_agentd.conf -k mysql.queries -o 342.45
Envoyer 342.45 comme valeur de l'élément mysql.queries de l'hôte supervisé. Utiliser l'hôte supervisé et le serveur Zabbix définis dans le fichier de configuration de l'agent.

zabbix_sender -c /etc/zabbix/zabbix_agentd.conf -s "Monitored Host" -k mysql.queries -o 342.45
Envoyer 342.45 comme valeur de l'élément mysql.queries de l'hôte Monitored Host en utilisant le serveur Zabbix défini dans le fichier de configuration de l'agent.

zabbix_sender -z 192.168.1.113 -i data_values.txt
Envoyer les valeurs du fichier data_values.txt au serveur Zabbix dont l'adresse IP est 192.168.1.113. Les noms d'hôtes et les clés sont définis dans le fichier.

echo "- hw.serial.number 1287872261 SQ4321ASDF" | zabbix_sender -c /usr/local/etc/zabbix_agentd.conf -T -i -
Envoyer une valeur horodatée depuis la ligne de commande au serveur Zabbix indiqué dans le fichier de configuration de l'agent. Le tiret dans les données d'entrée indique que le nom d'hôte doit également être utilisé depuis ce même fichier de configuration.

echo '"Zabbix server" trapper.item ""' | zabbix_sender -z 192.168.1.113 -p 10000 -i -
Envoyer une valeur vide d'un élément au serveur Zabbix dont l'adresse IP est 192.168.1.113, sur le port 10000, depuis la ligne de commande. Les valeurs vides doivent être indiquées par des guillemets doubles vides.

zabbix_sender -z 192.168.1.113 -s "Monitored Host" -k mysql.queries -o

```
       342.45  --tls-connect  cert  --tls-ca-file  /home/zabbix/zabbix_ca_file
       --tls-cert-file      /home/zabbix/zabbix_agentd.crt      --tls-key-file
       /home/zabbix/zabbix_agentd.key
              Envoyer 342.45 comme valeur de l'élément mysql.queries de l'hôte Monitored Host au
              serveur dont l'adresse IP est 192.168.1.113, en utilisant TLS avec un certificat.

       zabbix_sender -z 192.168.1.113  -s "Monitored Host" -k mysql.queries -o
       342.45  --tls-connect  psk  --tls-psk-identity  "PSK  ID Zabbix agentd"
       --tls-psk-file /home/zabbix/zabbix_agentd.psk
              Envoyer 342.45 comme valeur de l'élément mysql.queries de l'hôte Monitored Host au
              serveur dont l'adresse IP est 192.168.1.113, en utilisant TLS avec une clé
              prépartagée (PSK).

VOIR AUSSI
       Documentation

       zabbix_agentd(8),  zabbix_get(1),  zabbix_proxy(8),   zabbix_server(8),
       zabbix_js(1), zabbix_agent2(8), zabbix_web_service(8)

Zabbix                            2022-01-06                  ZABBIX_SENDER(1)
```

zabbix_server

```
ZABBIX_SERVER(8)            Manuel du gestionnaire système           ZABBIX_SERVER(8)

NOM
       zabbix_server - démon du serveur Zabbix

SYNOPSIS
       zabbix_server [-c fichier-de-configuration]
       zabbix_server [-c fichier-de-configuration] -R option-d'exécution
       zabbix_server [-c fichier-de-configuration] -T
       zabbix_server -h
       zabbix_server -V

DESCRIPTION
       zabbix_server est le démon principal du logiciel Zabbix.

OPTIONS
       -c, --config fichier-de-configuration
              Utiliser le fichier de configuration alternatif à la place de celui par défaut.

       -f, --foreground
              Exécuter le serveur Zabbix au premier plan.

       -R, --runtime-control option-d'exécution
              Exécuter des fonctions d'administration selon option-d'exécution.

           Options de contrôle à l'exécution
               config_cache_reload
                   Recharger le cache de configuration. Ignoré si le cache est actuellement en cours de ch

               snmp_cache_reload
                   Recharger le cache SNMP.

               housekeeper_execute
                   Exécuter le housekeeper. Ignoré si le housekeeper est actuellement en cours d'exécution

               trigger_housekeeper_execute
                   Exécuter le trigger housekeeper (supprimer les problèmes liés aux déclencheurs supprimé
```

```
            secrets_reload
                Recharger les secrets depuis Vault. Ignoré si le cache de configuration ou les secrets

            diaginfo[=section]
                Journaliser les informations de diagnostic internes de la section spécifiée. La section

            prof_enable[=target]
                Activer le profilage, affecte tous les processus si target n'est pas spécifié

            prof_disable[=target]
                Désactiver le profilage, affecte tous les processus si target n'est pas spécifié

            ha_status
                Afficher l'état du cluster à haute disponibilité. Peut être exécuté uniquement sur le n

            ha_remove_node
                ha_remove_node[=name|ID] Supprimer un noeud à haute disponibilité par son nom ou son ID

            ha_set_failover_delay[=delay]
                Définir le délai de basculement à haute disponibilité. Le délai peut être spécifié à l'

            log_level_increase[=target]
                Augmenter le niveau de journalisation, affecte tous les processus si target n'est pas s

            log_level_decrease[=target]
                Diminuer le niveau de journalisation, affecte tous les processus si target n'est pas sp

        Cibles de contrôle du niveau de journalisation
            process-type
                Tous les processus du type spécifié (alerter, alert manager, alert syncer, availability

            process-type,N
                Type de processus et numéro (par exemple, poller,3)

            pid Identifiant du processus

        Cibles de contrôle du profilage
            process-type
                Tous les processus du type spécifié (alerter, alert manager, alert syncer, availability

            process-type,N
                Type de processus et numéro (par exemple, history syncer,1)

            pid Identifiant du processus

            scope
                (rwlock, mutex, processing) peut être utilisé avec process-type (par exemple, history s

    -T, --test-config
           Valider le fichier de configuration et quitter.

    -h, --help
           Afficher cette aide et quitter.

    -V, --version
           Afficher les informations de version et quitter.

FICHIERS
    /usr/local/etc/zabbix_server.conf
           Emplacement par défaut du fichier de configuration du serveur Zabbix (s'il n'a pas été modif

VOIR AUSSI
    Documentation
```

```
       zabbix_agentd(8),  zabbix_get(1),  zabbix_proxy(8),   zabbix_sender(1),
       zabbix_js(1), zabbix_agent2(8), zabbix_web_service(8)

Zabbix                            2022-01-06                  ZABBIX_SERVER(8)
```

zabbix_web_service

```
ZABBIX_WEB_SERVICE(8)       Manuel de l'administrateur système      ZABBIX_WEB_SERVICE(8)

NOM
       zabbix_web_service - Service web Zabbix

SYNOPSIS
       zabbix_web_service [-c config-file]
       zabbix_web_service [-c config-file] -T
       zabbix_web_service -h
       zabbix_web_service -V

DESCRIPTION
       zabbix_web_service est une application permettant de fournir des services web aux composants
       Zabbix.

OPTIONS
       -c, --config config-file
              Utiliser le fichier de configuration config-file au lieu de celui par défaut.

       -T, --test-config
              Valider le fichier de configuration et quitter.

       -h, --help
              Afficher cette aide et quitter.

       -V, --version
              Afficher les informations de version et quitter.

FICHIERS
       /usr/local/etc/zabbix_web_service.conf
              Emplacement par défaut du fichier de configuration du service web Zabbix (s'il
              n'a pas été modifié lors de la compilation).

VOIR AUSSI
       Documentation

       zabbix_agentd(8),  zabbix_get(1),  zabbix_proxy(8),   zabbix_sender(1),
       zabbix_server(8), zabbix_js(1), zabbix_agent2(8)

Zabbix                            2022-01-06            ZABBIX_WEB_SERVICE(8)
```

Mention de copyright

La documentation Zabbix n'est PAS distribuée sous la licence AGPL-3.0. L'utilisation de la documentation Zabbix est soumise aux conditions suivantes :

Vous pouvez créer une copie imprimée de cette documentation uniquement pour votre usage personnel. La conversion vers d'autres formats est autorisée tant que le contenu réel n'est ni modifié ni édité de quelque manière que ce soit. Vous ne devez pas publier ni distribuer cette documentation sous quelque forme ou sur quelque support que ce soit, sauf si vous distribuez la documentation d'une manière similaire à celle utilisée par Zabbix pour la diffuser (c'est-à-dire électroniquement, en téléchargement sur un site web de Zabbix) ou sur une clé USB ou un support similaire, à condition toutefois que la documentation soit diffusée avec le logiciel sur le même support. Toute autre utilisation, telle que toute diffusion de copies imprimées ou l'utilisation de cette

documentation, en tout ou en partie, dans une autre publication, nécessite le consentement écrit préalable d’un représentant autorisé de Zabbix. Zabbix se réserve tous les droits sur cette documentation qui ne sont pas expressément accordés ci-dessus.